

430BOOST-SENSE1 – Capacitive Touch BoosterPack for the LaunchPad

User's Guide



Literature Number: SLAU337

March 2011

DRAFT ONLY

Preface	5
1 430BOOST-SENSE1 Overview	6
1.1 Overview	6
1.2 Kit Contents	7
2 Getting Started with 430BOOST-SENSE1 BoosterPack	7
2.1 Hardware Preparation	7
2.2 Software Preparation	8
2.3 Capacitive Touch User Experience	8
3 The Capacitive Touch BoosterPack Hardware	11
3.1 Driving the LEDs	13
3.2 Capacitive Touch sensors	14
4 LaunchPad Capacitive Touch BoosterPack User Experience Firmware	15
4.1 Project Import in CCS	15
4.2 Open project & workspace with IAR embedded workbench	15
4.3 Capacitive Touch Library	15
5 LaunchPad Capacitive Touch BoosterPack User Experience Software	17
5.1 LaunchPad Capacitive Touch BoosterPack User Experience GUI	17
5.2 MediaPad	17
5.3 UART Communication Protocol	17
6 Frequently Asked Questions (FAQ), Tips & Tricks	18
7 Reference	18
8 Appendix	19
8.1 Schematics	19
8.2 PCB Layout	20
8.3 Bill of Materials (BOM)	20

List of Figures

1	Capacitive Sense BoosterPack with LaunchPad	7
2	PC GUI is looking for LaunchPad.....	9
3	PC GUI in sleep mode	10
4	PC GUI in active mode	10
5	MediaPad	11
6	Capacitive Sense BoosterPack HW	12
7	Schematic LED multiplexing	13
8	LED driving	14
9	Capacitive Touch sensor areas.....	14
10	Capacitive Touch BoosterPack schematics.....	19
11	Capacitive Touch BoosterPack layout top layer.....	20
12	Capacitive Touch BoosterPack layout bottom layer and silkscreen	20

List of Tables

1	BoosterPack Interface	12
2	Bill of Materials.....	20

Read This First

If You Need Assistance

If you have any feedback or questions, support for the MSP430 device, the MSP-EXP430G2 LaunchPad and the 430BOOST-SENSE1 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 <http://focus.ti.com/docs/toolsw/folders/print/msp-exp430g2.html> or the LaunchPad wiki page at [http://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 is the first extension module for the MSP-EXP430G2 LaunchPad. These extension modules, designed specifically for the LaunchPad, are called BoosterPacks and each features an application example for one of the MSP430 ValueLine 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 ValueLine device.

This Capacitive Sense BoosterPack is designed to feature 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 sense and approximation sensing applications without the use of any external components. The BoosterPack includes a Capacitive Sense Board and an MSP430G2452 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 an application example on a Microsoft Windows PC.

430BOOST-SENSE1 features:

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

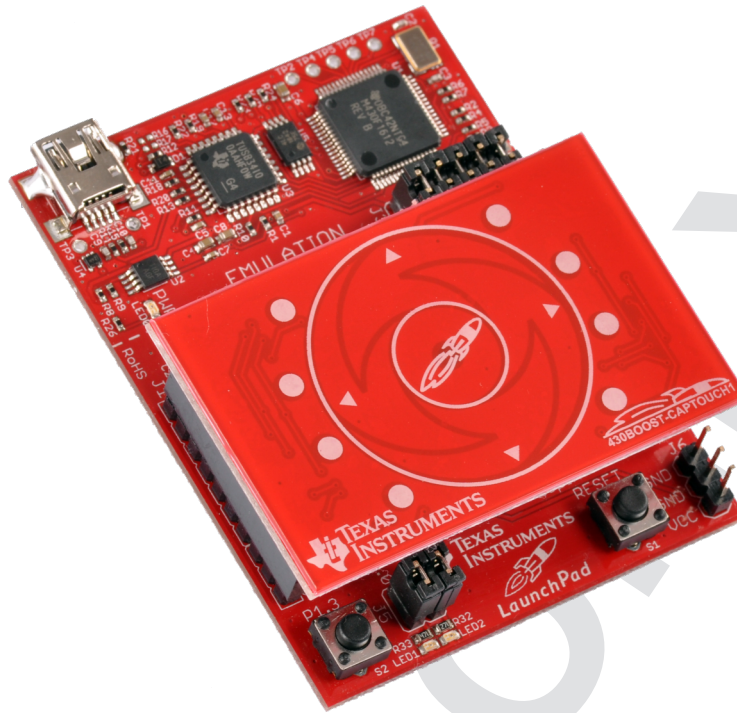


Figure 1. Capacitive Sense BoosterPack with LaunchPad

For latest information on the LaunchPad, other available BoosterPacks, software examples and how to program the included MSP430G2452 device take a look at the MSP-EXP430G2 User's Guide ([SLAU318](http://processors.wiki.ti.com/index.php/MSP430_LaunchPad_(MSP-EXP430G2))) or 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)).

1.2 Kit Contents

The 430BOOST-SENSE1 kit includes two components:

- The Capacitive Touch BoosterPack Board with 9 LEDs and 6 sensor areas
- A preprogrammed MSP430G2452 device
 - o Low-power 16Bit microcontroller with an 8-channel 10-bit ADC, Comparator, Universal Serial Interface supporting SPI and I2C, 8kB FLASH and 256B RAM memory

2 Getting Started with 430BOOST-SENSE1 BoosterPack

There are different preparations required to get the user experience application demo to work correctly and to start developing applications with the MSP430G2452 for the Capacitive Touch BoosterPack.

2.1 Hardware Preparation

Follow the described steps properly to ensure proper device operation.

1. Solder both 10-pin male headers on to the LaunchPad's breakout pin connections J1 and J2. These two 10-pin male headers as well as two 10-pin female headers come with the original LaunchPad kit.
Note: If the 10-pin female headers are already populated, simply 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 J5 connections to disconnect the LaunchPad LEDs and keep them from interfering with P1.0 & P1.6 functions of the Capacitive Sense BoosterPack.
3. Ensure jumpers VCC, TXD, & RXD of J3 connection are populated in order for the user experience

demo to operate properly.

Note: The Jumpers RST & TEST must also be populated when programming the device. They are not required for normal demo 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.
5. Plug the Capacitive Sense BoosterPack board in the LaunchPad with proper orientation by ensuring that the Texas Instruments logo and the texts on the BoosterPack are in the same direction with the texts and logo on the Launchpad.
6. Connect the LaunchPad with an USB cable to a PC or connect an external power supply (2.7V to 3.6V) to J6, the user experience demo application will light up the Center LED as soon as power is supplied to the board.

NOTE: The 32-kHz crystal/oscillator on Pin 18 and 19 is not required for the application to work.

2.2 Software Preparation

This step is 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 required the

All software necessary to run the User Experience demo can be found in the LaunchPad Capacitive Touch BoosterPack software package located 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)) Once installed all User Experience application demos are stores inside the Software folder within the selected installation directory.

2.2.1 LaunchPad USB Driver

In order for the PC to communicate to the hardware, the LaunchPad's 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 is 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 can be both located under [INSTALL_PATH]\Software. All software and firmware examples are also available as source code within 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] and 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

This part applies for both stand-alone demo (no PC required) and 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, USB battery pack, etc.) 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 hand or finger ~3-5cm above the BoosterPack in order to trigger the proximity sensor and to wake up 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. Firmly press any position on the wheel or the center button to perform a touch. Make sure to keep finger fully and 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. Slide along the wheel without releasing finger from the wheel to perform a gesture. The corresponding LEDs trace & follow the touch & gesture.
5. After a period 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 part describes instructions to run the application on both of the LaunchPad Capacitive Touch BoosterPack and the PC. Hence it assumes that the hardware is connected to the PC via USB cable [Step 1 of [Section 2.3.1](#)] and proper software installation [[Section 2.2](#)].

The steps described below also correspond to instructions described in [Section 2.3.1](#).

1. Start the CapTouch_BoosterPack_UserExperience_GUI.exe application located at [INSTALL_PATH]\Software\CapTouch_BoosterPack_UserExperience_GUI\

Upon startup, the GUI 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. The GUI will continue as normal whenever a LaunchPad is plugged into the PC.



Figure 2. PC GUI is looking for LaunchPad

2. Upon start-up or after long period of inactivity, the device goes into sleep mode. The GUI grays out accordingly to indicate such inactivity. Upon proximity sensor detection (wave hand ~3-5cm above the BoosterPack) the device returns to active mode and triggers the GUI to wake up as well.

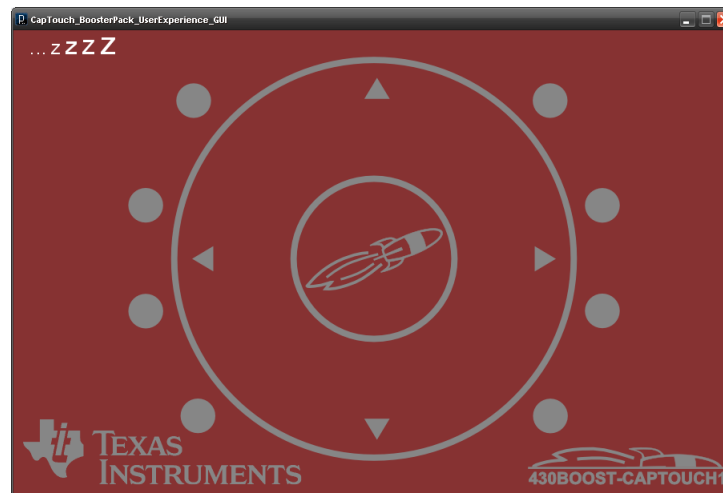


Figure 3. PC GUI in sleep mode

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 left upper corner of the PC GUI.
4. The gesture tracking [Start, Stop, Update] is visualized on the wheel with the coloration of the wheel slices. Gesture can be tracked for several revolutions of the wheel, in both clockwise and counter-clockwise directions.

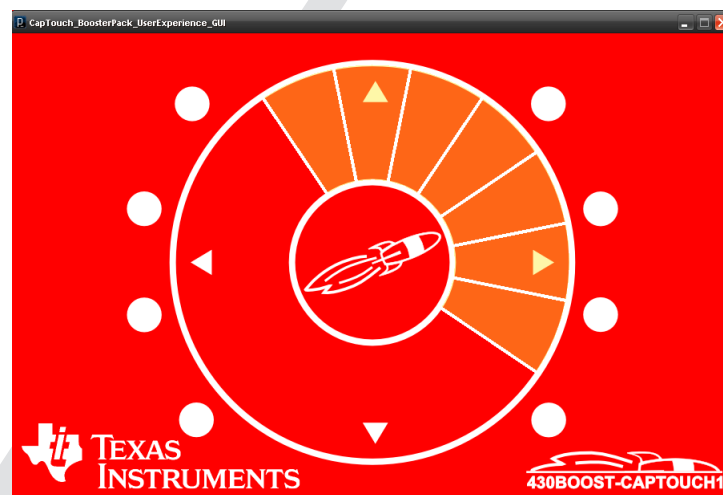


Figure 4. PC GUI in active mode

5. After a period of no capacitive touch activity, the board returns to sleep mode. The GUI also grays out and returns to 'sleep mode' as well.
* Embedded within the wheel, a hidden mode can be unlocked. Correct sequence [similar to a rotational combination lock] reveals a secret.

2.3.3 MediaPad

This part describes instructions to run the application on both of the LaunchPad Capacitive Touch BoosterPack and the PC. Hence it assumes that the hardware is connected to the PC via USB cable [Step 1 of 2.3.1] and proper software installation [Step 2.2].

Start the MediaPad.exe application located at [INSTALL_PATH]\Software\MediaPad\

Upon execution, the application starts looking 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 which then exits the application. If a LaunchPad COM port is found, the application prompts a greeting start-up message. Once the message is closed, the application minimizes itself to the taskbar.

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.

The following touches or gestures can be used for media control in a Windows system.

1. Center button press: Start Media Player [Windows Media Player by default]**
2. Bottom arrow button press: Play/Pause
3. Left arrow button press: Previous Track
4. Right arrow button press: Next Track
5. Scroll wheel clockwise: Volume Up
6. Scroll wheel counter-clockwise: Volume Down

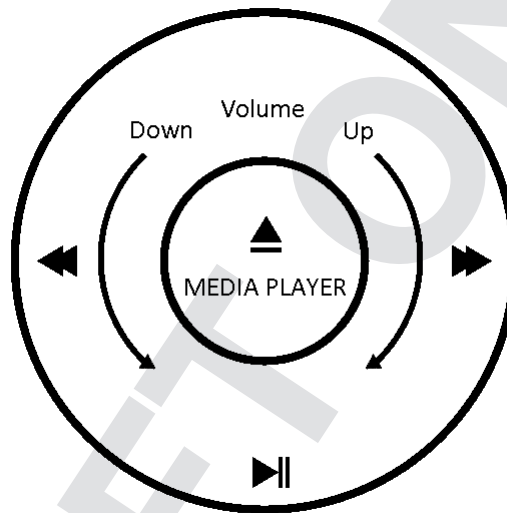


Figure 5. MediaPad

Note: The Microsoft .NET runtime library might be required in order to execute the MediaPad software. While most recent Windows PC come pre-installed with .Net runtime library, a new/re- install of the library might be necessary.

3 The Capacitive Touch BoosterPack Hardware

The Capacitive Touch BoosterPack Board is showing a typical Capacitive Touch application example for the new MSP430G2xx2 Touch-sense enabled I/O Pins. The board is showing three different capacitive sensor types, a single button right in the middle, a wheel made up from 4 single capacitive sensors and a proximity sensor all around the PCB. Additionally there are 9 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 much of the microcontrollers I/O Pins. This way only 5 pins are use to drive all 8 LEDs by a timeshared signal.

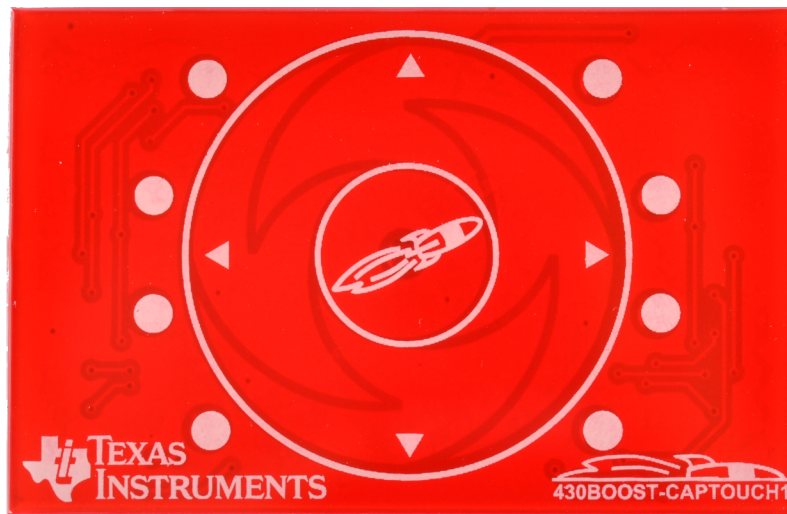


Figure 6. Capacitive Sense BoosterPack HW

The Capacitive Touch BoosterPack Board is showing a typical Capacitive Touch application example for the new MSP430G2xx2 Touch-sense enabled I/O Pins. The board is showing three different capacitive sensor types, a single button right in the middle, a wheel made up from 4 single capacitive sensors and a proximity sensor all around the PCB. On top of the PCB there is a 0.8mm thick layer of acrylic to insulate the PCB like its used on most Capacitive Touch applications. Additionally there are 9 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 much of the microcontrollers I/O Pins. This way only 5 pins are use to drive all 8 LEDs by a timeshared signal.

Table 1. BoosterPack Interface

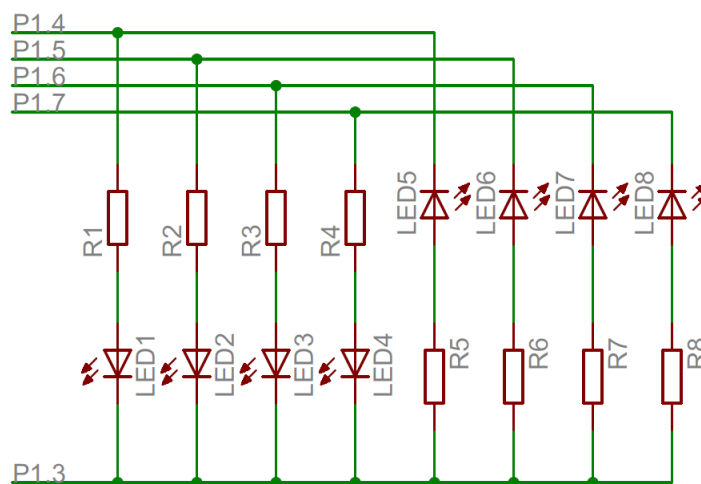
Pin	MSP430 Port	BoosterPack Signal	Description
1	VCC	NC	Supply voltage, not Connected to BoosterPack
2	P1.0	LED9	The white center LED
3	P1.1/TXD	NC	Backchannel UART transmit data output, not connected to BoosterPack
4	P1.2/RXD	NC	Backchannel UART receive data input, not connected to BoosterPack
5	P1.3	LEDx	LED base to drive the 8 multiplexed LEDs
6	P1.4	LED1	LED1 positive LED5 negative drive
7	P1.5	LED2	LED2 positive 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 LED7 negative drive
15	P1.7	LED4	LED4 positive LED8 negative drive

Table 1. BoosterPack Interface (continued)

Pin	MSP430 Port	BoosterPack Signal	Description
16	RST/SBWTIO	NC	Reset line for SBW JTAG data, not connected to BoosterPack
17	TEST/SBWTCK	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 ValueLine device and can be set by just setting this port as output. The other 8 LEDs are multiplexed as seen on [Figure 7](#). 4 LEDs are connected with the diodes cathode to the ports P1.4 to P1.7 the other 4 are connected with the diodes anode. All LEDs are connected to port P1.3 to either source or drain 4 LEDs at a time. To drive a particular set of these 8 LEDs two steps are required at first the port P1.3 acts as GND drain. The first nibble can then drive the LEDs 1 to 4 directly the other LEDs are not active during that time. The next step will light the other LEDs 5 to 8 by switching port P1.3 to a VCC source, the pattern for the second nibble then has to be inverted and set to port P1.4 to P1.7.


Figure 7. Schematic LED multiplexing

The switching between both 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's recommended to set the non driven LED signals to input mode. The Figure 4 is showing the signals of all the LED driving Pins required to light up LED1,3,6 and 7. It doesn't matter if the mode on P1.3 is set prior 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 timeshared approach to drive the LEDs due to the shared CPU and Timer resources utilized by the Capacitive Touch Library functions & the UART transmissions.

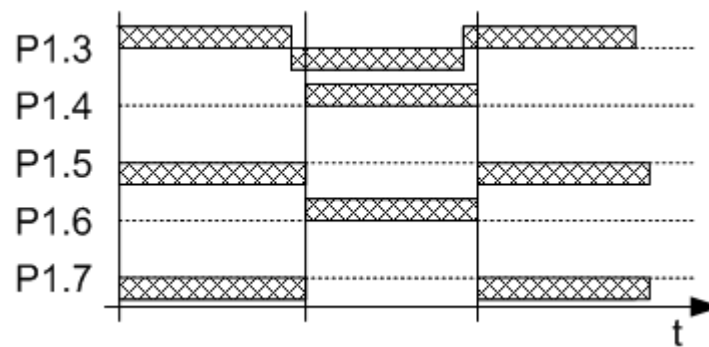


Figure 8. LED driving

3.2 Capacitive Touch sensors

The 6 different Capacitive Touch sensor areas are connected to the port 2 of the device. The pins on port 2 of the MSP430 ValueLine devices with enabled Touch-sense I/Os have no analog functionality and a smaller internal capacitance, which make these ports more sensitive than Capacitive Touch sensors on the port 1. The Capacitive Touch BoosterPack hardware can also be used with other MSP430s the Resistors R10 to R15 have to be populated with a resistor to realize a RC discharge which can be measured with a timer.

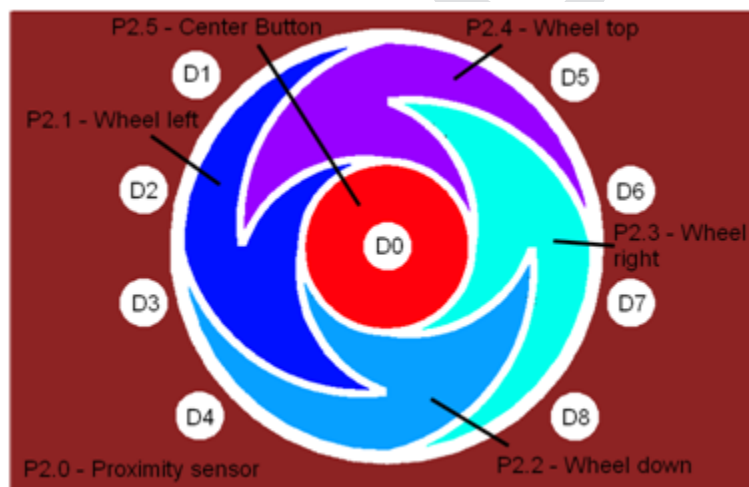


Figure 9. Capacitive Touch sensor areas

The [Figure 9](#) is showing the connection of the Capacitive Touch sensor areas to the MSP430.

The Touch-sense feature of the I/Os can be enabled with setting the secondary port select PxSEL2 and clearing of the PxSEL bit. The selected pins will start oscillating right away and the frequency gives a direct indication about the capacitance connected to the port pin. The Touch-sense I/Os are oscillating within a frequency range from 1 to 2 MHz, which is strongly dependant by the supply voltage, the device package and some environmental influences. For more information about the Touch-sense feature of the new MSP430 ValueLine devices, some code snippets or applications examples visit 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 take a look at the MSP430 Capacitive Touch Library, also used in the application demo included in this kit.

4 LaunchPad Capacitive Touch BoosterPack User Experience Firmware

This chapter describes the firmware application that is provided with the project. Detailed information on the project was constructed, the usage of the Capacitive Touch Library, and how to set up and import the projects are also described. The source code for the MSP430G2452 firmware application can be located under [INSTALL_PATH]\Source\ as described in [Section 2.2](#).

This 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 new to the MSP430G2xx2 family devices. The User Experience application also utilizes the Capacitive Touch Library to realize & measure the capacitive touch and proximity sensors. The Capacitive Touch Library also provides layers of abstractions to generate higher logical outputs such as logical touches and their position (in this hardware, a four-button wheel).

The User Experience application starts up and remains in 'sleep' mode, sampling the proximity sensor every ~8.3ms [$VLO/100=12kHz/100=120Hz$]. Upon registering a valid proximity event (hand/finger/object hovering ~3-5cm from the BoosterPack), the application wakes up to operate in the 'active' mode. During the wake up period, the LEDs surrounding the wheel light in a wake-up sequence, starting with a slow clockwise and ending with a fast counter-clockwise motion. 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 as well as simple gestures [Clockwise & Counter-clockwise] while 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 predetermined amount of 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 SW TimerA to provide application and capacitive sensing data to the PC via the UART-USB back channel. The application sends UART data upon events such as wake up, sleep, touch, or gesture.

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

4.1 Project Import in CCS

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

*Ideally, workspace should be in completely independent folder, not containing or contained by the project/package folder.

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 & workspace with IAR embedded workbench

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

4.3 Capacitive Touch Library

The Capacitive Touch Library 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 very 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 and therefore an RO implementation is chosen, where 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 TimerA0 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 GPIO are defined first and then once the appropriate sensor characteristics are defined 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, but 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 125KHz while for the button and wheel the frequency is increased to 1Mhz. The interval count is 8192: 8.192ms gate time for the button and wheel elements and 65.5ms for the proximity element. The wheel is a special kind of sensor where each element contributes to the sensor performance. The wheel is made up of 4 elements, divided into 64 points or sections, and requires that the cumulative response exceed 75 percent. This last point 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 since the desired output is simply a binary indication if the threshold is exceeded or not. 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 was a little more complicated, as several measurements are required at various positions. Please refer to section "Sensor arrays: Wheels and Sliders" in SLAA490 for a more detailed explanation.

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

3. API Calls

There are 5 different API functions which are called several times throughout the application.

- The TI_CAPT_Init_Baseline and TI_CAPT_Update_Baseline functions are used to initialize and update 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 user experience code the initialization and updates are performed at the beginning after a power up sequence and also 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 is used to measure the proximity sensor. The variable dCnt is updated with the measured value. In the application, this variable is compared to a threshold value. Since 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. Once 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 is used to determine if the middle button has been detected. This is a fairly straightforward function returning either a 0 or 1 to indicate a threshold crossing (touch). The middle LED is illuminated to indicate a touch.
- The TI_CAPT_Wheel function indicates the position on the wheel if touched or simply returns a defined illegal 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 8 LEDs around the board.

For more information on the library, refer to the Capacitive Touch Library documentation. ([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 said information in the GUI. Processing is a platform independent open source programming language and environment, specialized on 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 9600baud UART connection and starts receiving data.

The GUI processes event and capacitive touch data and visualizes said information 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 the Getting Started chapter (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 Requirement

- Processing download from <http://www.processing.org>
- serial library [included with processing installation]
- pitaru.sonia_v2_9 sound library: <http://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 the LaunchPad Capacitive Touch BoosterPack into Microsoft Windows virtual keystrokes for Windows media control. The application implements auto-detection code that automatically seeks for a LaunchPad-compatible USB COM port before establishing the proper connection. Further behavior of the application is described in the Getting Started chapter (2.3.3) and the ReadMe.txt in the project source code directory.

5.2.1 Requirements

Microsoft Visual C++ 2010 Redistributable Package (included in any version of Visual Studio 2010). (available at <http://www.microsoft.com/downloads>)

5.3 UART Communication Protocol

Per each event (wake up, go to sleep, touch/press, or gesture), a UART packet of two bytes is sent via the Application UART back channel of the LaunchPad. They 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 0x0-0xF, one nibble]
- GESTURE START: 0xFC 0x2z [z = touch position 0x0-0xF, one nibble]
- GESTURE STOP: 0xFB 0xFB
- GESTURE & GESTURE END POSITION : 0xGG 0x2z GG = [binary] b????????
GG: MSB is direction: 0 = clockwise, 1 = counter-clockwise; last 7 bits = count of gesture movement
z = ending position of the immediate gesture, 0x0-0xF, one nibble.

The PC applications can receive and decipher the UART information to translate them into appropriate actions.

6 Frequently Asked Questions (FAQ), Tips & Tricks

1. The LaunchPad is unable to program the MSP430G2452.
Some of the revision 1.4 LaunchPads have problems with supporting the MSP430G2452 devices. Please update the LaunchPad firmware with the application provided here http://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 board flat on the table before using the Capacitive Touch BoosterPack. Do not hold the sides of the board while in use. The contacts on the back of the Capacitive Touch BoosterPack are otherwise touched and the Touch-sense I/Os are not able to detect user interaction.
3. My application is not able to light up all the LEDs?
The LEDs around the wheel are multiplexed, hence they can't all be turned on simultaneously. Its required to use a timeshared approach to light up all LEDs at one. See chapter [Section 3.1](#).
4. The button or the wheel is sometimes not detecting the first touch.
Make sure to wave hand to wake up the device before actually touch the wheel or button. The capacitive sensors are activated right after the wake up sequence is finished.
5. Windows 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 Reference

The primary sources of MSP430 information are the device-specific data sheets and user's guides. The most up-to-date versions of those documents can be found at the Texas Instruments MSP430 page or the MSP430 LaunchPad wiki.

www.ti.com/msp430, [http://processors.wiki.ti.com/index.php/MSP430_LaunchPad_\(MSP-EXP430G2\)](http://processors.wiki.ti.com/index.php/MSP430_LaunchPad_(MSP-EXP430G2))

For in-depth knowledge on the supporting IDEs such as CCS and IAR, download the latest version from the web pages above and refer to included user's guides and documentation inside the installation folder. Documents describing the IAR tools (Workbench/C-SPY, the assembler, the C compiler, the linker, and the library) are located in common\doc and 430\doc. All necessary CCS documents can be found inside the msp430\doc inside the CCS installation path.

The FET user's guide also includes detailed information on how to set up a project for the MSP430 using IAR or CCS, and it is included in most of the IDE releases and on the TI MSP430 side.

8 Appendix

8.1 Schematics

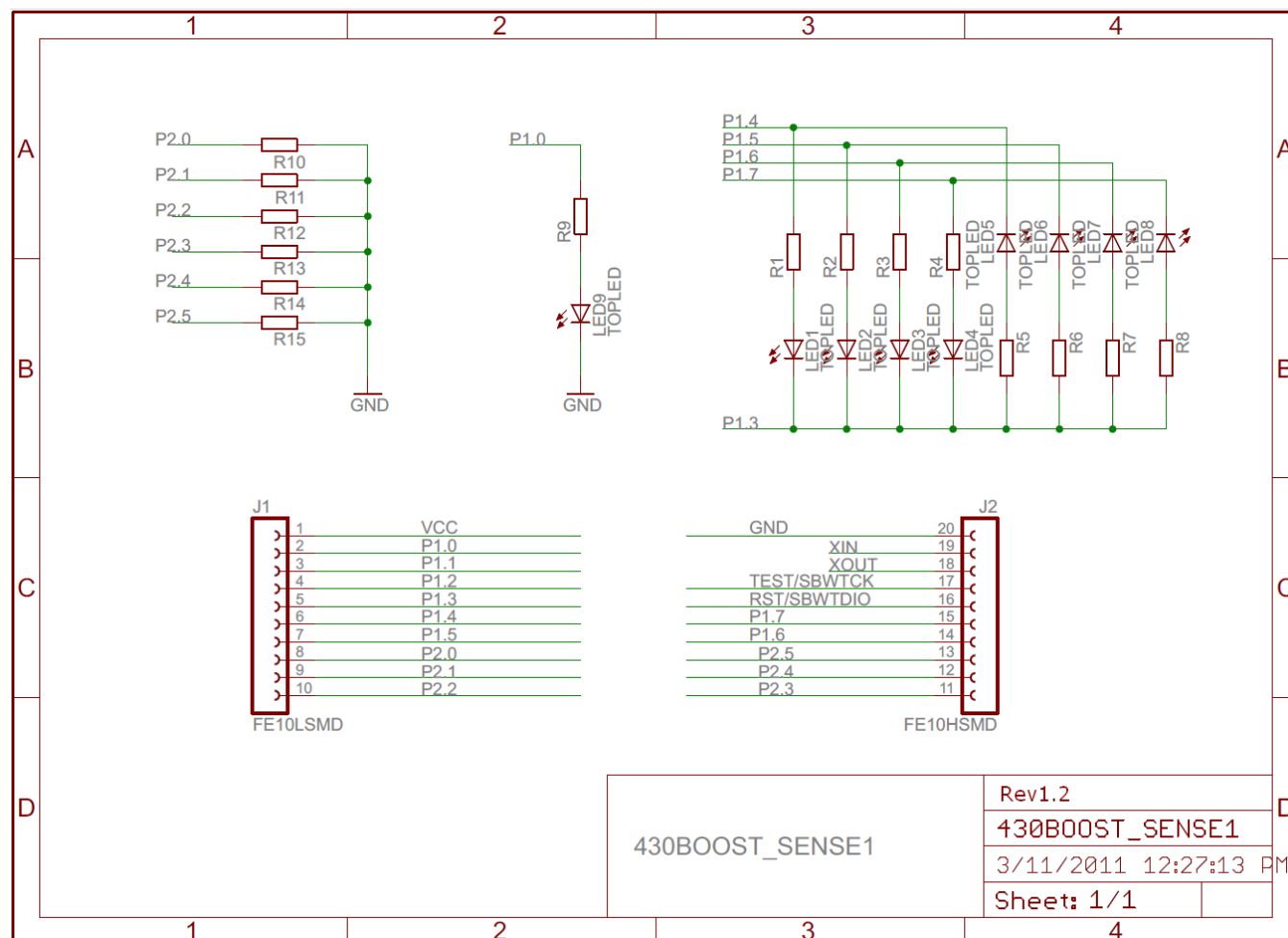


Figure 10. Capacitive Touch BoosterPack schematics

