TI Designs: TIDM-02004
Gesture-Based Capacitive Touch Speaker Interface Reference Design

Description

The Gesture-Based Capacitive Touch Speaker Interface Reference Design demonstrates an ultra-low-power panel for controlling speaker systems. The design uses an MSP430™ microcontroller (MCU) with CapTIvate™ technology for recognizing capacitive touch events. Using a self-capacitance configuration of sensors to detect proximity, tap, drag, and swipe gestures, this design can enhance a speaker system's interface and improve the user experience. The MSP430FR2633 MCU drives all capacitive sensing functionality.

Resources

TIDM-02004	Design Folder
MSP430FR2633	Product Folder
CAPT-MSP-FR2633 Development Kit	Tool Folder
MSP-EXP430F5529LP LaunchPad Development Kit	Tool Folder
CapTIvate Design Center	Tool Folder

TI E2E™ Community

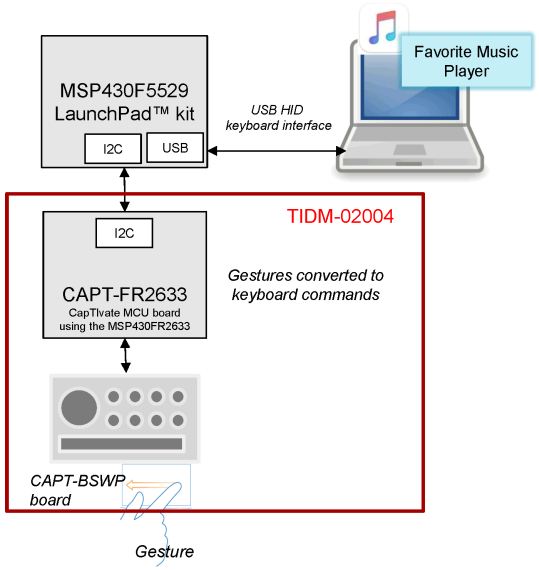
ASK Our E2E Experts
WEBENCH® Calculator Tools

Features

- MSP430 CapTIvate MCU for capacitive touch sensing
- Capacitive touch button gestures: tap, tap and hold
- Capacitive touch slider and wheel gestures: tap, double tap, swipe and slide
- Low power consumption: <5 µA while idle using wake-on-proximity, <500 µA in active mode
- Moisture tolerant design: tolerance to moisture spray and droplets in certain environments, following recommended design guidelines

Applications

- Smart Speakers (With Voice Assistant)
- Soundbars
- Wired Speakers
- Wireless Speakers



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 Description

Capacitive touch button, slider and wheel human-machine interfaces can create an enhanced user experience over traditional mechanical style buttons, allowing sleek designs and improved product reliability. This reference design for wireless smart speakers demonstrates how capacitive touch inputs using simple finger gestures, such as tap, double tap and swipe can be used to control a Windows® media player.

1.1 Key System Specifications

Table 1 lists the key system specifications of the TIDM-02004.

Table 1. Key System Specifications

Feature	Specification	Details
Button	8 buttons, self capacitive	
Slider	4 element, self capacitive, 140 mm (5.5 in), resolution = 1000 counts	User programmable
Wheel	3 element, self capacitive, 35 mm (1.5 in) diameter, resolution = 100 counts	User programmable
Gesture (wheel)	Tap, swipe left or right, slide left or right	See Section 3.2.1
Gesture (slider)	Tap, double tap, swipe left or right, slide left or right	See Section 3.2.2
Gesture (button)	Tap, tap and hold	See Section 3.2.3
Sensor scan rate	20 ms (50 Hz)	User programmable
Power Consumption	Active mode: approximately **480 µA**	See Section 3.7.2
	Low-power wake on proximity: approximately **5 µA**	
Moisture tolerance	Design is tolerant to condensation, fine spray	See Section 3.7.3
Featured MCU: MSP430FR2633	The MSP430FR2633 is a low-power FRAM MCU with integrated CapTIvate technology for capacitive sensing	See Section 1.3
MSP430FR2633 memory footprint		See Section 3.6.3

1.2 Introduction

The TIDM-02004 design demonstrates the use of capacitive touch gestures to control the play, pause, volume up and down, and next and previous track functions of the Windows Media Player application. Compared with traditional mechanical style buttons, which are limited to only simple interactions, capacitive touch technology opens up new possibilities for product design, enhancing user experience, improving product reliability and reducing product cost.

The focus of this reference design is to show how combining intelligent gesturing firmware with a capacitive touch sensor design can create new possibilities for human-machine inputs. This reference design, along with the accompanying firmware examples, provide the reader a path to implementing similar functionality in their designs. What is not covered in detail in this reference design are details on USB, HID and interfacing to a Windows environment.

1.3 MSP430FR2633 CapTIvate MCU

The MSP430FR2633 is an ultra-low-power, FRAM-based MSP430 MCU featuring CapTIvate Technology. CapTIvate Technology is TI's robust capacitive sensing solution. The MSP430FR2633 MCU is designed for user interface applications with integrated capacitive touch and a strong MSP430 peripheral set.

Features:

- 16 capacitive touch inputs that can support up to 64 electrodes in mutual-capacitance mode
- Parallel scanning of up to four electrodes at a time
- CapTIvate Software Library included in a preprogrammed 12KB of ROM
- Four 16-bit timers and a 16-bit counter-only real-time clock (RTC)
- Three enhanced serial communications peripherals for UART, IrDA, SPI, and I^2C
- 19 I/Os with 16 interrupt pins for wake-up from low-power modes

- High-performance, 8-channel, 10-bit analog-to-digital converter (ADC)
- Clock system with an operating speed of up to 16 MHz

2 System Overview

2.1 Block Diagram

Figure 1 shows how the TIDM-02004 demonstrates capacitive touch gestures to control a Windows media player. The reference design block diagram includes the CAPTIVATE-FR2633 Target MCU module, CAPTIVATE-BSWP self capacitive demo panel and MSP430FR2633 MCU.

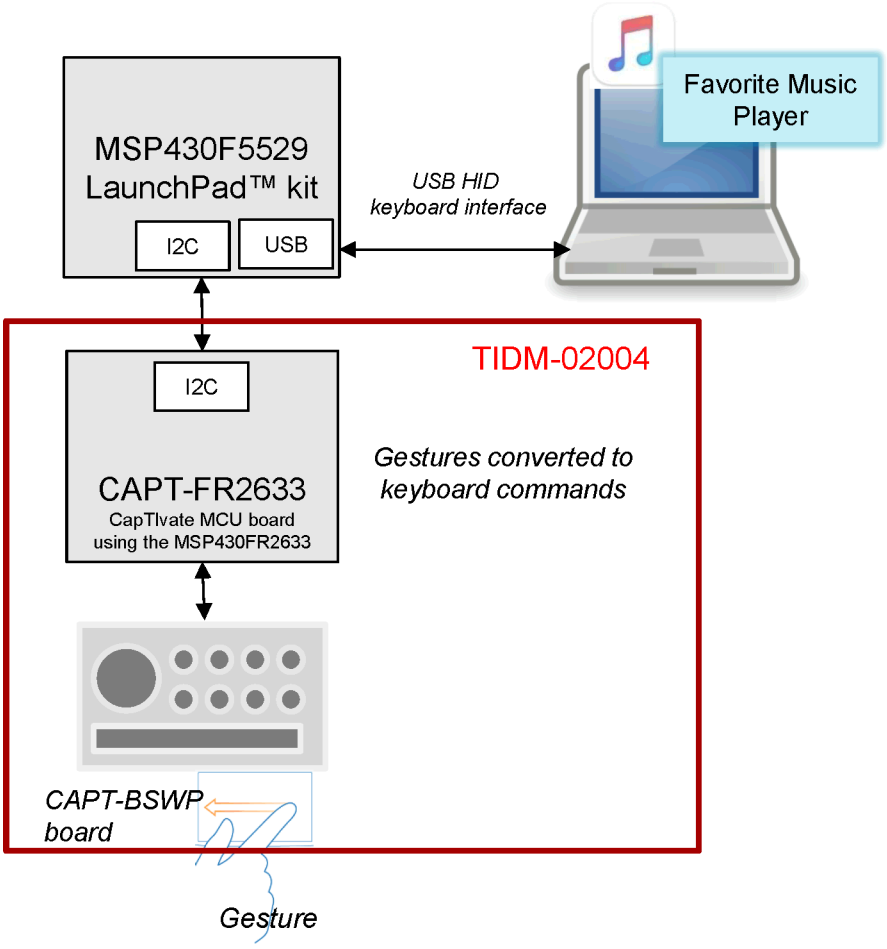


Figure 1. TIDM-02004 Block Diagram

2.2 Design Considerations

This reference design can be easily assembled from existing TI products available on the TI e-store.

CAPT-MSP-FR2633 CapTIvate Capacitive Touch development kit

EXP-MSP430F5529LP USB LaunchPad development kit

To experiment further with capacitive touch and designing a custom PCB, refer to the CapTIvate Technology Guide for guidelines on button, slider, and wheel sensor designs.

2.3 Highlighted Products

MSP430FR2633 Target MCU

The MSP430FR2633 is a 16-bit microcontroller with programmable ferroelectric memory (FRAM) and CapTIvate capacitive sensing technology. CapTIvate technology is a flexible and robust capacitive sensing technology for user interface applications such as buttons, sliders, wheels, and proximity sensors.

CAPTIVATE-PGMR Board

The CAPTIVATE-PGMR board is used to program and debug the MSP430FR2633 target MCU. It is included in the CAPT-MSP-FR2633 development kit or can be ordered separately. For this reference design, the CAPTIVATE-PGMR is needed only to initially program the MSP430FR2633 target MCU with the demonstration firmware. It is not needed for the demonstration.

2.3.1 MSP430FR2633 MCU Block Diagram

MSP430FR2633 MCUs feature a diverse peripheral set that makes them ideal for use in many capacitive sensing applications. Figure 2 shows the block diagram of the MSP430FR2633 MCU.

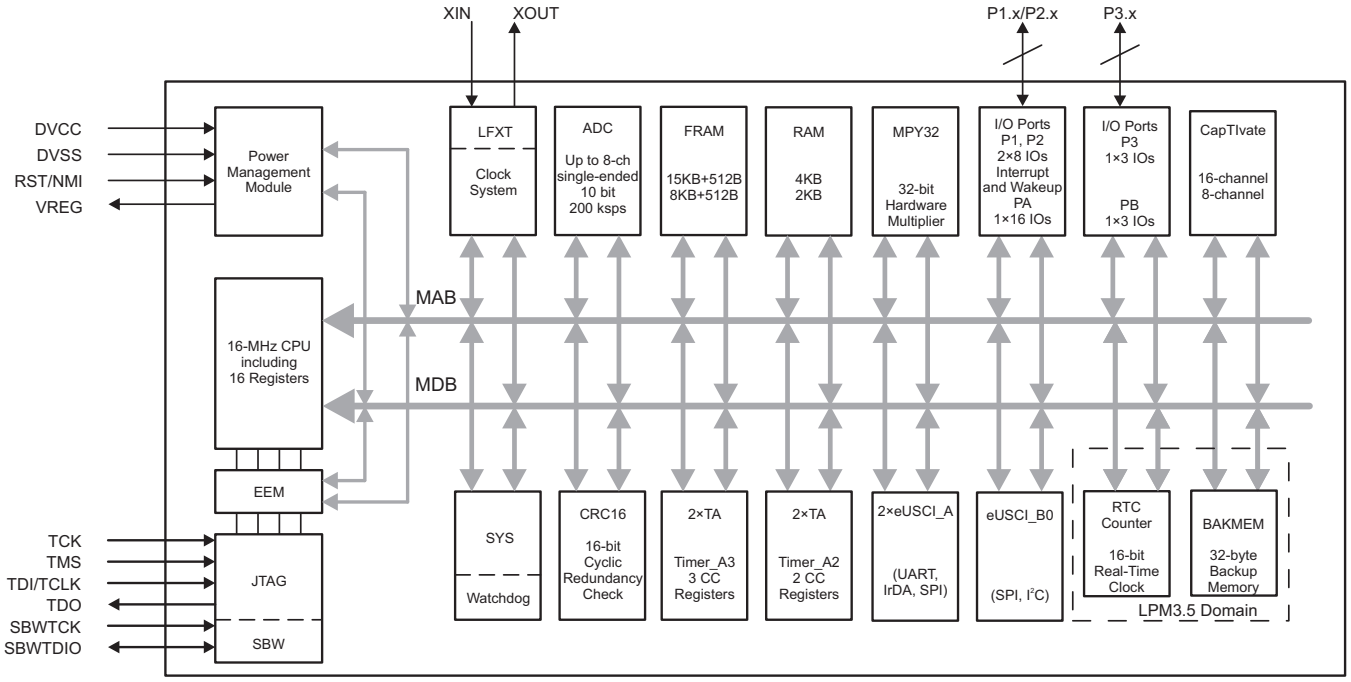


Figure 2. Block Diagram of MSP430FR2633

2.3.2 CapTIvate Technology Block Diagram

CapTIvate Technology enables capacitive sensing on the TIDM-02004. CapTIvate Technology is an MSP peripheral dedicated to providing robust capacitive sensing measurements. Figure 3 shows the block diagram of the CapTIvate peripheral.

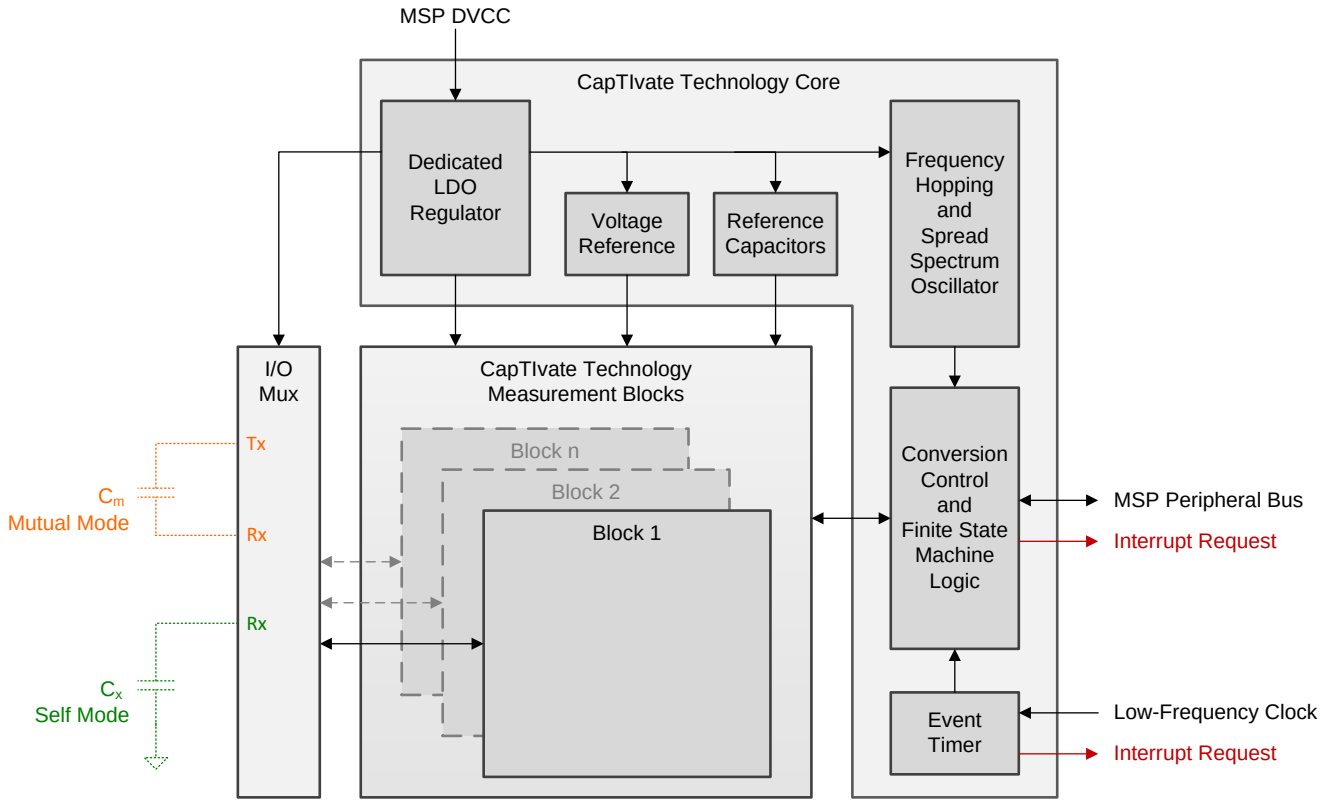


Figure 3. CapTIvate Technology Block Diagram

3 Hardware, Software, Testing Requirements, and Test Results

3.1 System Design Theory

The TIDM-02004 reference design leverages the CapTIvate CAPT-MSP-FR2633 capacitive touch development kit featuring an MSP430FR2633 MCU with CapTIvate Capacitive Touch Technology and the CAPTIVATE-BSWP demonstration panel with self capacitive button, slider and wheel sensors created from copper patterns designed on the PCB. As a user's finger interacts with these sensors, changes in capacitance are measured by the MSP430FR2633 MCU and processed to determine the validity and type of gesture. Each valid gesture is then mapped to a corresponding keyboard value and transmitted to the MSP430 USB HID keyboard device over an I2C interface where it is reported to the Windows application.

3.2 Capacitive Touch Sensors

Compared with traditional mechanical buttons, capacitive touch wheel, slider and button sensors offer a product designer more flexibility and functionality when considering an application's human-machine interface design.

For detailed information about capacitive touch sensors and how to design them, please refer to the CapTIvate Technology Guide, Design Guide chapter for guidelines on button, slider and wheel sensor designs.

3.2.1 Wheel Sensor

A capacitive touch wheel sensor is constructed as a circular copper pattern on the PCB containing 3 or 4 interdigitated sensing elements. The interdigitation provides a high resolution, linear output as the finger moves around the wheel . A wheel's basic functionality reports a finger position anywhere on the sensor. By adding some gesturing intelligence to the firmware, both finger direction and speed can also be derived.

Specific regions or zones on the wheel's surface can be defined or mapped in software creating a flexible user-interface that allows various application functions to be controlled depending where the wheel is touched. Incorporating motion gesture features, such as Slide or Swipe, adds another level of user action possibilities. Figure 4 shows some of the possible gesture features that could be used for controlling the functionality on a wireless smart speaker.

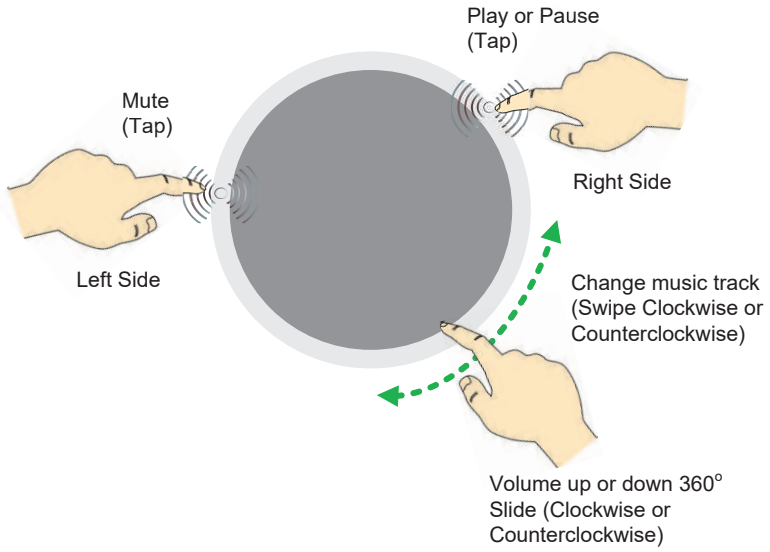


Figure 4. Wheel Sensor Gesture Examples

One additional benefit of the capacitive touch wheel is the ability to re-purpose the wheel interface hardware design from product to product and change only features or functionality as needed in firmware.

3.2.2 Slider Sensor

A capacitive touch slider sensor shares many of the same characteristics as the wheel sensor. It is constructed as a linear copper pattern on the PCB containing 3 or 4 interdigitated sensing elements. And, like the wheel, the slider provides a high resolution linear output as the finger moves from left to right and reports a finger position anywhere on the sensor.

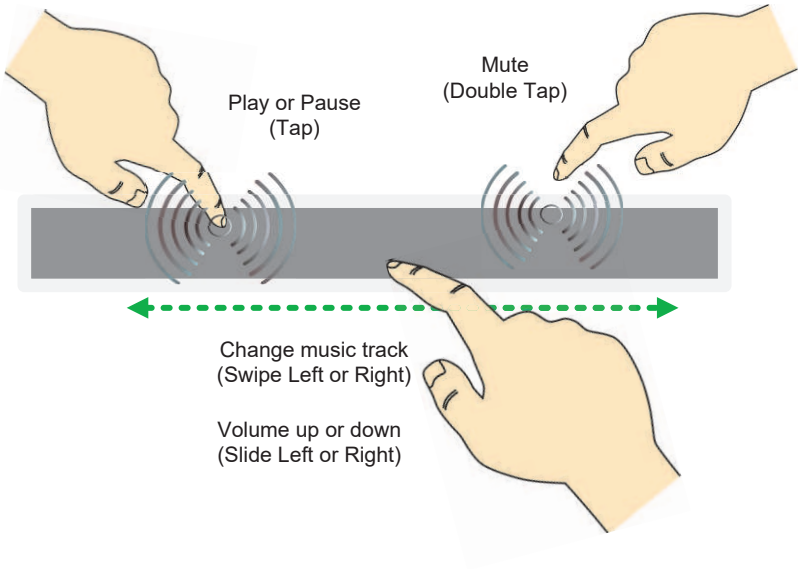


Figure 5. Slider Sensor Gesture Example

3.2.3 Button Sensor

A capacitive touch button sensor is a simple sensor constructed as a filled circular copper pattern on the PCB, slightly larger than the diameter of a finger. Unlike a wheel or slider sensor, the button is limited to providing a binary output, such as on/off. But unlike it's mechanical counterpart, implementing simple gesture features, such as tap-and-hold or Double-Tap can provide the additional functionality for the application.

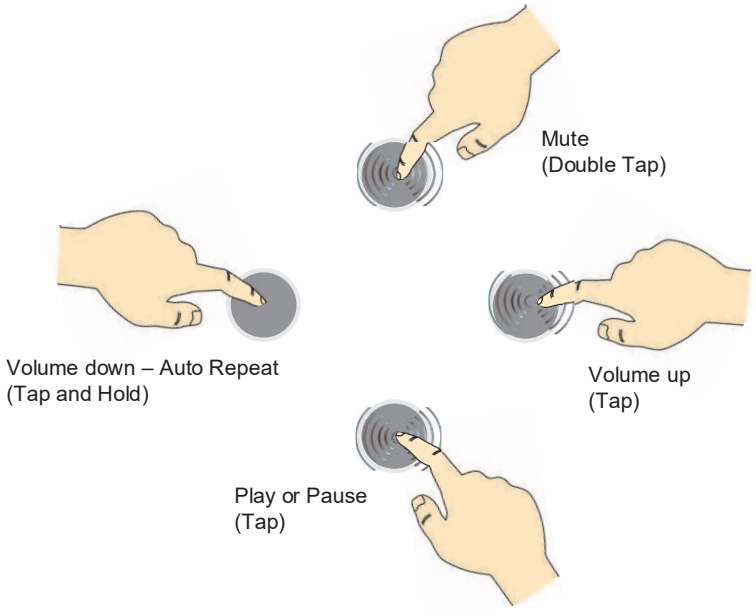


Figure 6. Multi-Button Sensor Gesture Example

3.3 Gesture and Key Mapping

This section provides details about the mapping between a gesture and a specific media player function used in this reference design demonstration.

3.3.1 Media Player

In this Windows media player demonstration, gestures such as *swipe right*, for example, are mapped in the firmware to the corresponding media player function that selects the next media track. Table 2 lists the seven supported gestures and their corresponding actions with equivalent keyboard key combinations.

Table 2. Media Player Gesture Mapping

Media Player Function	Key Combination	Button Gesture	Slider Gesture	Wheel Gesture
Play or pause	CTRL-P	Tap	Tap	Tap
Previous music track	CTRL-B	Tap	Swipe left	Swipe counterclockwise
Next music track	CTRL-F	Tap	Swipe right	Swipe clockwise
Mute	F7	Tap	Double tap	Tap
Volume up	F8	Tap or tap-and-hold	Slide right	Slide clockwise
Volume down	F9	Tap or tap-and-hold	Slide left	Slide counterclockwise

3.3.2 USB HID Keyboard Data

Each keyboard key value is defined in the USB HID Usage Tables, Keyboard/Keypad Page (0x07), available from USB-IF. Typing these same keys or key combinations directly on the Windows keyboard controls the media player the same as the capacitive touch demo.

As shown in USB HID Usage document, each key is represented by an 8-bit value. For example, reporting the F7 key requires an 8-bit value (0x00 to 0xFF). Reporting a key-combination, such as CTRL-P requires two bytes, because two keys are being pressed simultaneously. Table 3 lists each keyboard key's 8-bit value.

Table 3. Key Values From USB HID Usage Table for Keyboard/Keypad Page (0x07)

Keyboard Key	Key Value
KEY-B	0x05
KEY-F	0x09
KEY-P	0x13
KEY-F7	0x40
KEY-F8	0x41
KEY-F9	0x42
KEY-MOD-LCTRL	0xE0

The key value is transmitted by the MSP430FR2633 to the MSP430 USB HID device over an I^2C interface. When transmitting the key value, the key or key combination is reported as a 16-bit value, in which the lower byte represents the primary key and upper byte represents the key modifier, if applicable. If a key has no modifier, then the upper byte is 0x00. Table 4 lists the final 16-bit key representations that are reported to the MSP430 USB HID device.

Table 4. Media Player Functions

Media Player Function	Key Combination	16-Bit Representation
Play or pause	CTRL-P	0xE013
Previous music track	CTRL-B	0xE005
Next music track	CTRL-F	0xE009
Mute	F7	0x0040
Volume up	F8	0x0041
Volume down	F9	0x0042

3.4 Capacitive Touch Gestures

Adding touch gesturing to a capacitive touch wheel, slider, or button sensor enables a more flexible human-machine interface (HMI) compared to a traditional mechanical button operation in which a button is either on or off. For example, with a capacitive touch wheel, a simple finger tap anywhere on the wheel sensor can represent a button press. Multiple touch zones can be mapped on the wheel, providing the equivalent of having several discrete buttons. In addition, the wheel can detect the motion of a finger to control user inputs like a volume control or can detect a rapid swipe motion to provide some other type of control input. Of course, moving from a traditional discrete on-off button functionality to a more feature-capable wheel does require some intelligence to translate these actions and motions into the desired functionality. This intelligence is gesture detection.

The following sections describe the gestures, motion parameters, and timing diagrams for each of the supported gestures. For more detailed information about gesture software and tuning, refer to Capacitive Touch Gesture Software and Tuning.

3.4.1 How Gesture Detection Works

To understand how gesture detection works, first look at how a gesture is detected. The MSP430FR2633 measures change in capacitance on the CAPTIVATE-BSWP panel caused by a finger touch on a button, slider, or wheel sensor. The CapTIvate library firmware includes algorithms that determine if a sensor is touched and, if the sensor is a slider or wheel sensor, the position of the finger on the sensor. The MSP430FR2633 has a dedicated 16-bit CapTIvate timer that is set by default to generate a periodic capacitive touch measurement interrupt every 20 milliseconds, or 50 times a second. This rate is user configurable. During each interrupt, the three sensors are measured, followed by the gesture processing, which uses the periodic sensor sampling rate as the time base measurement for finger touch and release events. Because the CapTIvate technology has a dedicated timer, none of the general-purpose 16-timers on the MCU are used, leaving them available for the main application.

NOTE: Important Concept: Gesture timing is based on the sensor sampling rate.

To determine a specific gesture there are one or two attributes needed. The first attribute is time. This is measured by counting the number of sensor measurement samples between two events, such as a finger touch followed by release. For example, when sampling a sensor every 20 milliseconds, a touch that lasts 10 sample periods represents a touch for 200 milliseconds. The time attribute applies to buttons, sliders, and wheels. The second attribute is distance. This is the distance that a finger has moved and applies to only wheel and slider sensors.

By assigning parameters to these time and distance attributes, rules can be created to help define each gesture. Why is this important? Because gesture duration and speed can vary from user to user, gesture parameters help improve gesture repeatability and detection accuracy. Each sensor type has its own gesture parameters and is configurable in software to allow a specific "user touch and feel" to be tailored for the application.

Because each sensor can have different gesture behaviors, processing is specific to each sensor type. For example, in this reference design there are wheel gestures, slider gestures and button gestures assigned to the corresponding sensors. In software, the sensor gesture is essentially a state machine that is executed on every measurement sample as part of the sensor's callback function and uses the sensor's timing and distance parameters to control the processing.

Figure 7 shows that a tap gesture is a momentary touch followed by release and is common to all three sensor types. Notice that the gesture detection only begins after there is a continuous touch for at least the first four samples and is only reported as a tap gesture when the finger is released within the specified time window. If released outside of the time window, the gesture is ignored.

NOTE: The specific time and distance parameter values in the following figures represent the parameters used for this reference design demonstration. These parameters are user configurable and can be tuned for any application.

3.4.2 Tap Gesture

A tap gesture is defined as a momentary finger touch followed by a release (see Figure 7). The touch duration window is set by the minimum and maximum limits defined for the gesture. The software parameters are TouchSampleCount_Min and TouchSampleCount_Max.

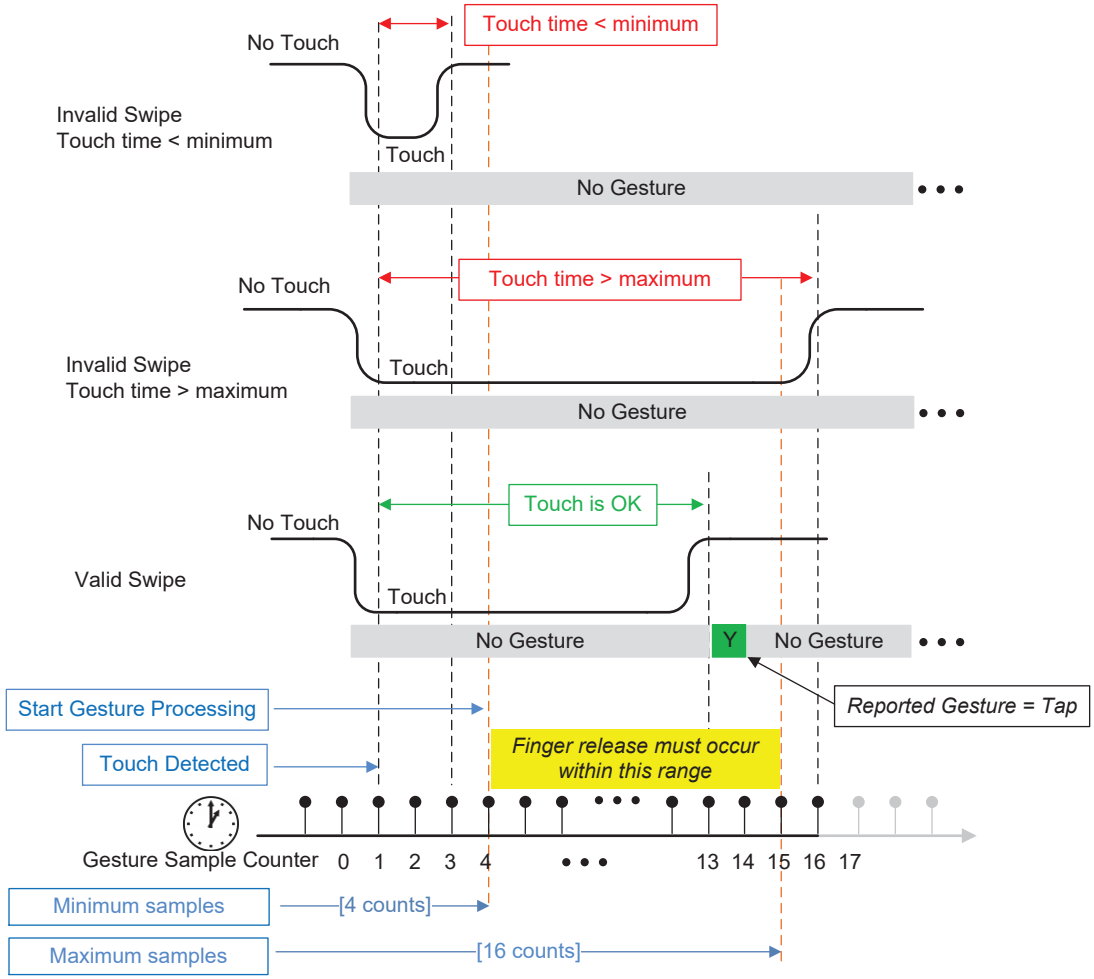


Figure 7. Tap Gesture Detection

3.4.3 Double-Tap Gesture

A double-tap gesture is defined as a momentary finger touch followed by a release, then a second touch and release. Figure 8 shows that the double-tap gesture is essentially a tap followed by a delay that defines a window when the second touch can occur. The software parameters are TouchSampleCount_Min, TouchSampleCount_Max and DoubleTapDelayCount_Max.

NOTE: A sensor that supports double-tap also supports tap by default. The reader must be aware that if a tap and not a double-tap is detected, the tap gesture is not reported until after the double-tap delay period expires. This can create a slight delay in tap response, but when properly tuned, the delay has a minimal impact on user interaction.

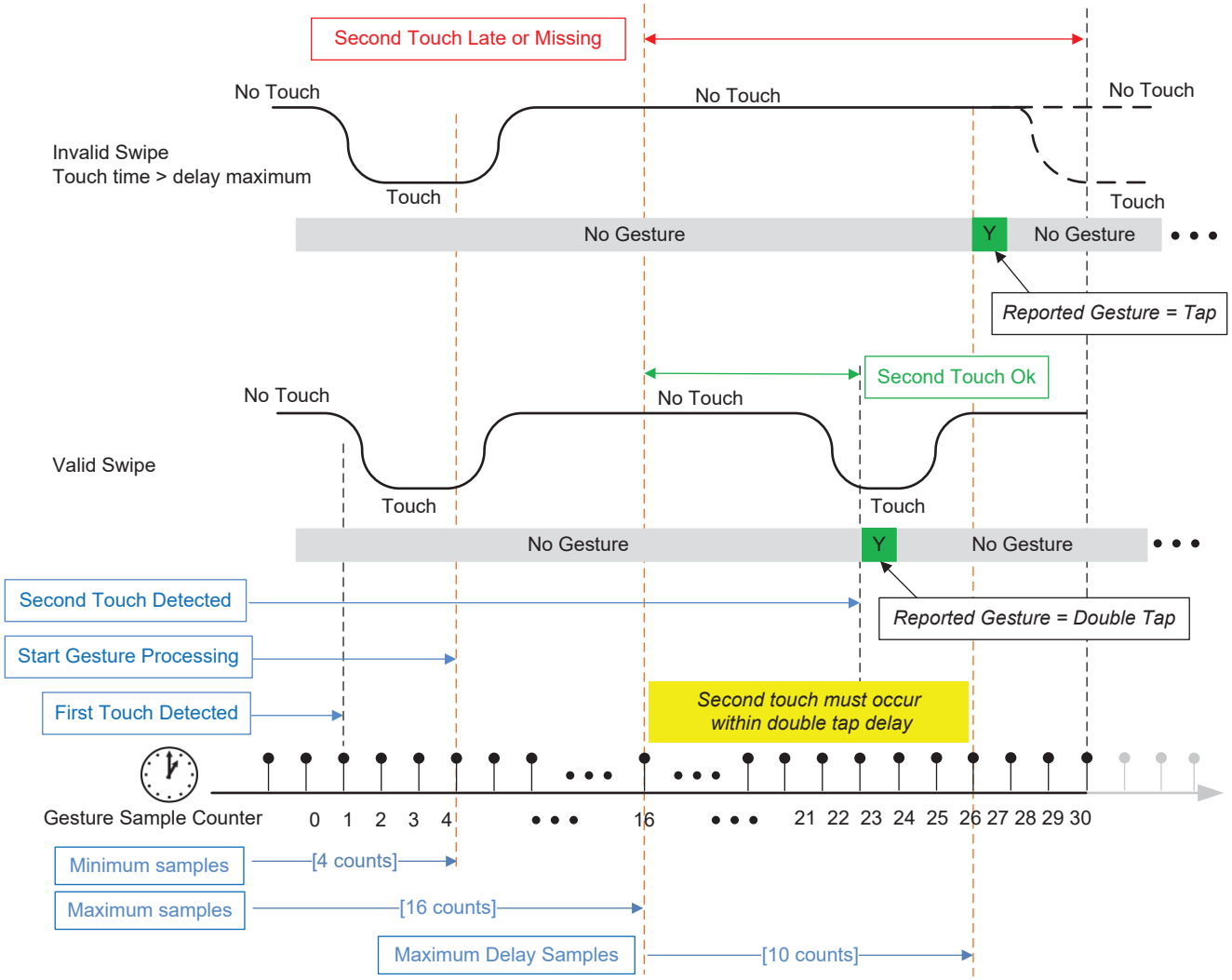


Figure 8. Double Tap Gesture Detection

3.4.4 Tap-and-Hold Gesture

A tap-and-hold gesture is defined as a long finger touch followed by a release (see Figure 9). Just like a tap, the tap-and-hold has a min limit and a min hold limit that defines how long the finger must remain touching to qualify as a valid tap-and-hold gesture. The software parameters are TouchSampleCount_Min and TouchHoldSampleCount_Min.

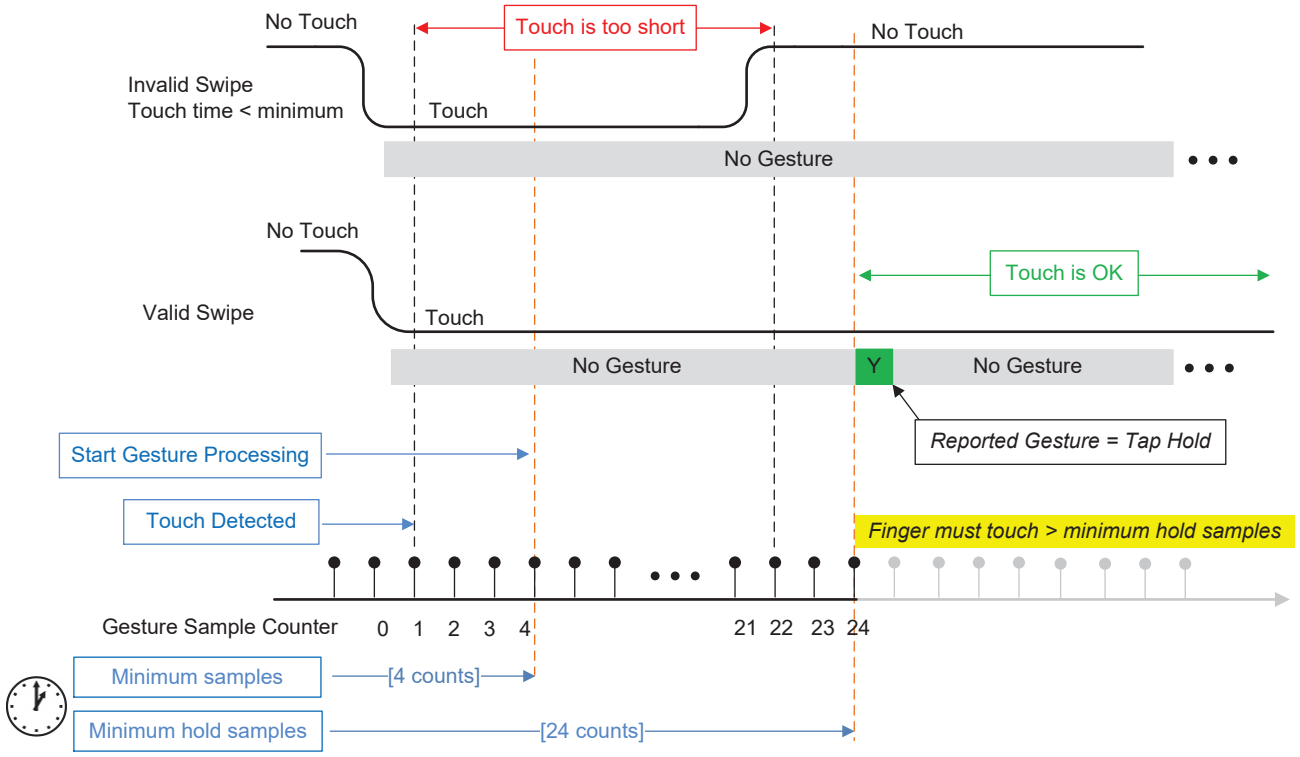


Figure 9. Tap-and-Hold Gesture Detection

3.4.5 Swipe Gesture

A swipe gesture is defined as a brief finger touch with the finger moving, first a minimum distance to show it is a motion gesture, then the finger must move a minimum swipe distance then release the finger within the allowed time. Figure 10 shows that the swipe gesture requires both time and distance parameters. The software parameters are TouchSampleCount_Min, SwipeSampleCount_Max, FingerDistance_Min, and SwipeDistance_Min.

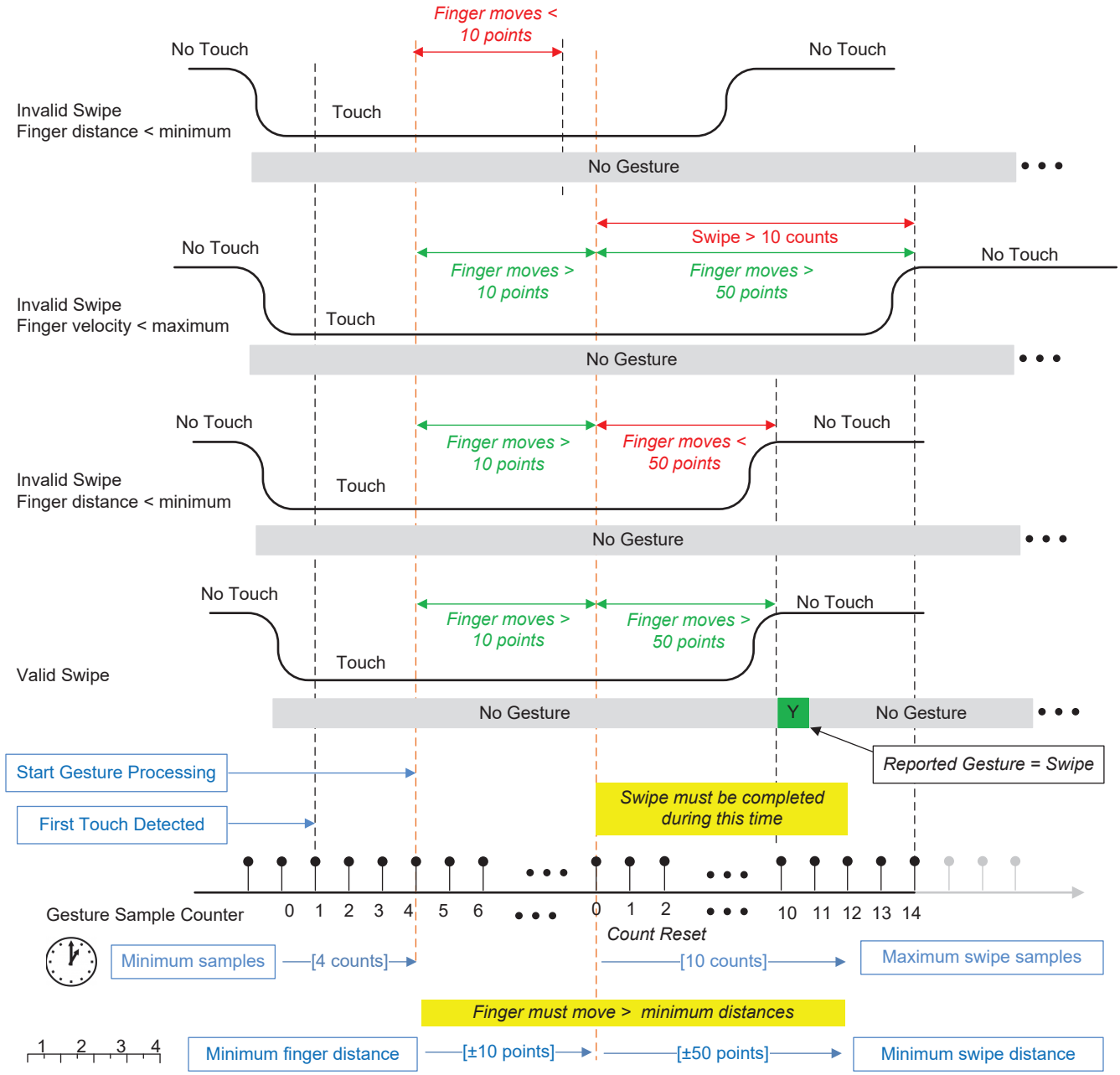


Figure 10. Swipe Gesture Detection

3.4.6 Slide Gesture

A slide gesture behavior is similar to a swipe gesture, but is defined as a continuous finger touch with motion where the distance moved meets the minimum distance parameter. Figure 11 shows that as long as the finger is touching the sensor and moves the minimum required distance in any direction, a slide gesture is reported. The software parameters are TouchSampleCount_Min and SlideStepSize.

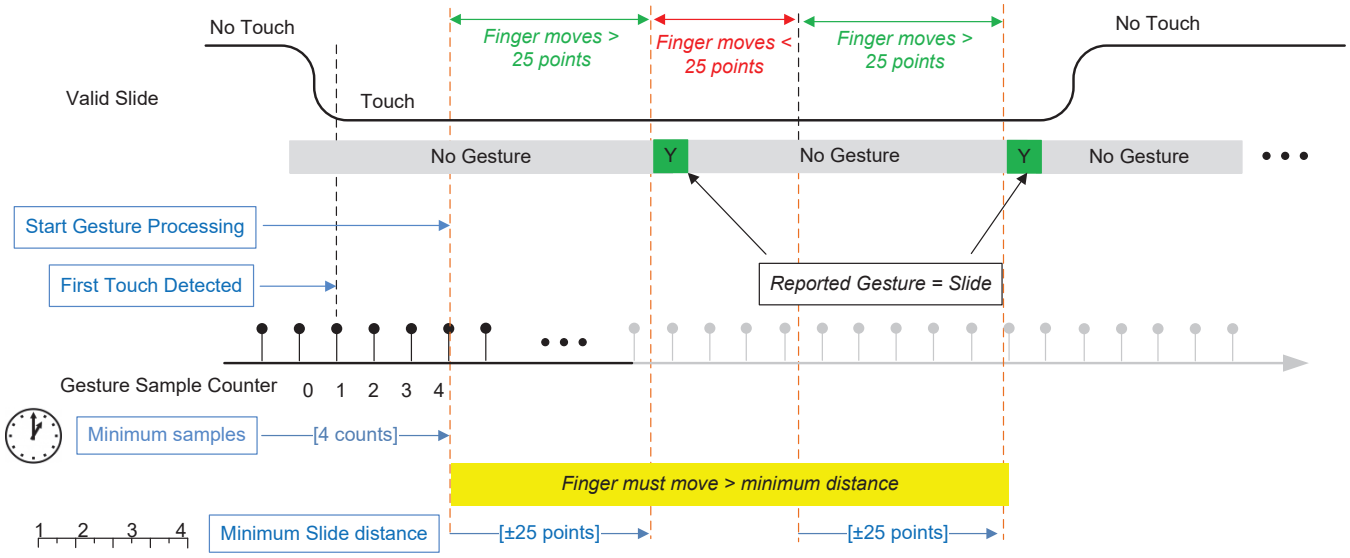


Figure 11. Slide Gesture Detection

3.5 Getting Started With Hardware

The required hardware is described in Section 2.2. If the target MCUs have not been programmed with the demonstration firmware, follow the procedure in Section 3.6. After programming, connect the MSP430FR2633 and MSP430F5529 LaunchPad development kits together and connect to a Windows PC using a USB cable. The MSP430F5529 should enumerate as a HID interface device automatically. No drivers are required.

3.6 Getting Started With Firmware

The example firmware developed for the TIDM-02004 was developed using CCS v8.2.0.00007 and TI Compiler version 18.1.4 LTS. To evaluate the example firmware, download the latest version of CCS. The example projects can be imported into a CCS workspace from [TI Design Software Install Root]/Firmware/Source/*.

3.6.1 Programming Target MCU

To program the MSP430FR2633, import the CAPT-FR2633-MediaPlayerDemo project (see Section 3.6.2) and compile the software. Connect a CAPT-PGMR programmer PCB to the CAPTIVATE-FR2633 MCU target board, then connect a USB cable to the CAPT-PGMR. In CCS, program the target by clicking on the small green bug icon. When complete, remove the CAPT-PGMR board.

To program the MSP430F5229 LaunchPad development kit, import the MSP-EXP430F5529LP_USBKeyboardHost_GestureDemo project (see Section 3.6.4) and compile the software. Connect a USB cable to the MSP430F5529 LaunchPad development kit. In CCS, program the target by clicking on the small green bug icon.

3.6.2 MSP430FR2633 CapTIvate Gesture Firmware

The MSP430FR2633 example project can be imported into a CCS workspace from [TI Design Software Install Root]/TIDM-02004_firmware/demo_src/CAPT-FR2633_MediaPlayer.

This firmware example is based on the CAPTIVATE-BSWP demo provided with the CapTIvate Design Center installation.

The source code for the MSP430FR2633 that is relevant to this demo is organized in multiple files (see Table 5).

Table 5. Source Code Files for Demo Application

Name	Description
main.c	Application main function
demo\media_player.c, demo\media_player.h	Demo initialization and typdefs header file
gestures\gesture_definitions.h	Defines which sensors are included in the build
gestures\gestures.h	Function prototypes, enumerations, and gesture data structure types
gestures\media_player_buttons.c	Button sensor gesture processing function
gestures\media_player_wheel.c	Wheel sensor gesture processing function
gestures\media_player_slider.c	Slider sensor gesture processing function

3.6.3 Memory Footprint

The demo firmware for this reference design has the largest memory footprint because it supports all three sensors. A more typical application would have one or maybe two sensor types and therefore the memory footprint would be smaller.

Table 6 lists some possible configurations to give the reader a sense of how the memory size varies the different sensors and features. Table 7 lists the program memory specific to the gesture type and its handler. These values are included in the memory sizes in Table 6.

Table 6. Memory Size Combinations

Sensors	FRAM (bytes)	RAM (bytes)	Configuration
Slider, wheel, and buttons	7852	1340	Default demo configuration
Slider, wheel, and buttons	7426	1338	Low-power wake on proximity disabled
Slider, wheel, and buttons	9242	1722	GUI UART communications enabled
Slider	6488	742	
Wheel	6522	710	
Buttons	6410	870	

Table 7. Individual Sensor Gesture Related Program Memory Sizes

Sensor	Handler (bytes)	Gesture (bytes)	Total (bytes)
Slider	160	404	564
Wheel	160	454	614
Buttons	160	312	472

Compiler = TI v18.1.4 LTS, Optimization level = -03. Unless otherwise stated, the reported code sizes are with the GUI communications option disabled = CAPT_INTERFACE (__CAPT_NONE_INTERFACE__) located in CAPT_UserConfig.h.

3.6.4 MSP430F5529 HID Keyboard Device Firmware

The MSP430F5529USB HID example project can be imported into a CCS workspace from [TI Design Software Install Root]/TIDM-02004_firmware/hostmcu_demo_src/MSP-EXP430F5529LP_USBkeyboardHost.

The source code for the MSP430F5529 that is relevant to this demo is organized in multiple files in Table 8.

This firmware example is based on the USB HID H8_Keyboard example firmware that is available from TI Resource Explorer.

Table 8. Source Code Files for Demo Application

Name	Description
main.c	Application main function
keyboard.c	Functions to handler key processing and reporting
keyboard.h	Function prototypes, enumerations and data structure types for keyboard

3.7 Testing

This section describes the testing that was performed on the TIDM-02004. It includes descriptions of test setups and results.

3.7.1 Gesture

Using the demonstration instructions from Section 4 to verify each sensor's operation.

Table 9. Gesture Test

Test Conditions	Result
Test all gesture combinations on wheel, slider, and button sensors. The default sensor sampling rate is 20 ms (50 Hz).	Pass

3.7.2 Power Measurements

The MSP430FR2633 operates in Active mode while the user is touching any sensor on the CAPT-BSWP board. The measured current averages approximately 480 µA in this mode.

When the sensors are no longer being touched, the CPU enters low power mode LPM3 indefinitely and the CapTIvate peripheral is configured to operate in wake-on-proximity mode. In this mode, the CapTIvate peripheral is only looking for the presence of a finger or hand, waking the CPU if detected. Operating in this wake-on-proximity mode, the CPU can achieve very low power. The measured current averages approximately 5 µA in this mode.

Two methods are available to measure the MSP430FR2633 power consumption.

1. Remove jumper J3 on the MSP430FR2633 MCU board and connect a current meter between jumper J3 "3.3V LDO" pin and J3 "MCU VCC" pin.
2. Connect the CAPT-PGMR board and use the programmer's Energy Trace™ feature to measure the current. Note, on the MSP430FR2633 MCU board, move jumper J3 between jumper J3 "MCU VCC" pin and J3 "3.3V Metered" pin.

For either option, return jumper J3 to the original position when done.

NOTE: The MSP430F5529 LaunchPad development kit must be disconnected from the MSP430FR2633 MCU board during the measurement. The MSP430FR2633 MCU board continues to respond normally to finger gestures.

For more information about current measurements on the MSP430FR2633, refer to the Experiments with Low Power section of the CapTIvate™ Technology Guide.

Test Conditions (Method 1)	Result
Active mode	480 µA
Low power, Wake on Proximity	5 µA

3.7.3 Moisture Tolerance

The moisture tolerance test is performed with the CAPT-BSWP panel in the horizontal position sitting on a bench top. All electronics, including the 48-pin connector on the CAPT-BSWP board are covered with plastic. A spray bottle set to fine mist and pumped three times from a distance no closer than 30 cm (12 in). The CAPT-BSWP panel is not designed for maximum moisture tolerance, and it can be susceptible to false triggers under certain conditions. For information about moisture tolerant designs, see the TIDM-1021 Liquid Tolerant Capacitive Touch Keypad Reference Design.

Table 10. Moisture Tolerance Test

Test Conditions	Results
Spray bottle set to fine mist	No false triggers detected

4 Demonstration

The following description assumes that the target MCUs have been programmed and the MSP430FR2633 and MSP430F5529 LaunchPad development kits have been connected together and connected to a Windows PC using a USB cable. The MSP430F5529 should enumerate as a HID interface device automatically. No drivers are required.

NOTE: During this demonstration, leave the mouse focus on the media player application. Failing to do so allows keyboard commands to be sent to other open applications and can result in undesired behavior.

When power is applied to the TIDM-02004 reference design hardware, LED1 and LED2 located on the MSP430FR2633 MCU board will turn on momentarily, then turn off. Touching the slider, wheel, or buttons causes LED1 to turn on and remain on as long as the finger remains touching. When a valid gesture is detected, LED2 toggles on and off, indicating the gesture is valid and the corresponding key value has been sent to the Windows application.

Launch the Windows Media Player application on the PC. Select a favorite song and mouse click to get the song started.

4.1 Wheel

4.1.1 Volume Control

With a song playing, touch anywhere on the wheel and slide your finger clockwise to increase the volume and counter-clockwise to decrease the volume. Experiment with with various starting positions on the wheel. You should see that this function works independent of where you start the gesture.

4.1.2 Changing Music Tracks

With a song playing, provide a swipe anywhere on the wheel in a clockwise direction to select the next song. Swipe in a counter-clockwise to select the previous song. Experiment with the starting position. You should see this function works independent of where you start the gesture.

NOTE: The wheel resolution is 100 points. The swipe gesture requires that your finger move a minimum distance of 10 points.

It is also possible to select the next or previous songs with a tap gesture on the regions shown in Figure 12.

4.1.2.1 Play/Pause

With a song playing, Tap once on the region shown in Figure 12 to pause the music. Tap again to play the music.

4.1.3 Mute

With a song playing, tap once on the region shown in Figure 12 to mute the music. Tap a second time to un-mute the music.

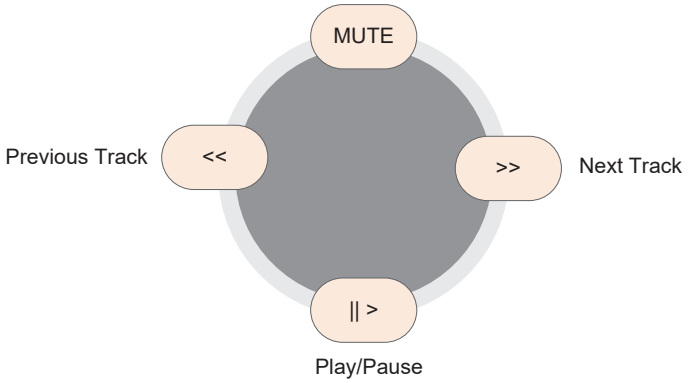


Figure 12. Wheel for Music Control

4.2 Slider

4.2.1 Volume Control

With a song playing, touch anywhere on the slider and slide your finger right to increase the volume and left to decrease the volume. Experiment with various starting positions. You should see this function works independent of where you start the gesture.

4.2.2 Changing Music Tracks

With a song playing, touch the slider and quickly swipe your finger right and release to select the next song. Make sure you move the minimum required distance. Quickly swipe your finger left and release to select the previous song. Experiment with various starting positions. You should see this function works independent of where you start the gesture.

NOTE: The slider resolution is 1000 points. The swipe gesture requires that your finger move a minimum distance of 50 points.

4.2.3 Play or Pause

With a song playing, tap once anywhere on the slider to pause the song. Tap again to play the music.

4.2.4 Mute

With a song playing, double-tap anywhere on the slider to mute the song. Double-tap again to un-mute the music.

4.3 Buttons

4.3.1 Volume Control

With a song playing, tap multiple times, or tap and hold, on the volume buttons shown in Figure 13 to increase or decrease the volume.

4.3.2 Changing Music Tracks

With a song playing, tap once on the regions shown in Figure 13 to select the next or previous songs.

4.3.3 Play or Pause

With a song playing, tap once on the region shown in Figure 13 to pause the music. Tap again to play the music.

4.3.4 Mute

With a song playing, tap once on the region shown in Figure 13 to mute the music. Tap a second time to un-mute the music.

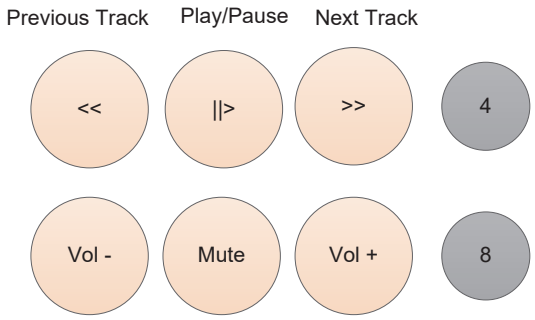


Figure 13. Buttons for Music Control

5 Design Files

5.1 Schematic

To download the schematic for the TIDM-02004, see the design files at http://www.ti.com/tool/TIDM-02004.

5.2 Bill of Materials

To download the bill of materials for the TIDM-02004, see the design files at http://www.ti.com/tool/TIDM-02004.

6 Software Files

To download the software files for this reference design, see the link at http://www.ti.com/tool/TIDM-02004.

7 Related Documentation

1. CapTIvate™ Design Center GUI for MSP430™ Capacitive Sensing MCUs
2. Texas Instruments E2E™ Community

7.1 Trademarks

MSP430, CapTIvate, Energy Trace, E2E are trademarks of Texas Instruments.
Windows is a registered trademark of Microsoft Corporation.
All other trademarks are the property of their respective owners.

8 Terminology

Self capacitance: The method of measuring changes in capacitance with respect to earth ground

HMI: Human-machine interface

CDC: CapTIvate Design Center

9 About the Author

Dennis Lehman is a Senior System Application Engineer at Texas Instruments supporting capacitive touch design solutions using MSP430 MCUs with CapTIvate technology. He has been with Texas Instruments for 8 years with a total of 15 years of experience in the semiconductor industry supporting 32-, 16-, and 8-bit MCU products. He has a BSCE from San Diego State University.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

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