

430BOOST-SENSE1 - Capacitive Touch BoosterPack for the LaunchPad

User's Guide



Literature Number: SLAU337

April 2011

Preface	5
1 430BOOST-SENSE1 Overview	6
1.1 Overview	6
1.2 Kit Contents	7
2 Getting Started With 430BOOST-SENSE1 BoosterPack	8
2.1 Hardware Preparation	8
2.2 Software Preparation	8
2.3 Capacitive Touch User Experience	9
3 Capacitive Touch BoosterPack Hardware	13
3.1 Driving the LEDs	14
3.2 Capacitive Touch Sensors	15
4 LaunchPad Capacitive Touch BoosterPack User Experience Firmware	16
4.1 Import Project in CCS	16
4.2 Open Project and Workspace in IAR Embedded Workbench	16
4.3 Capacitive Touch Sense Library	17
5 LaunchPad Capacitive Touch BoosterPack User Experience Software	19
5.1 LaunchPad Capacitive Touch BoosterPack User Experience GUI	19
5.2 MediaPad	19
5.3 UART Communication Protocol	20
6 Frequently Asked Questions (FAQ), Tips, and Tricks	21
7 References	21
8 Schematics and PCB Layout	22
8.1 Schematics	22
8.2 PCB Layout	23
8.3 Bill of Materials (BOM)	23

List of Figures

1	Capacitive Sense BoosterPack With LaunchPad.....	6
2	PC GUI Looking for LaunchPad	10
3	PC GUI in Sleep Mode	10
4	PC GUI in Active Mode	11
5	MediaPad	12
6	Capacitive Sense BoosterPack Hardware	13
7	Schematic LED Multiplexing	14
8	Driving LEDs	14
9	Capacitive Touch Sensor Areas	15
10	Capacitive Touch BoosterPack Schematic.....	22
11	Capacitive Touch BoosterPack Layout, Top Layer	23
12	Capacitive Touch BoosterPack Layout, Bottom Layer and Silkscreen	23

List of Tables

1	BoosterPack Interface	13
2	Bill of Materials.....	23

Read This First

If You Need Assistance

If you have any feedback or questions, support for MSP430 devices, the MSP-EXP430G2 LaunchPad, and the 430BOOST-SENSE1 Capacitive Touch BoosterPack for the LaunchPad is provided by the Texas Instruments Product Information Center (PIC) and the TI E2E Forum (<https://community.ti.com/forums/12.aspx>). Contact information for the PIC can be found on the TI web site at support.ti.com. Additional device-specific information can be found on the MSP430 web site at www.ti.com/msp430.

Related Documentation from Texas Instruments

The primary sources of MSP430 information are the device-specific data sheets and user's guides. The most up-to-date versions of the user's guide documents available at www.ti.com/msp430. Information specific to the MSP-EXP430G2 LaunchPad Experimenter Board and the different BoosterPacks can be found at focus.ti.com/docs/toolsw/folders/print/msp-exp430g2.html or the LaunchPad wiki page at [processors.wiki.ti.com/index.php/MSP430_LaunchPad_\(MSP-EXP430G2\)](http://processors.wiki.ti.com/index.php/MSP430_LaunchPad_(MSP-EXP430G2)).

User's guides and detailed information on setting up a project for the MSP430 using Code Composer Studio or IAR Embedded Workbench can be found at the *Tools & Software* section of the MSP430 landing page www.ti.com/msp430.

FCC Warning

This equipment is intended for use in a laboratory test environment only. It generates, uses, and can radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to subpart J of part 15 of FCC rules, which are designed to provide reasonable protection against radio frequency interference. Operation of this equipment in other environments may cause interference with radio communications, in which case, the user will be required to take whatever measures may be required to correct this interference his own expense.

430BOOST-SENSE1 - Capacitive Touch BoosterPack for the LaunchPad

1 430BOOST-SENSE1 Overview

1.1 Overview

The 430BOOST-SENSE1 Capacitive Touch BoosterPack is the first extension module for the MSP-EXP430G2 MSP430 LaunchPad Value Line Development Kit (see [Figure 1](#)). Extension modules such as this one, designed specifically for the LaunchPad, are called BoosterPacks, and each features an application example for one of the MSP430 Value Line devices. The BoosterPacks can be connected to the MSP-EXP430G2 with both 10-pin male headers (included in MSP-EXP430G2 kit) soldered onto the board and, therefore, use all available pins on the MSP430G2452 Value Line device.



Figure 1. Capacitive Sense BoosterPack With LaunchPad

The Capacitive Sense BoosterPack is available for purchase from the TI eStore:
<https://estore.ti.com/430BOOST-SENSE1-Capacitive-Touch-BoosterPack-P2361C42.aspx>.

The Capacitive Sense BoosterPack features the new MSP430G2xx2 devices, which are capable of driving up to 16 touch-sense enabled I/O pins. The included MSP430G2452 device allows for low-cost capacitive-sensing and approximation-sensing applications without the use of any external components. The BoosterPack includes a capacitive sense board and an MSP420G2452 device preprogrammed with a demo application. The user experience application demonstrates capacitive sense as standalone feature by showing the user interaction directly with the onboard LEDs, by a GUI, or by an application example on a Microsoft Windows PC.

430BOOST-SENSE1 features:

- Nine LEDs giving instant feedback to user interaction
- Six touch-sense areas (a button, a 4-element wheel, and a proximity sensor)
- One preprogrammed MSP430G2452 device

For latest information on the LaunchPad, other available BoosterPacks, software examples, and how to program the included MSP430G2452 device, see the *MSP-EXP430G2 LaunchPad Experimenter Board User's Guide* ([SLAU318](#)) or the LaunchPad wiki page [processors.wiki.ti.com/index.php/MSP430_LaunchPad_\(MSP-EXP430G2\)](http://processors.wiki.ti.com/index.php/MSP430_LaunchPad_(MSP-EXP430G2)).

1.2 Kit Contents

The 430BOOST-SENSE1 kit includes two components:

- One Capacitive Touch BoosterPack board with nine LEDs and six sensor areas
- One preprogrammed MSP430G2452 device

This device is a low-power 16-bit microcontroller with an 8-channel 10-bit ADC, comparator, universal serial interface that supports SPI and I2C, 8kB flash memory, and 256B RAM memory

2 Getting Started With 430BOOST-SENSE1 BoosterPack

The following sections describe the preparation necessary to run the user experience application demo and to start developing applications with the MSP430G2452 for the Capacitive Touch BoosterPack.

2.1 Hardware Preparation

To prepare the Capacitive Touch BoosterPack hardware for its first use:

1. Solder both 10-pin male headers onto the LaunchPad's breakout pin connections J1 and J2. These two 10-pin male headers and two 10-pin female headers come with the original LaunchPad kit.

NOTE: If the 10-pin female headers are populated, use the 10-pin male headers as adapter to further extend the connections to the LaunchPad. The additional distance adds minimal base capacitance and does not affect the user experience of the kit.

2. Remove the J5 connections on the LaunchPad to disconnect the LaunchPad LEDs and keep them from interfering with P1.0 and P1.6 functions of the Capacitive Sense BoosterPack.
3. Ensure jumpers VCC, TXD, and RXD of the J3 connection are populated for the user experience demo to operate properly.

NOTE: The jumpers RST and TEST must also be populated when programming the device. They are not required for normal application operation.

4. Replace the existing MSP430 device in the LaunchPad MCU socket with the MSP430G2452 device that comes with the Capacitive Sense BoosterPack kit.

NOTE: Some revision 1.4 LaunchPad kits need a firmware update to support the MSP430G2452 devices; see FAQ #1 in [Section 6](#).

5. Connect the Capacitive Sense BoosterPack board to the LaunchPad with proper orientation by ensuring that the Texas Instruments logo and the text on the BoosterPack are in the same direction as the text and logo on the Launchpad.
6. Connect the LaunchPad with an USB cable to a PC or connect an external power supply (2.7 V to 3.6 V) to J6. The user experience demo application lights the center LED when power is supplied to the board.

NOTE: The 32-kHz crystal/oscillator on pins 18 and 19 is not required for the user experience application to run.

2.2 Software Preparation

The steps described in the following sections are not required for a LaunchPad Capacitive Touch BoosterPack stand-alone demo. For all other purposes that require PC software interaction, proper installation of the hardware driver and the software is required. To develop applications, its also necessary to install one of the IDEs shown on the Tools and Software section of the MSP430 landing page www.ti.com/msp430. More information on how to start developing applications for the LaunchPad and how to install the drivers and IDEs that it requires can be found on the LaunchPad wiki page [http://processors.wiki.ti.com/index.php/MSP430_LaunchPad_\(MSP-EXP430G2\)](http://processors.wiki.ti.com/index.php/MSP430_LaunchPad_(MSP-EXP430G2)).

All LaunchPad Capacitive Touch BoosterPack User Experience firmware and software described in the following sections are provided in both binary/executable and source code forms, along with drivers and supporting documentation. [10] A zip file containing these items can be downloaded from www.ti.com/lit/zip/slac490.

The same software package link and updates can also be found on the LaunchPad wiki page [http://processors.wiki.ti.com/index.php/MSP430_LaunchPad_\(MSP-EXP430G2\)](http://processors.wiki.ti.com/index.php/MSP430_LaunchPad_(MSP-EXP430G2)). When this package is installed, all user experience application demos are stored in the *Software* folder in the selected installation directory, and the source code for the projects can be found in the *Source* folder.

2.2.1 LaunchPad USB Driver

For the PC to communicate with the LaunchPad hardware, the LaunchPad USB driver must be installed. If this is the first time the LaunchPad is connected to the PC, install the USB serial COM port driver located at [INSTALL_PATH]\LaunchPad_Driver\LaunchPad_Driver.exe.

NOTE: The LaunchPad USB drivers are integrated into the IDE installer packages from Code Composer Studio (version 4+) or IAR Embedded Workbench (version 5.20+) and do not require a second installation if an IDE has already been installed.

2.2.2 Locate Software Programs

Two software programs are provided to work with the LaunchPad Capacitive Touch BoosterPack user experience demo. They are both installed to the [INSTALL_PATH]\Software folder. All software and firmware examples are also available as source code in the capacitive touch software package.

2.3 Capacitive Touch User Experience

The Capacitive Touch User Experience consists of three projects:

- The firmware application that can operate in stand-alone mode (LaunchPad Capacitive Touch BoosterPack Firmware Demo)
- A processing GUI that displays the information from the LaunchPad Capacitive Touch BoosterPack visually (LaunchPad Capacitive Touch BoosterPack GUI Demo)
- A Visual Studio program that uses the LaunchPad Capacitive Touch BoosterPack input to control media in Windows (MediaPad).

2.3.1 LaunchPad Capacitive Touch BoosterPack Firmware Demo

The application described in this section can be used as either a stand-alone demo (no PC required) or as a demo with PC applications running. If PC application is desired, make sure to start the PC application execution before proceeding with step 2.

1. Plug the LaunchPad with Capacitive Touch BoosterPack into a USB source (such as USB port on PC, USB hub, or USB battery pack) via the mini-USB connector or to a battery pack via the power pin connector J6. The User Experience application starts up and remains in sleep mode, with only the center LED on.
2. Slowly wave your hand or finger approximately 3 to 5 cm above the BoosterPack to trigger the proximity sensor and to wake the device. During the wake-up period, the LEDs surrounding the wheel light in a wake-up sequence, starting with a slow clockwise rotation and ending with a fast counter-clockwise rotation. As this sequence ends, the device enters active mode.
3. To perform a touch, firmly press any position on the wheel or the center button. Make sure to keep your finger between the circles of the wheel.
 - Upon releasing a touch on the center button, the center LED toggles.
 - Touching a wheel position lights up the corresponding LEDs.
4. To perform a gesture, slide your finger along the wheel without releasing it from the wheel. The corresponding LEDs trace and follow the touch and gesture.
5. After a short time of no capacitive touch activity, the board returns to sleep mode. Only the center LED stays on.
6. Go back to step 2 to re-enable the application active mode.

2.3.2 LaunchPad Capacitive Touch BoosterPack GUI Demo

This section describes how to run the application on both the LaunchPad Capacitive Touch BoosterPack and the PC. It assumes that the hardware is connected to the PC via USB cable (see step 1 of [Section 2.3.1](#)) and that the software has been installed (see [Section 2.2](#)).

The following steps correspond to the instructions in [Section 2.3.1](#).

1. Start the CapTouch_BoosterPack_UserExperience_GUI.exe application located at [INSTALL_PATH]\Software\CapTouch_BoosterPack_UserExperience_GUI.
When the GUI starts, it checks for a valid LaunchPad USB serial COM port. If no compatible port connection is available, the GUI prompts user to plug in the LaunchPad Capacitive Touch BoosterPack (see [Figure 2](#)). The GUI continues normally if it detects that a LaunchPad is plugged into the PC.



Figure 2. PC GUI Looking for LaunchPad

2. At start-up or after long period of inactivity, the device enters sleep mode and the GUI is disabled (grayed out) to indicate sleep mode (see [Figure 3](#)). Upon proximity sensor detection (for example, wave your hand approximately 3 to 5 cm above the BoosterPack) the device returns to active mode and enables the GUI.

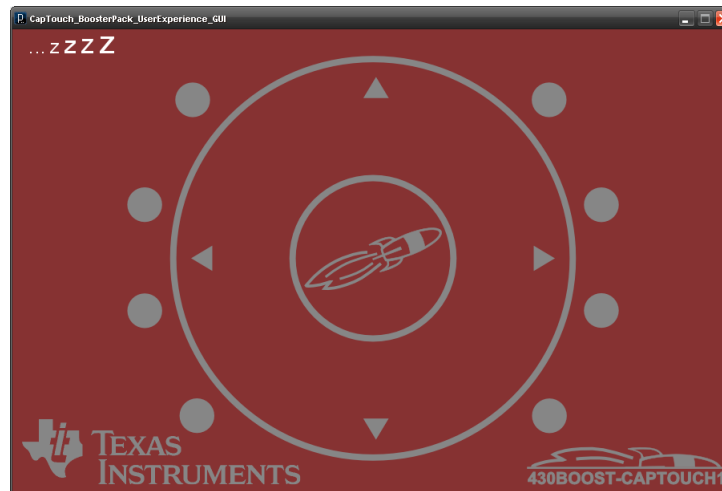


Figure 3. PC GUI in Sleep Mode

3. The 'Center Button' press data toggles the center circle color, mimicking the behavior of the center LED on the BoosterPack.
The 'Wheel Tap' is represented by lighting up a single slice on the wheel and displaying the field number on the top left corner of the PC GUI.

4. The gesture tracking (Start, Stop, and Update) is visualized on the wheel with the coloring of the wheel slices (see [Figure 4](#)). Gesture can be tracked for several revolutions of the wheel, in both clockwise and counter-clockwise directions⁽¹⁾.

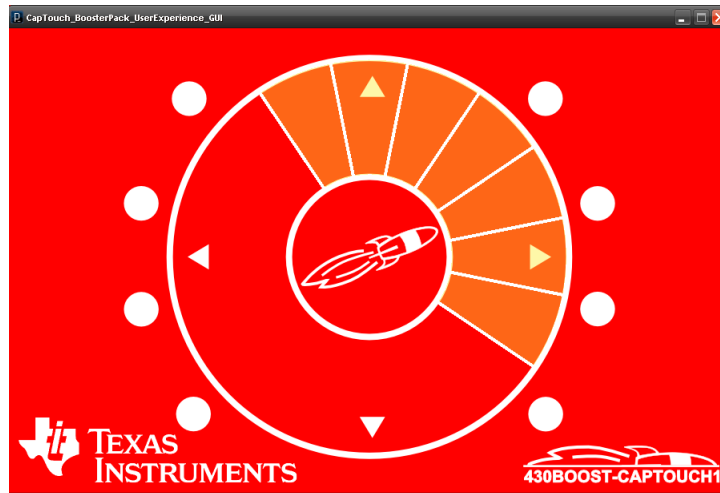


Figure 4. PC GUI in Active Mode

5. After a short time when no capacitive touch activity is detected, the board returns to sleep mode and the GUI is disabled.

2.3.3 MediaPad

This section describes instructions to run the MediaPad application on the LaunchPad Capacitive Touch BoosterPack and the PC. It assumes that the hardware is connected to the PC via USB cable (see step 1 of [Section 2.3.1](#)) and that the software has been installed (see [Section 2.2](#)).

1. Start the MediaPad.exe application located at [INSTALL_PATH]\Software\MediaPad\
2. At startup, the application searches for a LaunchPad or an eZ430 emulator compatible USB serial COM port.
If no compatible COM port is found, the application displays an error message and then exits.
If a LaunchPad COM port is found, the application displays a greeting message. When the user closes the message, the application minimizes itself to the taskbar.
3. When the LaunchPad Capacitive Touch BoosterPack is in sleep mode, no data is transferred and no activity occurs in the program. Use hand/finger wave motion to trigger the proximity sensor and wake up the device.
4. The following touches or gestures can be used for media control in a Windows system (see [Figure 5](#)).
 - (a) Center button press: Start media player (Windows Media Player by default)
 - (b) Bottom arrow button press: Play/Pause
 - (c) Left arrow button press: Previous Track
 - (d) Right arrow button press: Next Track
 - (e) Scroll wheel clockwise: Volume Up
 - (f) Scroll wheel counter-clockwise: Volume Down

⁽¹⁾ Using the wheel, a hidden mode can be unlocked. Input the correct sequence (similar to a rotational combination lock) to reveal a secret.

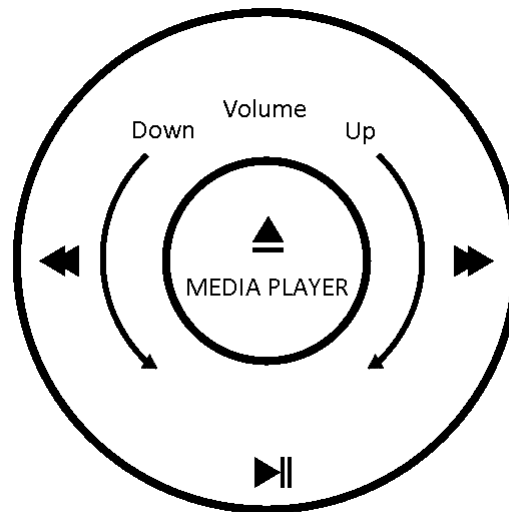


Figure 5. MediaPad

NOTE: The Microsoft .NET runtime library is required to run the MediaPad software. While most recent Windows PCs have the .Net runtime library installed, a new or reinstall of the library might be necessary.

3 Capacitive Touch BoosterPack Hardware

As shown in [Figure 6](#) and [Table 1](#), the Capacitive Touch BoosterPack is a typical capacitive touch application example for the new MSP430G2xx2 touch-sense enabled I/O pins. The board shows three different capacitive sensor types: a single button in the middle, a wheel made of four single capacitive sensors, and a proximity sensor around the edge of the PCB. In addition, there are nine LEDs on the board to give instant feedback to user interaction. The eight LEDs around the wheel are multiplexed to increase their number without using too many of the microcontrollers I/O pins. By using a time-shared signal, only five I/O pins are used to drive all eight LEDs.

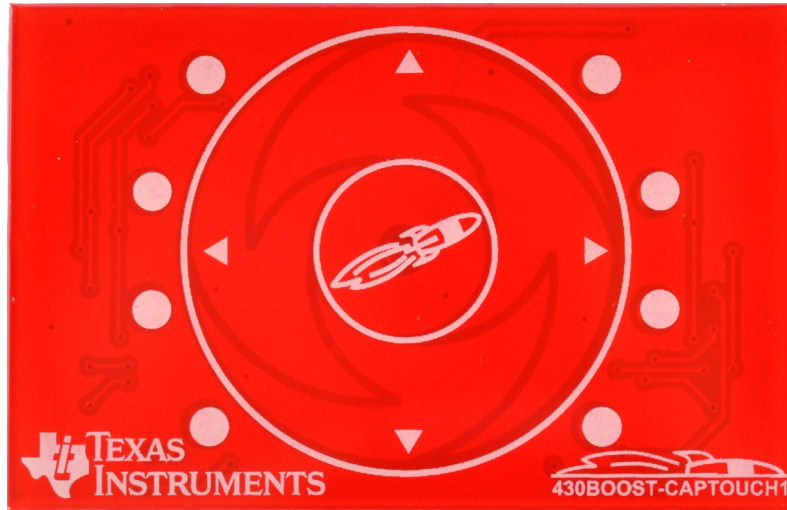


Figure 6. Capacitive Sense BoosterPack Hardware

Table 1. BoosterPack Interface

Pin	MSP430 Port	BoosterPack Signal	Description
1	VCC	NC	Supply voltage, not connected to BoosterPack
2	P1.0	LED9	The white center LED
3	P1.1/TXD	NC	Backchannel UART transmit data output, not connected to BoosterPack
4	P1.2/RXD	NC	Backchannel UART receive data input, not connected to BoosterPack
5	P1.3	LEDx	LED base to drive the eight multiplexed LEDs
6	P1.4	LED1	LED1 positive and LED5 negative drive
7	P1.5	LED2	LED2 positive and LED6 negative drive
8	P2.0	SENS0	Touch-sense proximity sensor
9	P2.1	SENS1	Touch-sense wheel sensor left
10	P2.2	SENS2	Touch-sense wheel sensor down
11	P2.3	SENS3	Touch-sense wheel sensor right
12	P2.4	SENS4	Touch-sense wheel sensor up
13	P2.5	SENS5	Touch-sense center button sensor
14	P1.6	LED3	LED3 positive and LED7 negative drive
15	P1.7	LED4	LED4 positive and LED8 negative drive
16	RST/SBWT DIO	NC	Reset line for SBW JTAG data, not connected to BoosterPack
17	TEST/SBW TCK	NC	Test line for SBW JTAG clock, not connected to BoosterPack
18	P2.6/XOUT	NC	Oscillator output, not connected to BoosterPack
19	P2.7/XIN	NC	Oscillator input, not connected to BoosterPack
20	GND	GND	Supply ground

3.1 Driving the LEDs

The white center LED is connected to port P1.0 of the Value Line device and can be turned on by setting this port as output. The other eight LEDs are multiplexed as shown in [Figure 7](#). Four LEDs are connected with the diode cathodes to the ports P1.4 to P1.7, and the other four LEDs are connected with the diode anodes. All LEDs are connected to port P1.3 to either source or drain four LEDs at any one time. To drive a particular set of these eight LEDs, two steps are required:

1. Port P1.3 acts as GND drain, so that the first nibble can drive LED1 to LED4 directly. The other LEDs are not active during this time.
2. Port P1.3 acts as a VCC source to light LED5 to LED8. The pattern for the second nibble must be inverted and set to port P1.4 to P1.7.

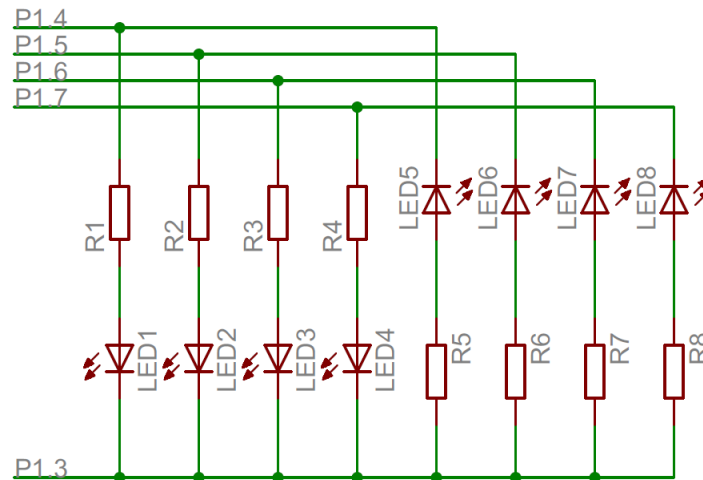


Figure 7. Schematic LED Multiplexing

Switching between modes can be controlled by a timer and has to be at least 100 times a second to generate the illusion of a constant light pattern. To reduce glitches while switching the LED modes, it is recommended to set the LED signals that are not being driven to input mode. [Figure 8](#) shows the signals of all the LED driving pins required to light up LED1, LED3, LED6, and LED7. The mode on P1.3 can be set before or after the LED settings, as long as the unused LEDs are set to input mode.

NOTE: The current user experience implementation is not using a time-shared approach to drive the LEDs, due to the shared CPU and timer resources utilized by the Capacitive Touch Sense Library [7] functions and the UART transmissions.

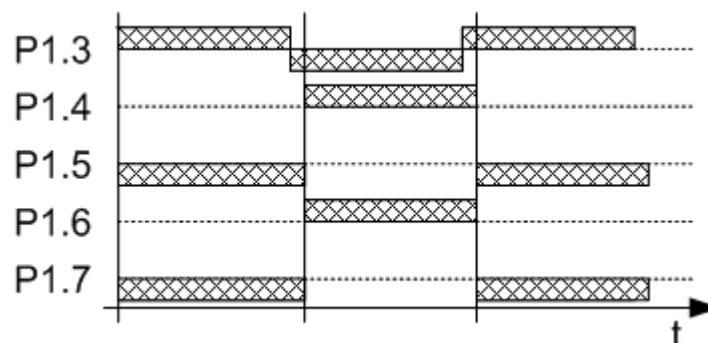


Figure 8. Driving LEDs

3.2 Capacitive Touch Sensors

The six different capacitive touch sensor areas are connected to the Port 2 of the device. On the MSP430 Value Line devices with enabled touch-sense, Port 2 I/Os have no analog functionality. These I/Os also have a smaller internal capacitance than Port 1, which make these them more sensitive than the capacitive touch sensors on Port 1.

Figure 9 shows the connection of the capacitive touch sensor areas to the MSP430.

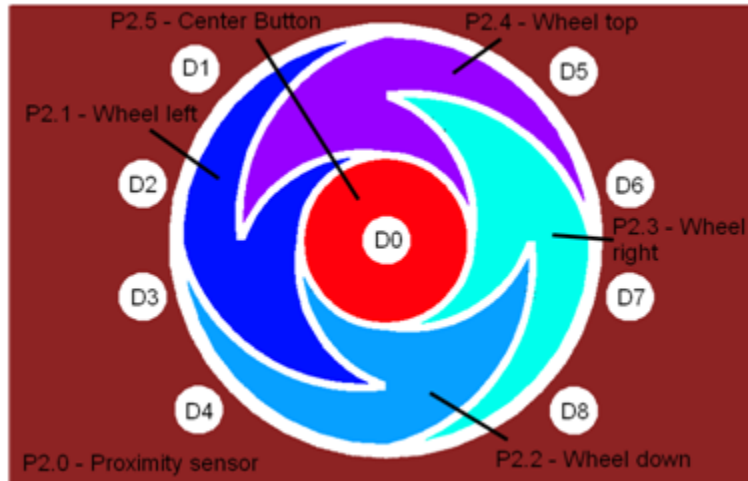


Figure 9. Capacitive Touch Sensor Areas

To enable the touch-sense feature of the I/Os, set the secondary port select PxSEL2 and clear the PxSEL bit. The selected pins start oscillating immediately, and the frequency is a direct indication of the capacitance connected to the port pin. The touch-sense I/Os oscillate within a frequency range of 1 to 2 MHz, which is strongly dependant on the supply voltage, the device package, and environmental influences. For more information about the touch-sense feature of the Value Line devices, to download code example, or to find application examples, go to the Capacitive Touch BoosterPack wiki page ([http://processors.wiki.ti.com/index.php/MSP430_LaunchPad_\(MSP-EXP430G2\)](http://processors.wiki.ti.com/index.php/MSP430_LaunchPad_(MSP-EXP430G2))) or get the MSP430 Capacitive Touch Sense Library (<http://focus.ti.com/docs/toolsw/folders/print/capsenselibrary.html>), which is used in the application demo included in this kit.

NOTE: The Capacitive Touch BoosterPack hardware can also be used with other MSP430 devices. Resistors R10 to R15 must be populated with a resistor to realize a RC discharge that can be measured with a timer.

4 LaunchPad Capacitive Touch BoosterPack User Experience Firmware

This section describes the firmware application that is provided with the project. Detailed information on the project construction, use of the Capacitive Touch Sense Library, and how to set up and import the projects is included. Source code for the MSP430G2452 firmware application is installed to the [INSTALL_PATH]\Source\ folder as described in [Section 2.2](#).

The User Experience application operates on the LaunchPad platform using the MSP430G2452 device and the Capacitive Touch BoosterPack plugin board. The capacitive touch and proximity sensing are enabled by the pin oscillator feature, which is new to the MSP430G2xx2 family devices. The User Experience application also uses the Capacitive Touch Sense Library [7] to realize and measure the capacitive touch and proximity sensors. The Capacitive Touch Sense Library provides layers of abstraction to generate higher logical outputs such as logical touches and their position (in this hardware, a four-button wheel).

The User Experience application starts in sleep mode, sampling the proximity sensor approximately every 8.3 ms ($VLO / 100 = 12 \text{ kHz} / 100 = 120 \text{ Hz}$). Upon registering a valid proximity event (for example, a hand, finger, or object hovering approximately 3 to 5 cm from the BoosterPack), the application enters the active mode. During the wake-up period, the LEDs surrounding the wheel light in a wake-up sequence, starting with a slow clockwise sequence and ending with a fast counter-clockwise sequence. As this sequence ends, the device enters active mode.

In active mode, the application samples and registers individual finger touches on the 16-position wheel or the center button. It also recognizes simple gestures (clockwise and counter-clockwise) when the finger moves along and remains on the wheel. Upon wheel position detection, the corresponding LEDs surrounding the wheel light up accordingly. Each individual tap on the center capacitive touch button toggles the center LED.

After a short time without any touch activity (on the wheel or on the center button), the application returns to sleep mode, enabling only the proximity sensor periodically.

A 9600-baud UART link is also implemented using software Timer_A to provide application and capacitive sensing data to the PC via the UART-USB back channel. The application sends UART data on events such as wake up, sleep, touch, or gesture.

For more detailed information on the firmware project, see the source code and the associated ReadMe.txt.

4.1 Import Project in CCS

To import the project into CCS:

- (a) Open CCS.
- (b) Select a new project workspace outside of the project folder⁽¹⁾.
- (c) Select Project-->Import Existing Project.
- (d) Browse to the [PROJECT_ROOT]\CCS folder.
- (e) Click Finish

NOTE: For CCS, while project root is in the outer directory, the CCS project files are located inside CCS. To enable the portability of the project, the file macros.ini is created to define the root. Additionally, all project code files (.c, .h) are added as linked resources with their relative path to the project root.

4.2 Open Project and Workspace in IAR Embedded Workbench

To open the project in IAR Embedded Workbench:

- (a) Browse to the [PROJECT_ROOT]\IAR folder.
- (b) Open the Sense_BoosterPack_UserExperience.eww workspace.

⁽¹⁾ The workspace should be in an independent folder, not containing or contained by the project/package folder.

4.3 Capacitive Touch Sense Library

The Capacitive Touch Sense Library ([CAPSENSELIBRARY](#)) is a configurable tool to abstract the various peripheral settings from the application and perform several capacitive touch functions through API calls. The following describes the configuration of the library to support the Capacitive Touch BoosterPack, the methodology to calibrate the different elements, and how the API calls are used in the application to create the user experience.

1. Configuration

The first step in the configuration process is identifying the methodology used to measure the capacitance. For the Capacitive Touch BoosterPack, the goal is to highlight the new PinOsc feature; therefore, an RO implementation is chosen, and the relaxation oscillator is implemented with the PinOsc. The RO implementation requires two timers (hardware or software timers): an interval timer (gate) and a frequency counter. The frequency counter is implemented with the Timer_A0 peripheral, and the interval timer is implemented with the WDT+.

The Capacitive Touch BoosterPack is represented in the various structures defined in the file structure.c. The element structures define the GPIO and the performance characteristics of each element. The GPIOs are defined first, the appropriate sensor characteristics are defined, and the performance characteristics are measured and added.

The sensor structure groups elements as appropriate and identifies the measurement characteristic for that group, namely the interval period. For the RO method, increasing the interval time increases the sensitivity; however, this is at the cost of response time, which is critical for supporting the PC GUI in this application.

The proximity sensor uses an SMCLK of 125 kHz. For the button and wheel, the frequency is increased to 1 MHz. The interval count is 8192: 8.192-ms gate time for the button and wheel elements and 65.5-ms gate time for the proximity element. The wheel is a special kind of sensor in which each element contributes to the sensor performance. The wheel is made up of four elements divided into 64 points or sections and requires that the cumulative response exceed 75 percent. This percentage is based upon the normalized response where meeting the threshold would represent 0% and the maximum response would represent 100%. This is to account for cases when the interaction is near the edges of the wheel instead of the middle.

2. Calibration

The calibration of the middle button and the proximity sensor are relatively straight forward, because the desired output is a binary indication of whether or not the threshold is exceeded. Using a controlled test fixture to represent the minimum touch (distance in the case of the proximity sensor), the values are recorded and input as the threshold value in the element structure.

The calibration for the wheel is more complicated, as several measurements are required at various positions. See the "Sensor Arrays: Wheels and Sliders" section in the *Capacitive Touch Sense Library User's Guide* ([SLAA490](#)) for a detailed explanation.

The calibration values for each element are recorded in the element structure in the file structure.c.

3. API Calls

There are five API functions that are called several times in the application.

- The `TI_CAPT_Init_Baseline` and `TI_CAPT_Update_Baseline` functions initialize and update, respectively, the baseline tracking performed by the library. Typically, these functions are called at the beginning of an application or after long periods of inactivity. In this application, the initialization and updates are performed after a power-up sequence and before a transition from the sleep (polling proximity sensor only) to active (polling button and wheel only). These functions are used at the transitions, because it is unknown how old the previous measurements are and if these still represent the current environment.
- The `TI_CAPT_Custom` function measures the proximity sensor. The variable `dCnt` is updated with the measured value. In this application, this variable is compared to a threshold value. Because this is a simple On/Off function, the `TI_CAPT_Button` function could have been used but, for demonstration purposes, the `TI_CAPT_Custom` function was chosen. When a threshold crossing is detected, an LED sequence is started, and the application transitions to the active state (polling the wheel and middle button). One possible enhancement of the proximity sensor application is to enable several different thresholds and indicate how close the user is with the LEDs on the BoosterPack.
- The `TI_CAPT_Button` function determines if the middle button has been detected. This function returns either a 1 to indicate a threshold crossing (touch) or a 0 to indicate that no touch was detected. The middle LED is illuminated to indicate a touch.
- The `TI_CAPT_Wheel` function indicates the position on the wheel if it was touched or returns a defined value if no touch was detected. This information is used by the application for gesture recognition (which is sent to the PC) and for illuminating the eight LEDs around the board.

For more information on the library, see the *Capacitive Touch Sense Library User's Guide* ([SLAA490](#)).

5 LaunchPad Capacitive Touch BoosterPack User Experience Software

5.1 LaunchPad Capacitive Touch BoosterPack User Experience GUI

Written in Processing, this Windows PC GUI application communicates with the LaunchPad to receive specific capacitive touch data from the LaunchPad Capacitive Touch BoosterPack and provides the visualization of that data in the GUI. Processing is a platform-independent open-source programming language and environment, specializing in visual arts, graphics, and interactive applications.

The GUI uses a small .NET utility (FindAppUART.exe) to automatically detect a proper LaunchPad/430Emulator device connected to the PC USB port. Upon correct USB COM port discovery, the application initiates a 9600-baud UART connection and starts receiving data.

The GUI processes event and capacitive touch data and visualizes the data on the GUI in a 16-slice wheel formation. Individual touches as well as gestures can be tracked in real time.

Further description of the behavior can be found in [Section 2.3.2](#) and the ReadMe.txt in the project source code directory.

The application also takes advantage of the serial library for USB COM serial communication, and the sound library pitaru.sonia_v2_9 (available at <http://sonia.pitaru.com/download.htm>) for audio effects.

5.1.1 Requirements

The following utilities and libraries are required when modifying the User Experience source code.

- Processing (www.processing.org)
- Serial library (included with Processing installation)
- pitaru.sonia_v2_9 sound library (sonia.pitaru.com/download.htm)
- FindAppUART.exe (included .NET utility)

5.2 MediaPad

The program MediaPad, written using Visual Studio, translates capacitive touch data from from the LaunchPad Capacitive Touch BoosterPack into Microsoft Windows virtual keystrokes for Windows media control. The application implements auto-detection code that automatically finds a LaunchPad-compatible USB COM port before establishing the proper connection. Further behavior of the application is described in [Section 2.3.2](#) and the ReadMe.txt in the project source code directory.

5.2.1 Requirements

When modifying the MediaPad source code, Microsoft Visual C++ 2010 Redistributable Package (included in any version of Visual Studio 2010) (available at <http://www.microsoft.com/downloads>) is required.

5.3 UART Communication Protocol

For each event (wake up, go to sleep, touch/press, or gesture), a UART packet of two bytes is sent via the application UART backchannel of the LaunchPad. The packets are specified as follows:

- WAKE UP [due to proximity sensor detection]: 0xBE 0xEF
- SLEEP [after period of inactivity]: 0xDE 0xAD
- CENTER BUTTON PRESS: 0x80 0x80
- WHEEL POSITION TOUCH/PRESS: 0x3z 0x3z
z = touch position 0x0 to 0xF, one nibble
- GESTURE START: 0xFC 0x2z
z = touch position 0x0 to 0xF, one nibble
- GESTURE STOP: 0xFB 0xFB
- GESTURE and GESTURE END POSITION : 0xGG 0x2
GG = a binary number
MSB is direction: 0 = clockwise, 1 = counter-clockwise
7 LSBs = count of gesture movement
z = ending position of the immediate gesture, 0x0 to 0xF, one nibble.

The PC application can receive and decipher the UART information to translate it into appropriate actions.

6 Frequently Asked Questions (FAQ), Tips, and Tricks

1. The LaunchPad is unable to program the MSP430G2452.
Some of the revision 1.4 LaunchPad kits must have a firmware update to support the MSP430G2452 devices. Update the LaunchPad firmware with the application provided at processors.wiki.ti.com/index.php/MSP430_LaunchPad_Firmware_Update.
2. The touch-sense I/Os are not working when the LaunchPad is picked up.
Place the board flat on a table or other stable horizontal surface before using the Capacitive Touch BoosterPack. Do not hold the board while it is in use; the contacts on the back of the Capacitive Touch BoosterPack may be touched, which prevents the touch-sense I/Os from detecting user interaction.
3. My application is not able to light up all the LEDs.
The LEDs around the wheel are multiplexed; therefore, they cannot all be turned on simultaneously. Its required to use a time-shared approach to light up all LEDs at once (see [Section 3.1](#)).
4. The button or the wheel is sometimes fails to detect the first touch.
Make sure to wave your hand 3 to 5 cm above the board to wake the device before actually touch the wheel or button. The capacitive sensors are activated immediately after the wake-up sequence is finished.
5. Windows Media Player is not starting.
Loading Windows Media Player might take some time, depending on the system. On some systems, another media player program is associated with the Windows media keys.

7 References

The primary sources of MSP430 information are the device-specific data sheets and user's guides. The most up-to-date versions of the documents can be found at the Texas Instruments MSP430 web page. [1] All MSP430 LaunchPad and BoosterPack information can be found at the MSP430 LaunchPad wiki. [2]

The MSP430 LaunchPad and Value Line devices are supported in the latest versions of Code Composer Studio [3] and IAR Embedded Workbench [4]. In-depth details on the supported IDEs (CCS and IAR) can be found in the documentation folders of the IDE installation. IAR tool documentation (Workbench/C-SPY, the assembler, the C compiler, the linker, and the library) is in the common\doc and 430\doc folders. CCS documents is in the msp430\doc folder under the CCS installation path.

The FET user's guides [5] [6] also include detailed information on how to set up a project for the MSP430 using CCS or IAR Embedded Workbench. These user's guide are also included in the latest IDE releases.

1. <http://www.ti.com/msp430>
2. [http://processors.wiki.ti.com/index.php/MSP430_LaunchPad_\(MSP-EXP430G2\)](http://processors.wiki.ti.com/index.php/MSP430_LaunchPad_(MSP-EXP430G2))
3. http://processors.wiki.ti.com/index.php/Download_CCS
4. <http://focus.ti.com/docs/toolsw/folders/print/iar-kickstart.html>
5. *Code Composer Studio v4.2 for MSP430(tm) User's Guide* ([SLAU157](#))
6. *IAR Embedded Workbench Version 3+ for MSP430(tm) User's Guide* ([SLAU138](#))
7. Capacitive Touch Sense Library (CAPSENSELIBRARY) (<http://focus.ti.com/docs/toolsw/folders/print/capsenselibrary.html>)
8. *Capacitive Touch Sense Library User's Guide* ([SLAA490](#))
9. *Capacitive Touch Sense Library Quick Start Guide* ([SLAA491](#))
10. Capacitive Touch BoosterPack Software and Design Documentation (www.ti.com/lit/zip/slac490)

8 Schematics and PCB Layout

8.1 Schematics

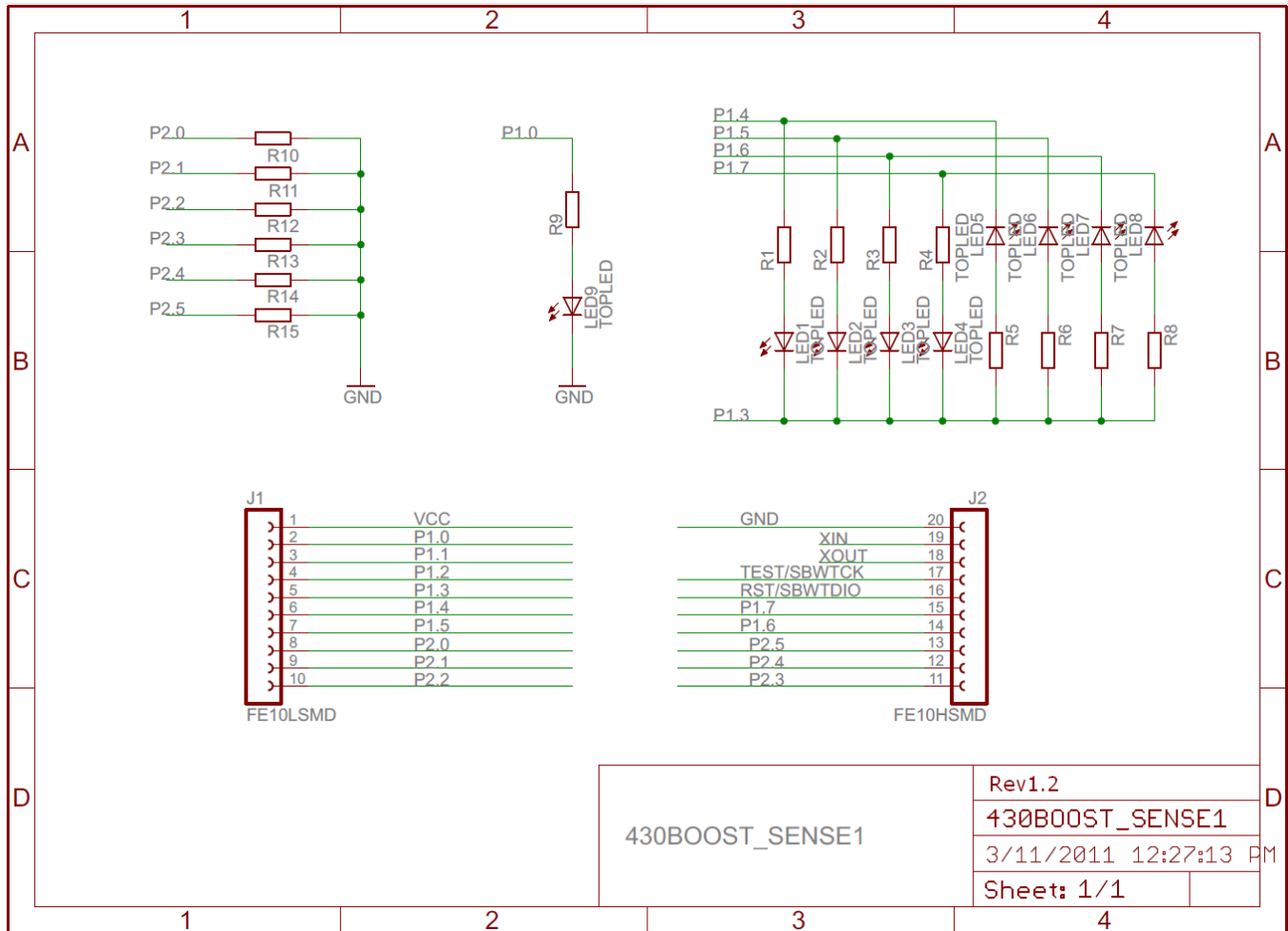


Figure 10. Capacitive Touch BoosterPack Schematic

8.2 PCB Layout

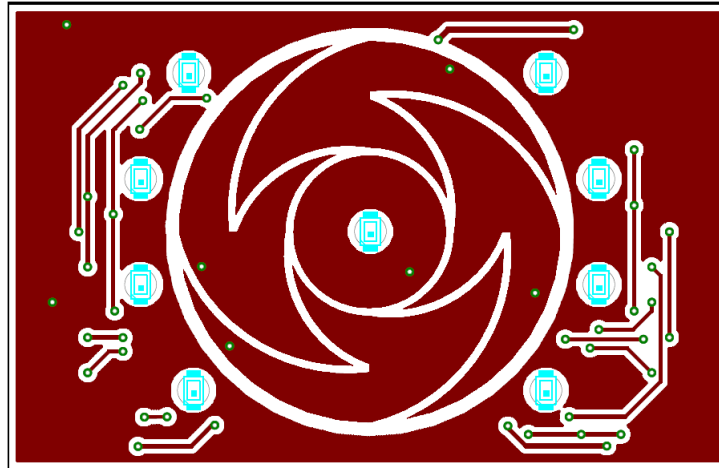


Figure 11. Capacitive Touch BoosterPack Layout, Top Layer

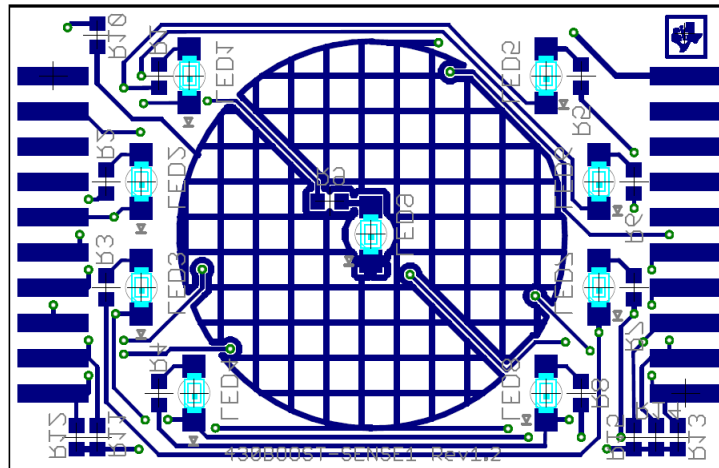


Figure 12. Capacitive Touch BoosterPack Layout, Bottom Layer and Silkscreen

8.3 Bill of Materials (BOM)

Table 2. Bill of Materials

Pos.	Ref Name	Number per Board	Description
1	R1 to R8	8	390-Ω SMD0603 resistor
2	R9	1	180-Ω SMD0603 resistor
3	LED1 to LED8	8	Top LED red wtr clr 631NM 1206
4	LED9	1	LED white round diffused 1206
5	J1, J2	2	Female header SSM-110-L-SV 2.54 mm
6	R10 to R15	0	SMD0603 resistor (not populated)

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

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

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI 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.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
RF/IF and ZigBee® Solutions	www.ti.com/lprf

Applications

Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Transportation and Automotive	www.ti.com/automotive
Video and Imaging	www.ti.com/video
Wireless	www.ti.com/wireless-apps

TI E2E Community Home Page

e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2011, Texas Instruments Incorporated