Application Note: 61xxAN2

LDS61xx Capacitive Touch Sensor IC Interface, Initialization and Reference Code Guide

Abstract:

This application note describes how to communicate with and initialize the LDS61xx PureTouch™ family of capacitive touch controllers.

Contents of this Application Note

This application note contains the following sections:

Section 1 – Explains the interface methods which can be used with the LDS61xx.

Section 2 – Explains the basic driver flow from "system initialization" to "single touch sensing".

Explains how to create the initialization code for different applications

Section 3 – Explains the calibration process and methods of optimization.

Section 1: LDS61xx Interface Methods

1. Introduction:

The three supported interface standards for communication between the HOST and LDS61xx are ***I^2C, SPI*** and ***SMBus***. Selection of the desired interface is facilitated by use of the I/F Toggle pin 0,1 (0/0=I^2C, 0/1=SMBus ,1/0 : SPI, 1/1=N/A).

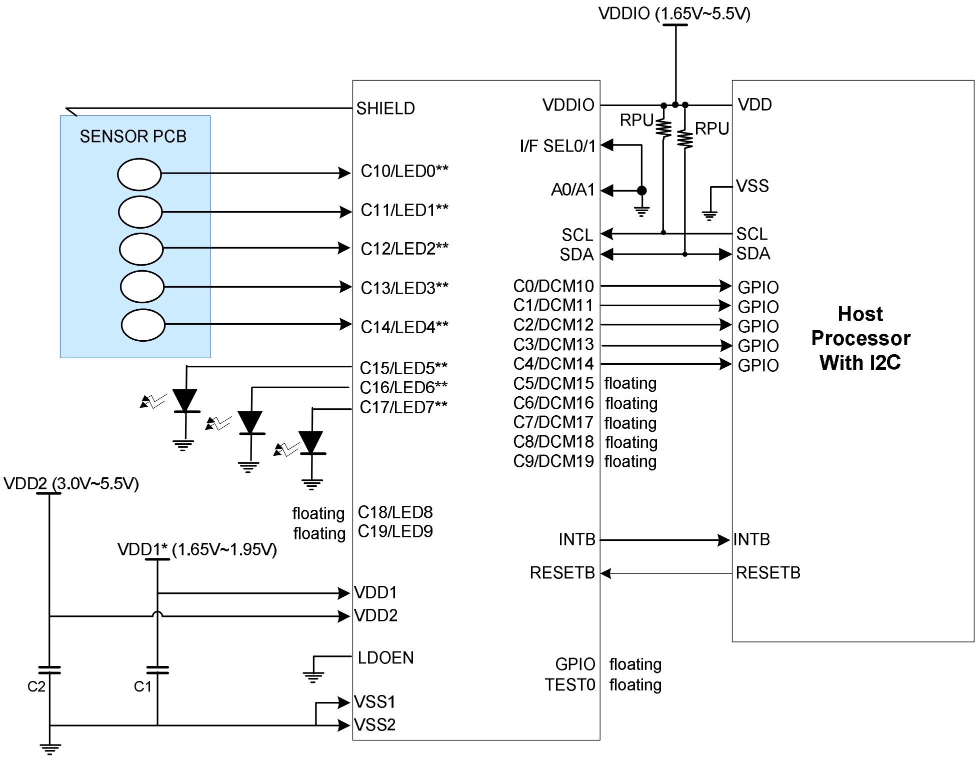
There are two methods that may be used to detect the touched state. One is the ***INTERRUPT*** method and the other is the ***POLLING*** method.

This document describes the ***INTERRUPT*** method of touch detection using the ***I^2C*** interface.

This application note is prepared based on C-code. This code is available upon request from IDT.

2. Connection between LDS61xx and HOST

The reference C-code provided by IDT is composed of the elements shown in **Figure 1.**



C1=C2: > 1uF

*For direct application of 1.8V voltage. For voltages >1.8V, apply voltage to VDDIO and tie VDD1 to ground through C1
** LED Driver functionality available on LDS612x offerings

Figure 1: Interface between HOST and LDS61xx

3. Communication Method

The LDS61xx controller is equipped with an “Interrupt Pin” that makes it easy to detect when a touch event occurs. By supporting SPI, I^{2}C and SMBus interfaces, the LDS61xx also makes it easy to implement any of these common serial interface standards for communication with a HOST processor.

(1) Interrupt Method:

The LDS61xx utilizes an “INTERRUPT” pin to inform the Host when a touch event occurs. The user can select the polarity of this pin with configuration register0x008 bit15 (default “0” = active low).

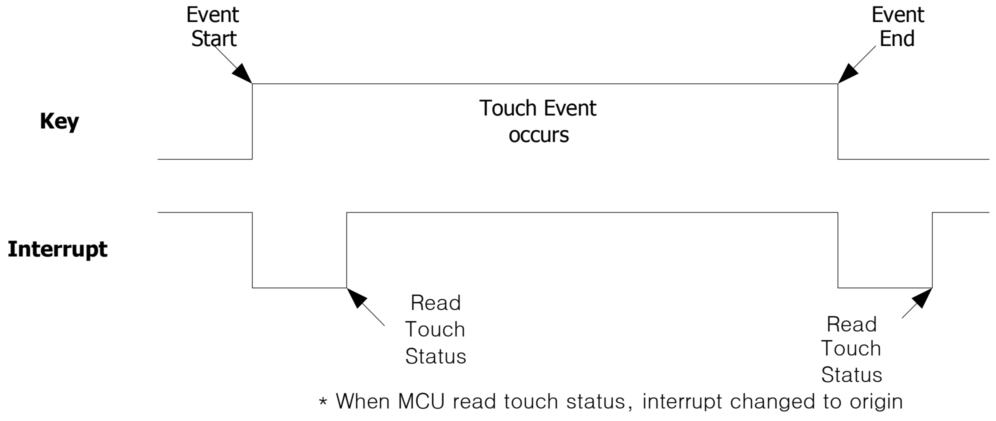


Figure 2: Touch Detecting by *INTERRUPT*

When a key/button is touched, the INTERRUPT signal is driven low as shown in Figure 2 (note: Figure 2 assumes the INTERRUPT polarity is in the active low default state). When the INTERRUPT pin is activated, the HOST reads the Touch Status Register (0x045~0x046) to determine which key(s) was touched. The Touch Status Register read will cause the LDS61xx to automatically clear the INTERRUPT pin to the inactive state, enabling it to trigger again to the active state when the next touch (or untouch) event occurs.

(2) Polling Method:

If there is no available interrupt input on the Host or if it is not possible to add an ISR (Interrupt Service Routine), the user can use the POLLING method instead. With this method, the detection of the touched state is made possible by reading the Touch Status data (register address 0x045~0x046) at regular intervals. This regular interval can be generated by a timer on the Host/MPU. For details, please refer to Figure 3.

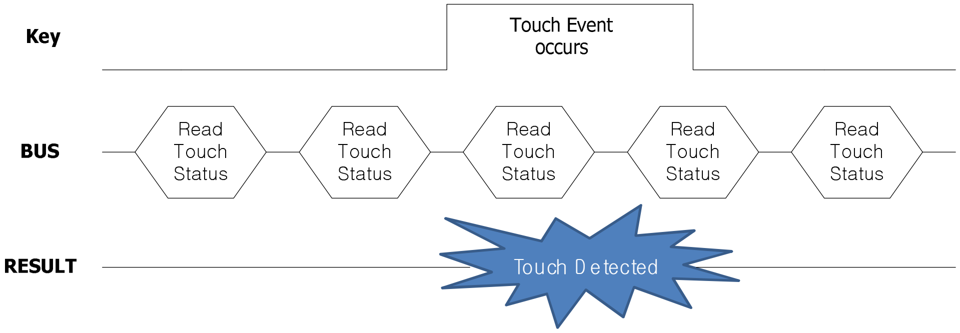


Figure 3: Touch Detection by *POLLING*

To detect the touched state, one of the above two methods is generally used by real-world applications. Either is effective, so the user may choose whichever of the two methods fits best for the specific application under consideration.

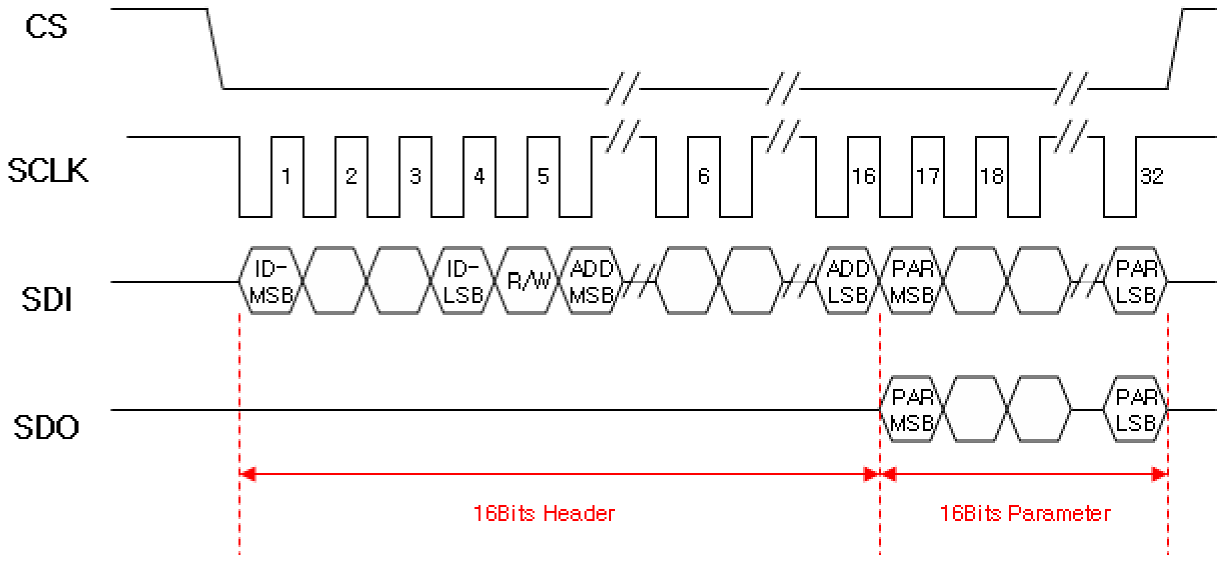
(3) SPI / I^{2}C / SMBus:

The LDS61xx supports three interface. SPI(Serial Peripheral Interface), I^{2}C(Inter-Integrated Circuit) and SMBus(System Management BUS) for as interfaces with the HOST. The selection interface is available by using the "IF_Toggle" (IF_SEL0,1) pin on the LDS61xx.

IF_SEL(1)	IF_SEL(0	Interface
LOW(GND)	LOW(GND)	I^{2}C
LOW(GND)	HIGH(VDDIO)	SMBus
HIGH(VDDIO)	LOW(GND)	SPI
HIGH(VDDIO)	HIGH(VDDIO)	N/A

Table 1: I/F Selection using "IF_Toggle"/IF_SEL pin

To use the SPI interface, the HOST should have a 4-wired SPI Module. If the host doesn't have this module, it may configure its GPIO pins for this purpose instead. Figure 4 shows the SPI Protocol of the LDS61xx.



ID: Device ID of LDS61xx is 0xE.
R/W: ‘0’ means WRITE OP and ‘1’ means READ OP
ADD: Register Address of LDS61xx
PAR: Parameter of LDS61xx when RD or WR.

Figure 4: SPI-Compatible

In order to use I^{2}C I/F, the HOST should have an I^{2}C Module. If the host does not have this module, it may configure its GPIO pins for this purpose instead.

Figures 5 and 6 show how to write and read data using the I^{2}C interface.

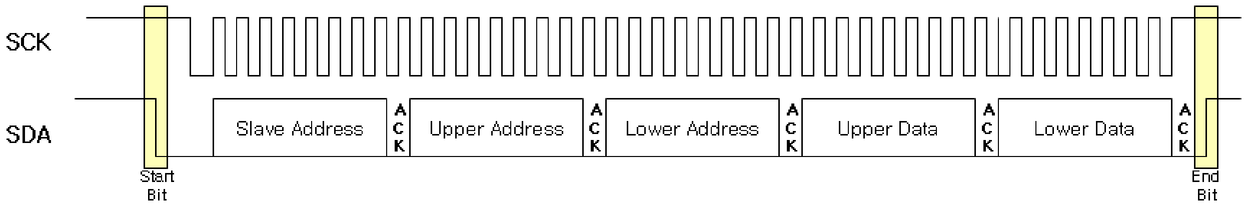


Figure 5: I^{2}C-Compatible (Write Operation)

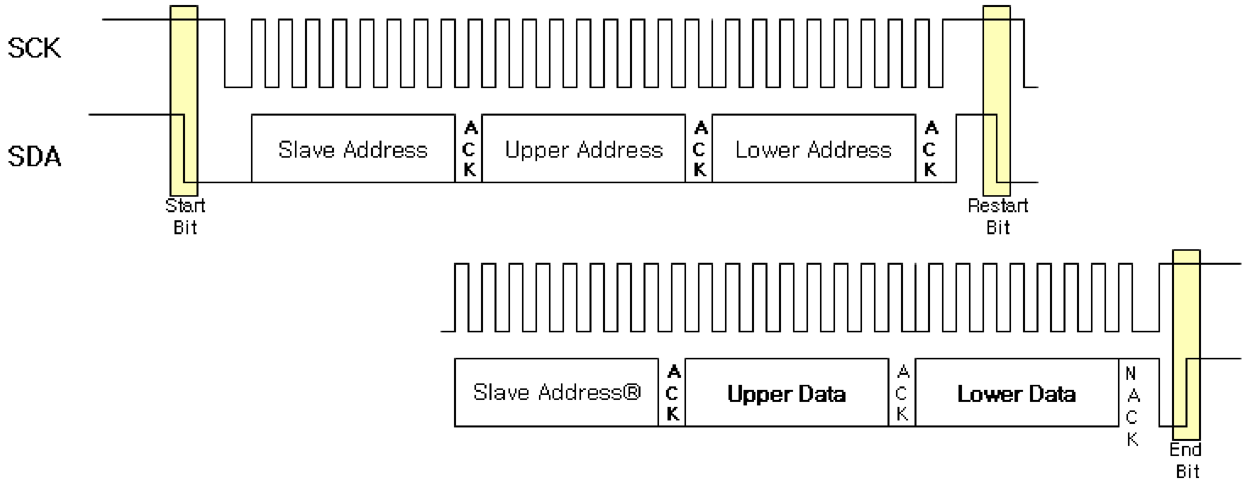


Figure 6: I^{2}C-Compatible (Read Operation)

Slave Address: Slave Address of LDS61xx used during Write Operation
* 0x2c(according to the condition of A0 and A1, this value can be changed to 0x2d, 0x2e, 0x2f)
Slave Address®: Slave Address of LDS61xx used during Read Operation
* 0x2c(according to the condition of A0 and A1, this value can be changed to 0x2d, 0x2e, 0x2f)
Upper / Lower Address: In order to access to 16bits Address, 8bits is used twice
Upper / Lower Data: In order to write/read the 16bits Data, 8bits is used twice
Bold Font: Indicates that the direction of Data transmission is from LDS61xx to HOST

Section 2: Basic Driver Flow Description and Initialization Code

1. Introduction:

In this section, the basic driver flow from “system initialization” to “single touch sensing” will first be covered.

Secondly, creation of initialization code will explained in some detail.

To create initialization code fit for most applications, the user should understand certain register definitions and operating mechanisms. The LDS6100 Family Datasheet, Register Documentation for the specific device being used, and Application Note LDS61xx AN1 (Enhanced Functionality) will be helpful in understanding these details. The default conditions for most registers should suffice in typical cases.

In certain applications, it may be necessary to adjust other configuration registers not covered in this application note. Your local IDT FAE can assist you with creating more comprehensive and specialized initialization code if your application requires such customization.

Note: The register examples shown in this application note use LDS6100/LDS6120 (20-channel product offerings) as representative devices. If lower channel products are used, please refer to the register documentation for the particular device to be utilized as register maps differ in some respects from device to device.

2. Basic Driver Flow Description:

Upon request, IDT will provide the Reference C-code to drive the LDS61xx. This C-code is prepared based on the LDS61xx EValuation Kit (“EVK”). The default reference code assumes the LDS61xx communicates with the MCU host using the INTERRUPT/“ISR” method and a software generated I^{2}C interface utilizing the Function of the EVK microcontroller. Additional host code configurations may be created upon request.

OVERVIEW:

The following flow chart shows the process from initialization of the LDS61xx to touch detection and reading of the touched value.

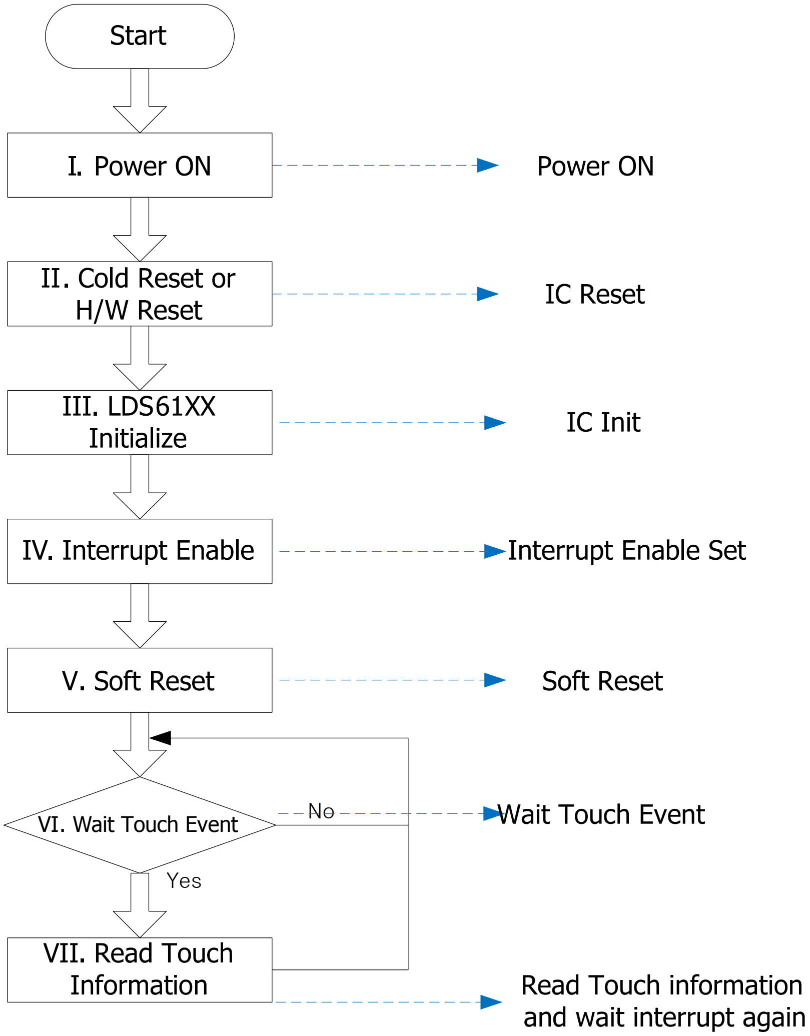


Figure 7: Basic Driver Flow Description

I. LDS61xx PowerON ( ):

“VDDIO”, “VDD1” and “VDD2” can be applied to the LDS61xx in any sequence.

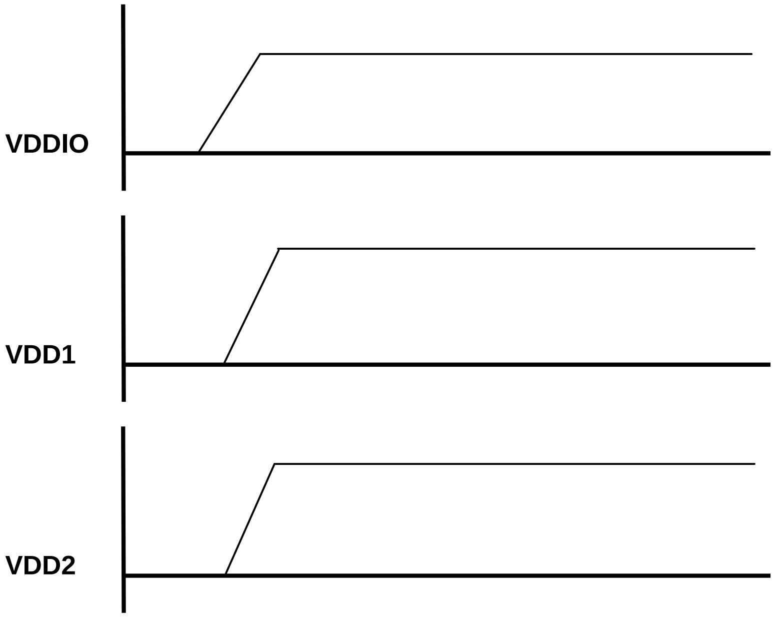


Figure 8: LDS61xx Power-on Sequence

II. COLD RESET or HW RESET:

After power-up, the LDS61xx must be reset using the COLD RESET command or the hardware reset (RESETB pin). Both result in equivalent outcomes, with the result being initialization of all configuration registers and internal memory to their default values. **Writing to register 0x0000 with any data results in a COLD RESET command.**

LDS61XX HW Reset :

To reset the LDS61xx using the RESETB pin, the reset timing shown in figure 9 is necessary.

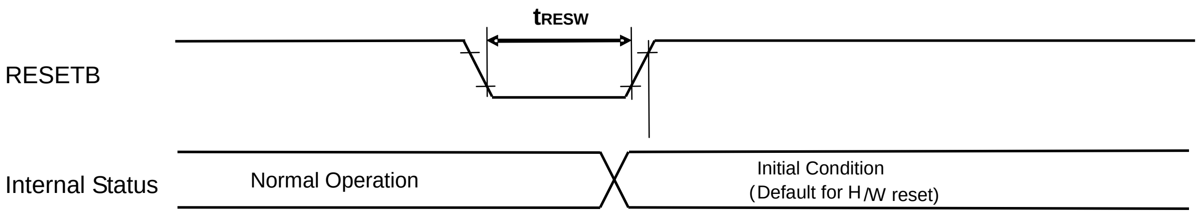


Figure 9: LDS61xx RESET Timing

Symbol	Parameter	Related Pins	MIN	TYP	MAX	Note	Unit
t_{RESW}	*1) Reset low pulse width	RESETB	10	-	-	-	μs

For a proper HW Reset to be recognized, the Reset low pulse width should ≥10μs.

III. LDS61xx Initialize:

During LDS61XX initialization, any register values differing from the default settings, as well as LED driver configuration (as relevant), should be set. Creating initialization code will be explained in detail later in this application note. The normal initialization procedure is as follows:

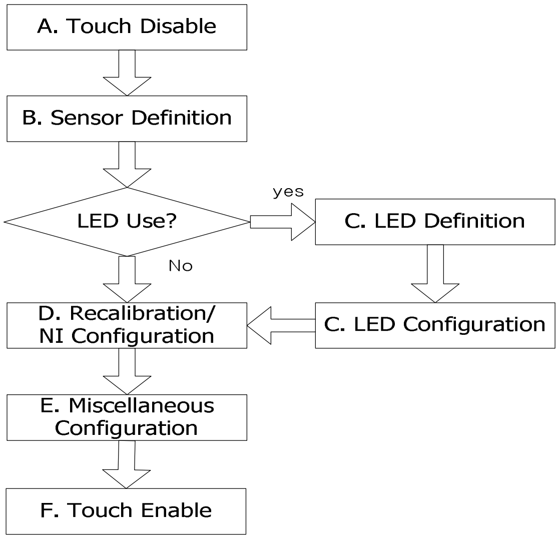


Figure 10: Basic initialization flow description.

A. Touch Disable (Register 0x040 Bit 15):

During initialization, touch functionality should be temporarily disabled (register 0x040, bit15 = 0).

	TOUCH CONFIG																
Register Address	**Direction**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0x040	R/W	TCH_ENABLE	Internal					BUTTON_MODE				Internal		READY	Decimation		
Default Values	8030h	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Register Name	Position	Description	Function	Remark
TCH_ENABLE	[15]	Touch Function Enable	0 : Touch function in idle state, 1: Touch function in active state	
BUTTON_MODE	[9:8]	Touch Preference Mode (Strongest Touch Modes)	0x0 : Unrestricted mode, All touches reported 0x1 : Strongest Touch Mode 0x2 : Two Strongest Touches Mode 0x3 : reserved	
DEVICE_READY	[3]	Device Ready	0 : Self initialization state, 1: OK for host communication If DEVICE_READY=0, all bits except DEVICE_READY should be ignored.	Read Bit
Decimation	[2:0]	CDC decimation	**0x0 : 1024 (default)** 0x1 : 512 0x2 : 256 0x3 : 128 0x4 : 2048	Consult IDT representative if decimation is changed from 1024 default

B. Sensor Definition (Registers 0x041, 0x042):

The channels to be used as touch sensors are specified with Registers 0x041 and 0x042, with a “1” bit designating a sensor as active.

		Touch Sensor Enable (Channels 0-9)															
Register Address	Direction	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x041	R/W							TOUCH_ENBL [9:0]									
Default Values	00FFh	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

		Touch Sensor Enable (Channels 10-19)															
Register Address	Direction	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x042	R/W							TOUCH_ENBL [19:10]									
Default Values	0300h	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Register Name	Position	Description	Function	Remark
TOUCH_ENBL	[9:0]	Touch enable for each channel	0 : Channel disabled as touch sensor 1 : Channel enabled as touch sensor (may be overridden by LED Enable) ***DCM register 0x00A has priority over 0x041 in case of dual***	

C. LED Definition (LDS612x products):

If integrated LED driver functionality will be utilized, the channels assigned as LED drivers are specified using Register 0x03F (“1” bits specify LED driver channels). If a channel is defined as both a touch sensor and an LED driver, the LED assignment bit will prevail.

		LED Driver Enable Configuration															
Register Address	Direction	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x03F	R/W							LED Driver Enable (LED9-LED0)									
Default Values	0000h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register Name	Position	Description	Function	Remark
LED_ENABLE	[9:0]	LED driver enable	0 : Disable 1 : Enable (LED enable bit dominates over Touch Sensor Enable in 0x042)	

LED Configuration (Registers 0x020-0x029):

These registers specify the minimum and maximum current for each LED driver, as well as the touch sensor to be associated with this LED driver. When the corresponding touch sensor is not touched, the LED driver will drive the minimum current level (which may be zero). When the touch sensor is touched, the LED driver will drive to the maximum current level.

Additional LED effects (dimming, blink, heartbeat) and associated effect timing are configured using registers 0x02F-0x039 and 0x03E. Please see Application Note 61xxAN1 for details on available effect options.

D. Recalibration & Noise Immunity Configuration (Registers 0x051, 0x052, 0x053, 0x077):

There are three registers that control the delay before a recalibration event occurs.

Register 0x051 controls the Ambient Configuration delay settings and determines how quickly ambient calibration occurs. Ambient Configuration occurs when the capacitance changes by an amount greater than the Plus Noise Level and less than the Ambient Threshold (by default between 3 and 10 capacitance value units). It is recommended to use the default values to start with. If faster ambient calibration is desired, the register value may be decreased to any non-zero value. If more time is desired before ambient calibration occurs, then a larger value may be utilized. Ambient calibration may be completely disabled using register 0x04E (set bit 13 to “1”) in special situations.

		AMBIENT CALIBRATION															
Register Address	Direction	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x051	R/W	CNT_DEC_LIMIT								CNT_INC_LIMIT							
Default Values	1F1Fh	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

Register Name	Position	Description	Function	Remark
CNT_DEC_LIMIT	[15:8]	Value determines how quickly ambient calibration is triggered (negative side)	0x0 : INVALID others : count limit	
CNT_INC_LIMIT	[7:0]	Value determines how quickly ambient calibration is triggered (posative side)	0x0 : INVALID others : count limit	

Register 0x052 controls the Recalibration Configuration delay settings and determines the delay/wait time before a recalibration is initiated when the capacitance changes by an amount greater than the Ambient Threshold but below the Threshold value (by default, above 10 capacitance value units and below Threshold). The default register value (decimal 99) results in a delay of 0.8s multiplied by the number of active sensors. When relatively few sensors are active, it is advisable to increase this register value to ensure a delay at least 4-5s. Too small a delay may result in a slowly approaching finger being calibrated out as it approaches the touch button.

		RECALIBRATION CONFIGURATION															
Register Address	Direction	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x052	R/W						RECAL_DELAY										
Default Values	0063h	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1

Register Name	Position	Description	Function	Remark
RECAL_DELAY	[10:0]	Wait time prior to forced recalibration when cap value above ambient threshold but below touch threshold (i.e. not an actual touch)	wait time = (RECAL_DELAY+1) * single round time * 4 Max time = 16.4s (1 ch) to 328s (20ch)	Default is 0.8s x # of active ch with 1024 decimation

Register 0x053 controls the Stuck Touch register and determines the delay/wait time before a recalibration is initiated when a touch is present (i.e. the capacitance value is ***above the Threshold value***). The default register value (decimal 99) results in a delay of 0.8s multiplied by the number of active sensors. When relatively few sensors are active, it is advisable to increase this register value to ensure that the delay time is longer than an expected valid touch. Too short a delay will result in an active touch being recalibrated to a non-touched state.

		STUCK TOUCH															
Register Address	Direction	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x053	R/W						CNT_TOUCH_LIMIT										
Default Values	0063h	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1

Register Name	Position	Description	Function	Remark
CNT_TOUCH_LIMIT	[10:0]	Wait time prior to forced recalibration when cap value is above threshold level (stuck touch scenario)	wait time = (CNT_TOUCH_LIMIT+1) * single round period * 4 Max time = 16.4s (1 ch) to 328s (20ch)	Default is 0.8s x # of active ch with 1024 decimation

Register 0x04E bits [3:0] control the SELC_Unit, which is a parameter used during the calibration process to reach a stable baseline/no-touch state. In most cases, a SELC_Unit value of “0” will result in the fastest calibration time. More detail on calibration is Section 3 (page 19).

		SELC CONFIG and CALIBRATION STATUS															
Register Address	Direction	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x04E	R/W	Internal	ACTIVE	AMB_DIS	Internal									SELC_UNIT			
Default Values	0002h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Register Name	Position	Description	Function	Remark
ACTIVE	[14]	touch calibration status	0 : in calibration state 1 : calibration finished	
AMB_DIS	[13]	ambient calibration disable	0 : ambient calibration active 1 : disable ambient calibration	Optional disabling of ambient cal
SELC_UNIT	[3:0]	SELC change amount during calibration	0 : adaptive SELC tracking algorithm used others : During tracking, SELC changes its value by the amount of SELC_UNIT	Refer to 6100 AN2 App Note

Register 0x077 sets the Noise Immunity configuration for the device.

	NOISE IMMUNITY ENABLE																
Register Address	Direction	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x077	R/W	NI_ENBL	INTERNAL			NI_OPTION											
Default Values	0000h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register Name	Position	Description	Function	Remark
Noise Immunity EN	[15]	Enable Noise Immunity	0 : disable, 1: enable.	
Noise Immunity Option	[11:0]	Noise Immunity Option	Setting Options.	

E. Miscellaneous Configuration:

Depending upon which features are utilized, additional configuration may be required. Examples of other features requiring register configuration include:

- Low Power (Power Savings) Mode
- DCM Mode
- Strongest/Two Strongest Touch Modes
- Built-in Slider/Scroll Mode
- Debounce
- Hysteresis
- Guard/Suppress Channel

Additional detail on these features may be found in the LDS6100 Family Datasheet or application note LDS61xxAN1 (Enhanced Functionality).

F. Touch Enable:

After Initialization, activate touch functionality by setting register 0x040, bit15 to 1.

IV. Interrupt Enable:

If the system uses Interrupt Mode to determine touch status, each channel's interrupt should be set with the Interrupt Enable registers 0x043 and 0x044.

	INTERRUPT ENABLE (Channels 0-9)																
Register Address	Direction	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x043	R/W							INT_TOUCH_EN[9: 0]									
Default Values	0000h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	INTERRUPT ENABLE (Channels 10-19)																
Register Address	Direction	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x044	R/W							INT_TOUCH_EN[19:10]									
Default Values	0300h	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Register Name	Position	Description	Function	Remark
TOUCH_INT_EN	[9:0]	Touch interrupt enable for each channel	0 : disable 1 : enable	

V. Soft Reset:

After all of the above configuration registers are written to, a Soft Reset should be initiated to perform the initial calibration. Writing any value to register 0x001 will initiate a Soft Reset.

VI./VII. Wait for Touch Event and Read Touch Information:

To monitor LDS61xx touch events, there are two possible methods. One is using the Interrupt pin with an Interrupt Service Route (ISR) and the other is using Polling.

Interrupt Method:

When the host is notified of a Touch event via the Interrupt Pin, the Touch Status Register (0x045 and 0x046) should be read via the I^{2}C Port in order to determine which key was touched.

		TOUCH STATUS (Channels 0-9)															
Register Address	Direction	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x045	R							INT_TOUCH_STS[9: 0]									
Default Values	0000h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

		TOUCH STATUS (Channels 10-19)															
Register Address	Direction	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x046	R	GPI_INT						INT_TOUCH_STS[19:10]									
Default Values	0000h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register Name	Position	Description	Function	Remark
GPI_INT	[15]	GPI interrupt status	0 : no GPI interrupt, 1 : GPI interrupt (Used if GPIO is configured as input via Register 0x009)	
TOUCH_INT_STS	[9:0]	Touch interrupt status for each channel	Indicates which touch sensor channel was activated by a touch when INTB signal is triggered	

Interrupt pin polarity may be changed from its default active low state via register 0x008 bit 15.

		INTB Config															
Register Address	Direction	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x008	R/W	INT_POL	Internal										INTB DRIVE			INTB MODE	
Default Values	0000h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register Name	Position	Description	Function	Remark
INT_POL	[15]	interrupt polarity	0 : active Low, 1: active High	
INTB DRIVE	[4:3]	interupt drive	0x0 : CMOS output 0x1 : OPEN-DRAIN output (pull up resistor required)	
INTB MODE	[1:0]	interupt signaling type	00: Fixed Duration Mode (INTB pin drives for 2us Fixed Duration) 01: Internal mode 1x: Read Reset Mode (INTB pin drives until Register 0x045 is read)	

Interrupt Configuration Register (Interrupt Polarity Setting)

POLLING:

To determine touch status without the use of the Interrupt pin, Touch Status registers 0x045 and 0x046 may be polled regularly to look for changes in touch status.

If a touch occurs, the corresponding bit for the touched sensor changes from ‘0’ to ‘1’. If an untouch occurs (finger lifts off of sensor), the corresponding bit for the untouched sensor changes from ‘1’ to ‘0’.

Ex) If touches occur at channels 3 and 12, the value of register 0x045 is changed from 0x0000 to 0x0008(Bit3) and the value of register 0x046 changes 0x0000 to 0x0004 (Bit2).

	TOUCH STATUS (Channels 0-9)																
Register Address	Direction	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x045	R							INT_TOUCH_STS[9: 0]									
Default Values	0000h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	TOUCH STATUS (Channels 10-19)																
Register Address	Direction	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x046	R	GPI_INT						INT_TOUCH_STS[19:10]									
Default Values	0000h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register Name	Position	Description	Function	Remark
GPI_INT	[15]	GPI interrupt status	0 : no GPI interrupt, 1 : GPI interrupt (Used if GPIO is configured as input via Register 0x009)	
TOUCH_INT_STS	[9:0]	Touch interrupt status for each channel	Indicates which touch sensor channel was activated by a touch when INTB signal is triggered	

Touch Status Register

3. Initialization Code Creation:

The following initialization code illustrates an example using the LDS6100 Evaluation/Tuning kit (EVK).

The LDS6100 EVK utilizes the LDS6120 device with five element one slider (sensors 0-4), five typical 7x7 touch buttons with associated LEDs (sensors 8, 11, 13, 16, and 18), and one proximity/guard channel (sensor 7) configured as a guard channel that masks all other channels if activated.

Each of the five 7x7 touch buttons have an associated LED driver (or two LED drivers in one case), assigned as follows:

- LED Driver 0 (C10/***LED0*** channel) assigned to sensor C8 (***C8***/DCM18 channel)
- LED Driver 2 (C12/***LED2*** channel) assigned to sensor 11 (***C11***/LED1 channel)
- LED Driver 4 (C14/***LED4*** channel) assigned to sensor 13 (***C13***/LED3 channel)
- LED Driver 5 (C15/***LED5*** channel) also assigned to sensor 13 (***C13***/LED3 channel)
- LED Driver 7 (C17/***LED7*** channel) assigned to sensor 16 (***C16***/LED6 channel)
- LED Driver 9 (C19/***LED9*** channel) assigned to sensor 18 (***C18***/LED8 channel)

(***bold italics*** above indicate function assigned to dual-purpose pins using TOUCH SENSOR ENABLE and LED DRIVER ENABLE registers)

LED on currents are set to maximum and effects for LED 0, 2, and 9 are set for dimming, blink, and heartbeat, respectively.

In this initialization code, commonly configured registers are included.

However, in non-typical applications, initializing additional registers may be necessary for optimizing initial calibration and configuring additional features/functionality. By reading this application note carefully, a user should be able to understand how to create initialization code fit for most common applications.

Please contact your local IDT FAE if the Ambient Configuration, Recalibration Configuration, Stuck Touch or other settings presented in this example do not result in robust touch operation for your application.

* This initial code designed for LDS6100 EVK. // ************************************ // LDS6100 EVK - IDT Touch Initial Code // ************************************	
// COLD RESET WR 0x0000 0x0000	*Cold Reset. Write any value to Register 0x00
// TOUCH DISABLE WR 0x0040 0x0030	* Touch Disable. Address 0x0040 bit 15.“0” => Disable, “1” => Enable.
// DCM CONFIG (DISABLE) WR 0x000A 0x0000	* DCM Interrupt Enable / Disable “0” => Disable, “1” => Enable.
// SLIDER CONFIG : 5ch WR 0x0074 0x841F	* Scroll / Slider interrupt Enable, Define Which Channels Make Up the Slider
// TOUCH SENSOR ENABLE WR 0x0041 0x019F WR 0x0042 0x014A	* Define Sensor Channel.
// INTERRUPT WR 0x0043 0x019F WR 0x0044 0x014A	* Define Interrupt Channel Enable / Disable
// LED DRIVER ENABLE WR 0x003F 0x02B5	* Define LED Channel.
// LED EFFECT CONFIG WR 0x002F 0xC005	* Define LED Effects and Timing
// LED0 WR 0x0020 0xF808	* Set LED 0 to max current and assign to sensor 8
// LED0 Dimming : Dimming WR 0x0030 0x0040	* Configure LED 0 for Dimming Effect
// LED2 WR 0x0022 0xF80B	* Set LED 2 to max current and assign to sensor 11
// LED2 Dimming : Blink WR 0x0032 0x8040	* Configure LED 2 for Blink Effect
// LED4 WR 0x0024 0xF80D	* Set LED 4 to max current and assign to sensor 13
// LED5 WR 0x0025 0xF80D	* Set LED 5 to max current and assign to sensor 13
// LED7 WR 0x0027 0xF810	* Set LED 7 to max current and assign to sensor 16
// LED9 WR 0x0029 0xF812	* Set LED 9 to max current and assign to sensor 18
// LED9 Dimming : Heartbeat WR 0x0039 0x4040	* Configure LED 9 for Heartbeat Effect
// Ambient Calibration WR 0x0051 0x0A1F	* Ambient Calibration Setting
// Recalibration Configuration WR 0x0052 0x07FF	* Recalibration Delay Setting (set to Max value)

// Stuck Touch WR 0x0053 0x07FF	* Stuck Touch Recalibration Delay (set to Max value)
// SELC_Unit Configuration WR 0x004E 0x5000	* Sets SELC_Unit = 0 to activate adaptive tracking algorithm
// Noise Immunity WR 0x0077 0x0000	* Noise Immunity register
// C0 ~ C4 Threshold : 60 WR 0x005F 0x0001 WR 0x0060 0x003C WR 0x0061 0x003C WR 0x0062 0x003C WR 0x0063 0x003C WR 0x0064 0x003C	* Select Memory Page 1 to Threshold Memory Page, Set Slider Sensors 0-4 sensitivities to 60 (from default 40)
// C13 Threshold : 35 WR 0x006D 0x0023	* Set Sensor 13 Threshold to 35 (all other sensors will have default Threshold of 40)
// C7 Ambient Threshold : 5 WR 0x005F 0x0002 WR 0x0067 0x0005	* Select Memory Page 2, Set C7 Ambient Threshold to 5 (from default value of 10)
// Guard Channel Enable : C7 WR 0x007C 0x0080 WR 0x007D 0x0000	* Guard Channel Enable for C7
// Guard Channel Mask WR 0x007E 0x291F WR 0x007F 0x0005	* Guard Channel Mask (Mask all channels when Guard Channel is active/touched)
// TOUCH ENABLE WR 0x0040 0x8038	* Touch Enable/All Touch Setting
// SOFT RESET WR 0x0001 0x0000	* Software Reset

Section 3: Calibration and Calibration Optimization:

(1) Initial Calibration:

- Once power is supplied to the LDS61xx and the Soft Reset command is provided after device initialization, initial calibration starts to search for the baseline SELC value for each sensor. SELC is an indication of the capacitance of each sensor and enables a stable baseline (no-touch) level to be maintained. The details of this process are summarized in Figure 11 and the associated detailed steps below.

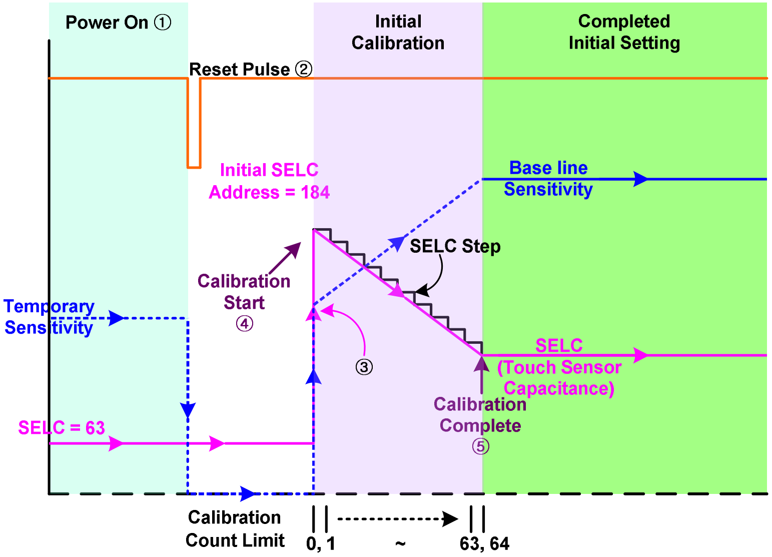


Figure 11: Initial Calibration.

① Once power is supplied to the LDS61xx, sensor sensitivities ("Cap Values") temporarily have a random value and the SELC value has a starting value of "63".

② After a Cold Reset and the initialization code is written to the LDS61xx, a Soft Reset command should be given. Upon the Soft Reset, sensitivities ("Cap Values") fall to "0", while the SELC value of each sensor momentarily remains at "63".

③ The IC then prepares for initial calibration by quickly increasing SELC to the Initial SELC value of 184 decimal (default), during which time, the sensitivities ("Cap Values") increase to a starting value correlated with the Initial SELC value. After this occurs, the IC is ready for Initial Calibration to begin.

④ Calibration Start

Initial Calibration searches for the appropriate SELC value for each sensor individually. This SELC value is effectively a representation of the actual capacitance of each sensor. As each sensor's characteristics are different based upon size, location, and proximity to signal and GND, the SELC value of each sensor will generally be different.

Starting from the Initial SELC value (default 184 decimal), the calibration engine will step in increments of the SELC Step Size ("SELC Unit", bits {3:0} of register 0x04E) until it finds the SELC value that balances the sensor's capacitance value. The number of iterations allowed is a function of the Calibration Count Limit. By default, the SELC_Unit = 2 (decimal) and Calibration Count Limit = 0 (or unlimited iterations). A SELC_STEP setting of "0" uses a more intelligent adaptive SELC tracking algorithm and converges to the proper SELC value more quickly in most cases. In typical situations, Calibration Count Limit should be kept at its default value and SELC_Unit should be set to "0" for the fastest calibration time.

⑤ Calibration Complete

By the end of the calibration process, the no-touch capacitance value ("Cap Value") of each sensor should be equal to roughly one-half of the Decimation Rate (default value of 1024, meaning the baseline Cap Value is roughly equal to 512). To make sure that initial calibration ended normally, you may verify that the sensitivity of sensor is roughly 512 (assuming you haven't changed the Decimation Rate) and that the SELC value is stable. If one or more of the above is not true, calibration may not have occurred properly.

(2) Optimizing the Initial SELC Value to Reduce Calibration Time:

By setting the Initial SELC value to be close to the final SELC value of each sensor, the initial calibration and recalibration times can be reduced significantly. Even with manufacturing variations in PCB/ITO sensor board characteristics, the SELC value of any given sensor should remain relatively unchanged from sensor board to sensor board.

Therefore, by determining the typical SELC value during prototyping and design-in, the set of optimized SELC values for each sensor may be included as part of the Initialization code to ensure that the starting SELC value of each sensor is close to the final value and in-system calibration times are reduced.

Register addresses 0x060-0x073 (memory page 0) are used to set the Initial SELC values for each sensor.

Related Available Documents:

- LDS6100 Family Product Brief
- LDS6100 Family Datasheet
- LDS61xx Detailed Register Documentation (5 versions available depending upon specific device being used)
- Other Relevant Application Notes
    - 60x0 AN1 (Factors Affecting Sensitivity)
    - 60x0 AN2 (PCB Design Guide)
    - 60x0 AN5 (Preliminary – Use of Shield)
    - 61xx AN1 (Enhanced Functionality)