Sparkfun Touch Potentiometer User Manual

Table of Contents

Touch Potentiometer

Description

The Sparkfun Touch Potentiometer is an intelligent linear capacitive touch sensor that implements potentiometer functionality with 256 positions. It can operate as a peripheral to a computer or embedded micro-controller or in a stand-alone capacity. The Touch Potentiometer provides both a dual-channel analog and PWM output for direct control of other circuitry. Configurable analog and PWM transfer functions support a wide variety of applications.

Features

- Dual host interfaces: Logic-level serial and I^2C^{TM}
- Dual 8-bit 20 k-ohm 3-terminal digitally controlled variable resistor outputs
- PWM output
- 8 LED display with multiple display modes and intensity levels
- Option for interpolated (soft) changes between touches
- Configurable touch sensor parameters for a variety of PCB covers
- Easily configurable I^2C address to allow multiple devices on one bus
- Configurable linear or non-linear PWM transfer function
- Configurable linear or simulated logarithmic variable resistor transfer function
- Variable resistor supports single- or dual-supply operation
- Simple register interface with jabber option
- Programmable power-on default operation
- Built-in calibration procedure
- User-accessible EEPROM data storage
- Built-in Low-dropout voltage regulator for operation up to 12 VDC

Applications

- Analog potentiometer replacement
- Stereo channel audio level control
- Computer peripheral
- Embedded system control
- Dimmer for LED lighting applications
- DAC replacement

danjuliodesigns LLC

Version Information

Datasheet Revision	Firmware Version	Comments
1.0	1.1	Initial Release
1.1	1.1	Fixed some typos. Updated LED dimmer and amplifier example circuit diagrams. Enhanced example description text.
1.2	1.3, 1.4	Updated for firmware versions 1.3 (danjuliodesigns version) and version 1.4 (Sparkfun version). 1. Firmware now requires 3 quick presses of the button to enter I^{2}C Address set mode. 2. Firmware now indicates entry into I^{2}C Address set mode with 3 quick blinks of the LED display. 3. Added indication of I^{2}C Address set mode in the Status register (bit 4). 4. Added backup value set of EEPROM entries at the end of the EEPROM space (locations 237 - 255) to support per-device calibration with factory default restore. Noted that values listed in the manual may differ in specific devices.
1.3	1.3, 1.4	Updated calibration procedure.
1.4	1.5, 1.6	Updated for firmware versions 1.5 (danjuliodesigns board) and 1.6 (Sparkfun board). 1. Firmware now supports built-in calibration function entered by 4 quick presses of the button. 2. Changed button operation to determine function by the number of presses detected in a two-second window. 3. Added two additional status bits indicating built-in calibration mode active and button pressed. 4. Added adaptive upper limit to idle count detection to account for a condition that may occur when the sensors are exposed to electrical noise (e.g. from a ground loop condition).

Disclaimer

Contact

Email: info@danjuliodesigns.com

Website: http://www.danjuliodesigns.com/products/touch_pot/touch_pot.html

Electrical Specifications

Parameter	Min	Typ	Max	Unit	Conditions
VIN Supply Range	4.6		12	V	post-LDO VCC = 4.5 - 5.1 volts
Logic Operating Current		8	25	mA	No significant current being sourced by the PWM output
Digital Potentiometer Supply (single-ended: V- connected to GND)	4.5		16.5	V	Jumper J1 removed
Digital Potentiometer Supply (dual-ended)	±4.5		±5.5	V	Jumpers J1 and J2 removed
Digital Potentiometer Operating Current (source/sink V-/V+)			1	µA	
Digital Potentiometer Ax, Bx, Wx signal Voltage Range	V-		V+	V	Note 1
Digital Input Low Level	VSS		0.8	V	
Digital Input High Level	2		VCC	V	
Digital Output Low Level			0.6	V	
Digital Output High Level	VCC - 0.7			V	
Maximum output current sunk by PWM signal			25	mA	
Maximum output current sourced by PWM signal			25	mA	
PWM Output frequency		488.3		Hz	
Digital Potentiometer Ax to Bx terminal resistance		20k		Ω	
Digital Potentiometer Differential Nonlinearity	-1	±0.25	+1	LSB	R_{WB}, Ax = no connect - Note 2
Digital Potentiometer Nonlinearity	-1	±0.5	+1	LSB	R_{WB}, Ax = no connect - Note 2
Digital Potentiometer Nominal Resistor Tolerance	-30		30	%	T_A = 25ºC
Digital Potentiometer Resistance Temperature Coefficient		35		ppm/ ºC	Wx = no connect
Digital Potentiometer Tap Resistance		78		Ω	

Parameter	Min	Typ	Max	Unit	Conditions
Digital Potentiometer Wiper Resistance		50	150	Ω	
Digital Potentiometer Channel Resistance Matching		0.1		%	Touch Pot Value = 50%
Digital Potentiometer Resistance Drift		0.05		%	
Digital Potentiometer Ax and Bx Capacitance		25		pF	
Digital Potentiometer Wx Capacitance		55		pF	
Digital Potentiometer Bandwidth -3 dB		310		kHz	
Digital Potentiometer Total Harmonic Distortion		0.014		%	
Digital Potentiometer CrossTalk		1		nV-sec	V_A = +5V, V_B = 0V, measure V_W with adjacent RDAC making full-scale code change
Digital Potentiometer Analog Crosstalk		-64		dB	V_{A1} = +5V, V_{B1} = 0V, measure V_{W1} with V_{W2} at 5V p-p at f = 10 kHz
Digital Potentiometer Resistor Noise Voltage		13		nV/ $\sqrt{Hz}$	
Serial Baud Rate		9600		Baud	Fixed Baud Rate
I²C Clock Rate			100k	Hz	
EEPROM write cycles	100K	1M		Write Cycles	$T_A \leq 85°C$
EEPROM characteristic retention	40			Year	Provided no other specifications are violated
Operating Temperature Range	0	20	60	°C	
Storage Temperature Range	-30	20	70	°C	

Notes

1. Digital Potentiometer signals Ax, Bx, and Wx must not exceed the voltage range between V- and V+. Power should be applied to V-, V+ and VIN before or at the same time voltages appear on Ax, Bx and Wx.
2. Resistor position nonlinearity error is the deviation from an ideal value measured between the maximum resistance and the minimum resistance wiper positions. Resistor differential nonlinearity measures the relative step change from ideal between successive tap positions. The AD5262 part used in the Touch Pot is guaranteed monotonic.

Mechanical Specifications
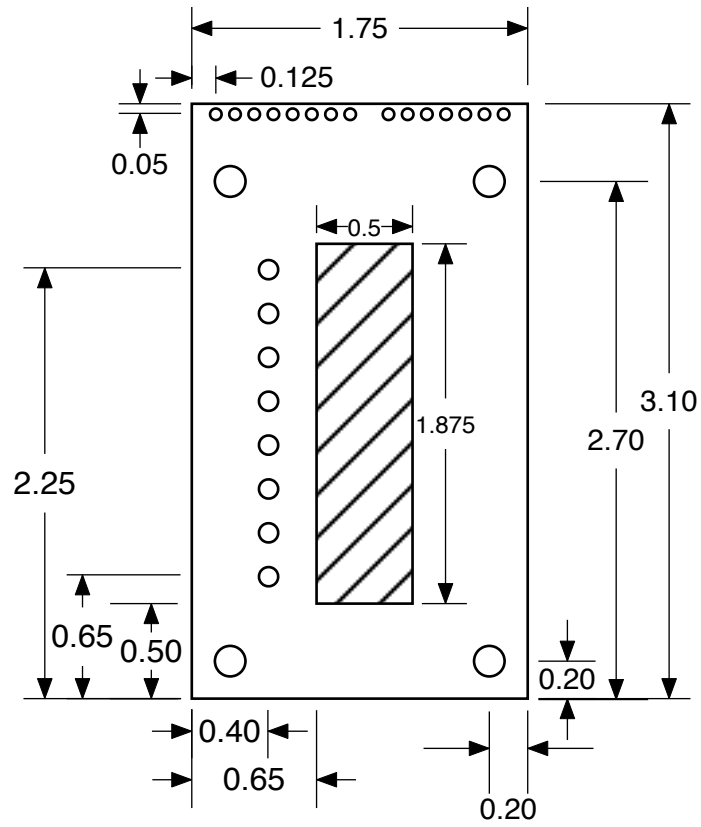
Board Dimensions (inches)

Getting Started

Block Diagram
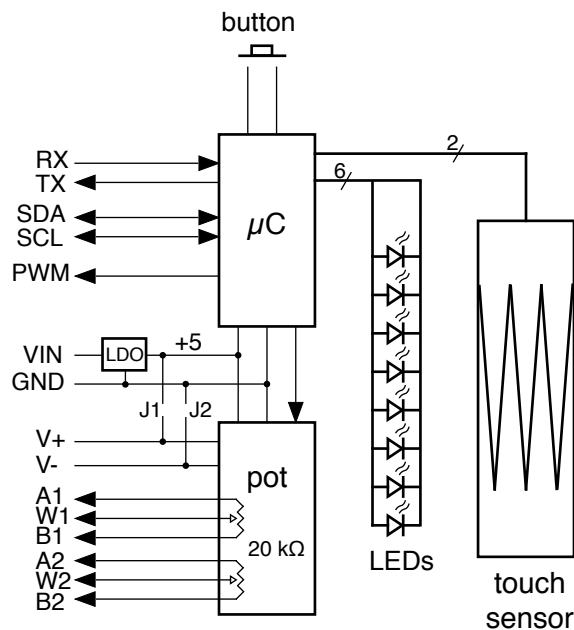
Touch Potentiometer Block Diagram

Circuit Description

The Touch Potentiometer is controlled by a Microchip PIC16F1829 8-bit micro-controller that provides the host interface, LED control, capacitive sense and peripheral control functions. The digital potentiometer functions are provided by an Analog Devices AD5262 integrated circuit. Capacitive sensing is done by two, interleaved sensors on the top side of the PCB that control an oscillator inside the micro-controller. Touching the sensors changes the oscillator frequency which is read by the micro-controller. Position is determined by a ratio-metric calculation of the changes in oscillator frequency for both sensors. The LEDs are controlled using pulse-width-modulation and multiplexing. A push button allows changing the I2C address and device calibration without an external computer or micro-controller. A Low-dropout voltage regulator allows operation up to 12 VDC.

Connections

Connections to the Touch Potentiometer are made with through-hole connectors on 0.1 inch centers or solder pads on the rear side of the circuit board (useful for mounting the touchpad under a flat bezel). There are two sets of connections, one set for the digital power and IO and another set of connections for the analog potentiometer power and signals.

Two jumpers, described later, allow powering the analog potentiometer from the digital +5V power.

A button allows the user to quickly change the I^{2}C address of the Touch Potentiometer using the touch sensor.

Note: Extreme care should be taken when soldering or de-soldering leads to the Touch Potentiometer solder pads or adding or removing a jumper. Too much heat or pressure may tear the pad or jumper location off the PCB. Use a fine-tip soldering iron and a minimum amount of solder.
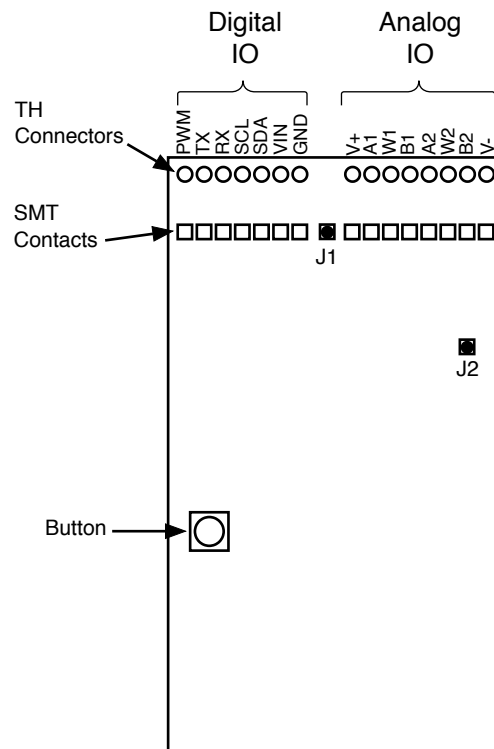
Board Layout (Rear View)

Digital IO

The Touch Potentiometer Digital IO connections consist of the VIN and ground power signals (5 - 12 VDC) for the micro-controller and digital portion of the AD5262, a TTL-level serial interface, an I^{2}C interface and the PWM output.

The Touch Potentiometer communicates to a host device using TTL-level serial interface or an I^{2}C interface. Both interfaces are active simultaneously. The serial interface operates at 9600 baud. Data is transmitted using 8 data bits, 1 stop bit and no parity (8N1). The I^{2}C interfaces is a 7-bit slave with a maximum clock rate of 100 kHz. It does not support General Call or 10-bit addressing.

The PWM output generates a signal with a duty-cycle that is proportional to the current Touch Potentiometer value. A value of zero results in a PWM output of 0% duty cycle (off). A value of 255 (full-scale) results in a PWM output of nearly 100% duty cycle (on).

Label	Function
GND	Ground
VIN	5 - 12 volt power input
RX	Serial TTL RX input
TX	Serial TTL TX output
PWM	Pulse Width Modulation Output
SCL	I²C SCL (clock)
SDA	I²C SDA (data)

Digital IO pad descriptions

Important Note about Power Supplies

The capacitive sensors may be adversely affected by electrical noise. They are sensitive to 50 and 60-Hz energy that may be coupled through power supplies with inadequate line filtering or configurations with a ground loop (for example, a system that has a DC power supply and is also connected to a computer with its own power supply). It may be necessary to include an AC line filter in front of some cheap switch-mode power supplies or to eliminate ground loop conditions or eliminate ground loops.

Analog IO

The Touch Potentiometer Analog IO signals consist of the AD5262 wiper and wiper power supply signals. The AD5262 supports two separate digital 20 k-ohm potentiometers, each with two terminals and a wiper connection. They have their own power supply connections allowing the voltage levels on the potentiometers to exceed the +5 volt logic power supply (see important note below).

Label	Function
A1, A2	A Terminals for potentiometer 1 and 2
W1, W2	Wiper Terminals for potentiometer 1 and 2
B1, B2	B Terminals for potentiometer 1 and 2

Label	Function
V+	Positive Power Supply. Connected at the factory to the 5V logic signal by jumper J1. With J1 removed may be connected to a positive voltage up to 15V. Note the sum of |V-| + |V+| must be 15V or less.
V-	Negative Power Supply. Connected at the factory to ground by jumper J2. With J2 removed may be connected to a negative voltage down to -5V. Note the sum of |V-| + |V+| must be 15V or less.

Analog IO pad descriptions

Important Notes about V+ and V-

Care must be taken with V+ and V- to prevent damage to the ICs on the Touch Potentiometer.

1. V+ and V- must always be connected to power and should be powered before or at the same time voltages appear on the A, B and W signals and 5V input.
2. By default, V+ is connected to 5V with jumper J1 and V- is connected to ground with jumper J2. Voltages on the A, B and W signals should not exceed the range of 0 - 5V with these jumpers installed. Remove these jumpers by removing the solder blob if a different power supply will be connected to V+ and/or V-.
3. The maximum voltage potential between V- and V+ is 15 volts. V- maximum is -5V. V+ maximum is 15V.
4. Electrical noise on V- and V+ may be coupled into the signal passing through the potentiometer. A power supply connected to V- and V+ may require additional filtering to eliminate this noise.

Jumpers

Jumpers J1 and J2 are designed to be bridged with solder (no other conductor necessary). They may be removed by carefully sucking up the solder using a vacuum solder sucker or solder wick.
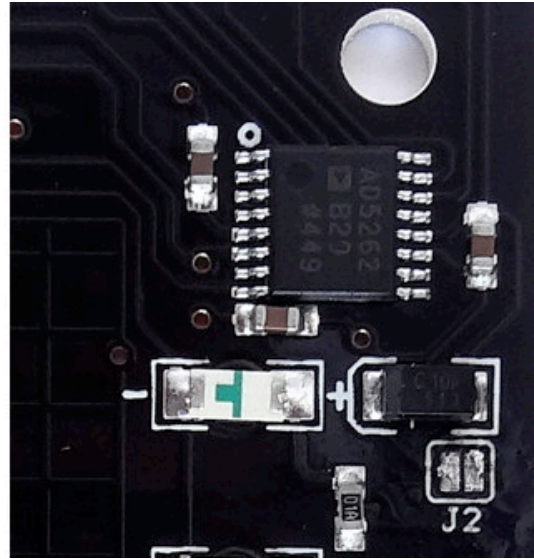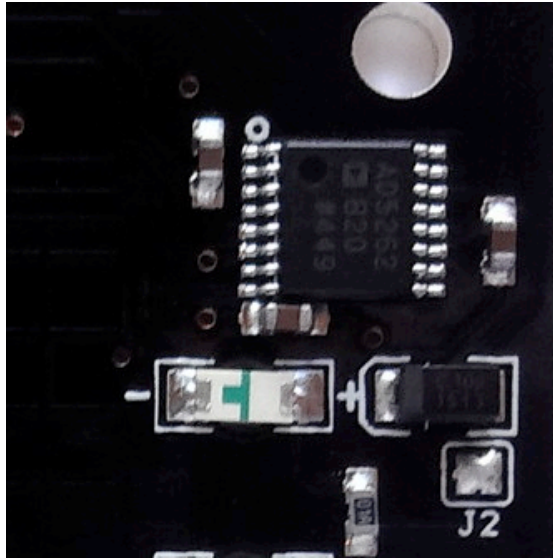
Jumper installed and removed

Quick Test

The Touch Potentiometer may be tested with only a 5 -12 volt regulated power supply (do not let the VIN supply exceed 12V).

Attach the Touch Potentiometer to the power supply and switch on the power supply. Four LEDs should light as shown in the following figure. This indicates the Touch Potentiometer is performing an initial calibration to determine its idle average values. Make sure you are not touching the sensor during the calibration. The calibration will last approximately two seconds and then the Touch Potentiometer should begin normal operation.
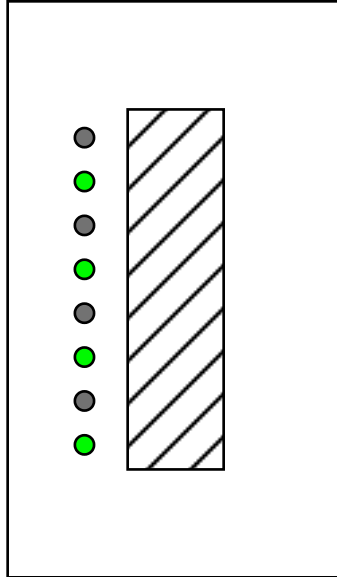
Initial Calibration

In normal operation, the factory default Touch Potentiometer will light all eight LEDs with a bright "spot" that indicates the current value ("linear" display mode). Sliding a finger along the sensor area will move the spot to follow the touch location. After a few seconds of inactivity the LED display will automatically dim.

PWM Output Example

The PWM output may be used to control external devices such as the intensity of LED lighting strips. It is a 5V logic-level, active-high signal with a maximum drive capability of 25 mA. It may be used to directly control the gate of a MOSFET transistor, base of a bipolar transistor (through an appropriately sized base resistor) or connect directly to another 5V logic input. The PWM output has a period of approximately 488 Hz and a minimum high time of 2 uSec. It may be configured with a linear or non-linear transfer function. The non-linear transfer function is useful in dimming applications.

12VDC LED Strip Dimmer
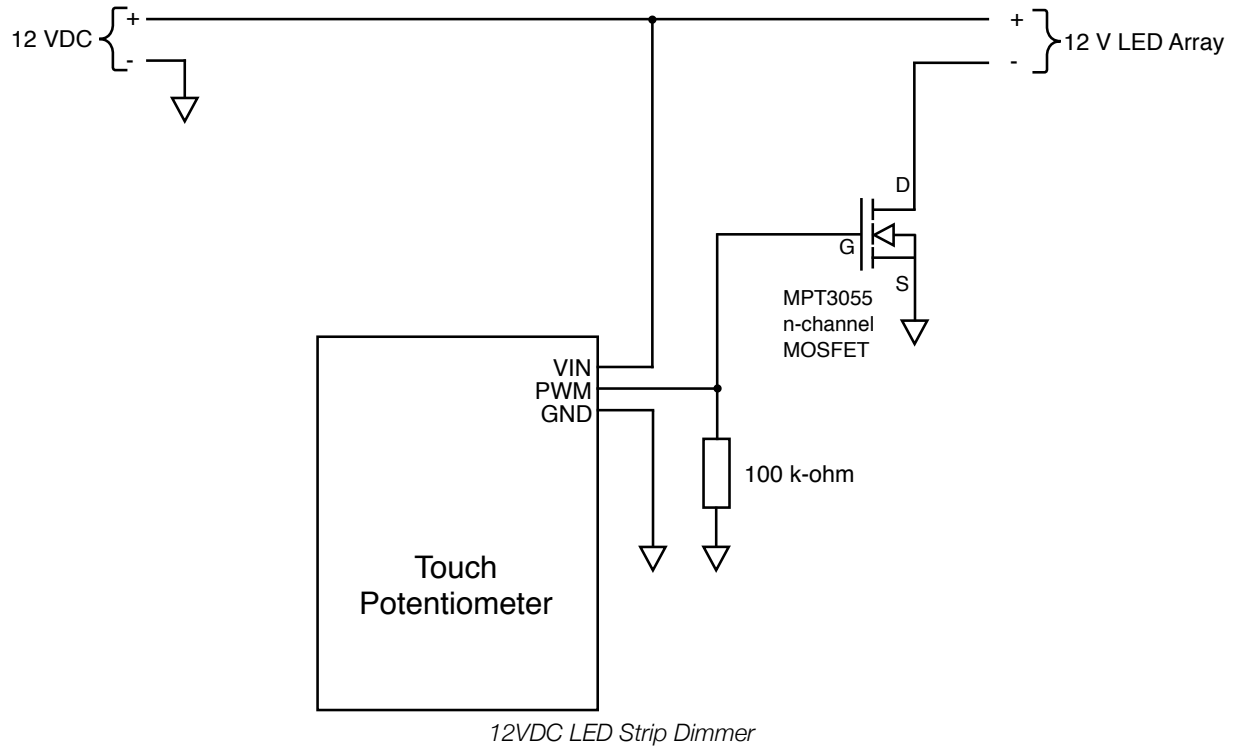
The PWM output controls a low-side switch in this example. The 100 k-ohm resistor keeps the MOSFET turned off while the micro-controller in the Touch Potentiometer starts up (and the PWM output is not driven).

Serial Interface with a Computer

A computer with a USB interface and terminal emulator program can access the Touch Potentiometer using the serial interface connected to a USB-to-serial device like a FTDI FT232R break-out board or a micro-controller like the Arduino Leonardo that has both a USB interface with Communications Device Class (CDC) support and a serial port. The Touch Potentiometer serial interface operates at 5V logic levels with a data rate of 9600 baud, eight data bits, no parity and one stop bit (8N1).
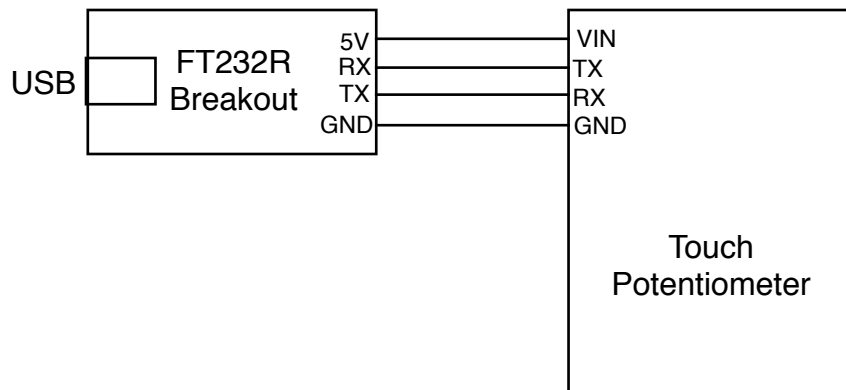
USB Serial Interface

The serial interface is connected to a computer using a FTDI FT232R breakout board. The user may interact with the Touch Potentiometer using a terminal emulator program running on the computer. The Touch Potentiometer implements a simple ASCII command interface described in the section "Command List".

A few example commands are described below. It is assumed the Touch Potentiometer has just been powered up (untouched) and the terminal emulator started and connected to the serial port. The terminal emulator is also configured to locally echo typed characters (the Touch Potentiometer does not echo back received characters).

1. Type the character 'G' followed by the Return key. This "gets" the current potentiometer value, a number between 0 and 255. The Touch Potentiometer powers up with a value of 0.
2. Type the string "W 1 128" followed by the Return key. This "sets" the current potentiometer value to mid-point. You should see the LEDs immediately change to represent the change.
3. Type 'G' followed by the Return key again. The Touch Potentiometer will respond with the value of 128.
4. Experiment by changing the value with your finger and reading it out using the 'G' command.
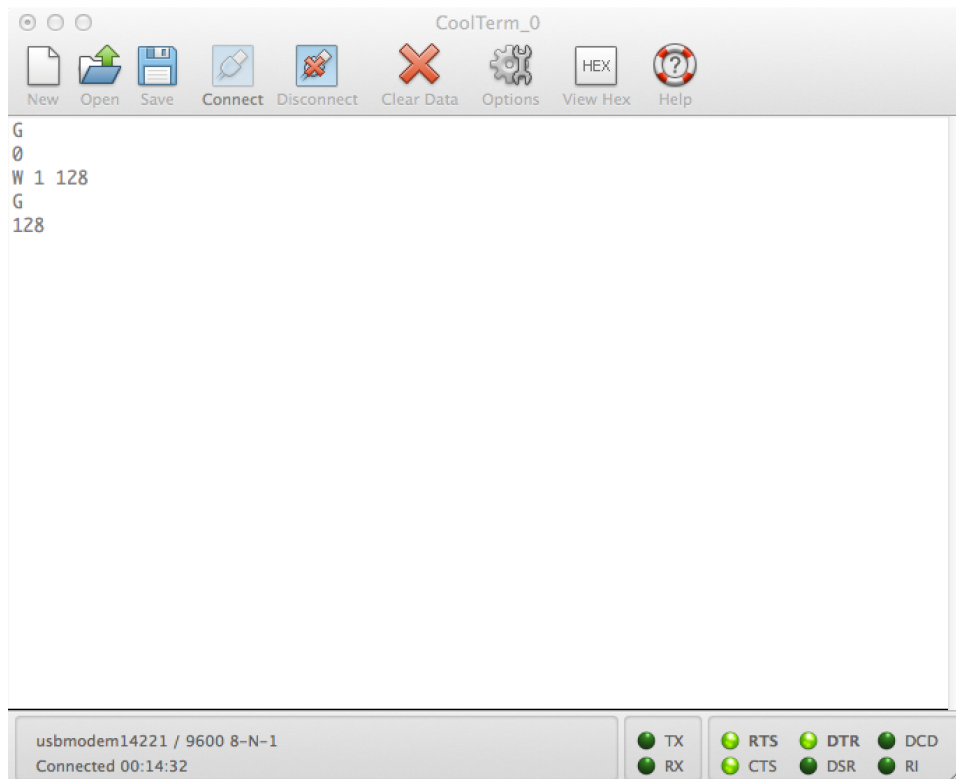
Example Serial Session

I²C Interface with Arduino

A micro-controller can access the Touch Potentiometer using an I²C interface. The Touch Potentiometer supports 7-bit addressing and a maximum transfer rate of 100 kHz. It utilizes two I²C addresses. The first address is used to directly access (get or set) the Touch Potentiometer value and the subsequent address is used for accessing control registers and other resources. The address can be changed. See Changing I²C Address for more information.
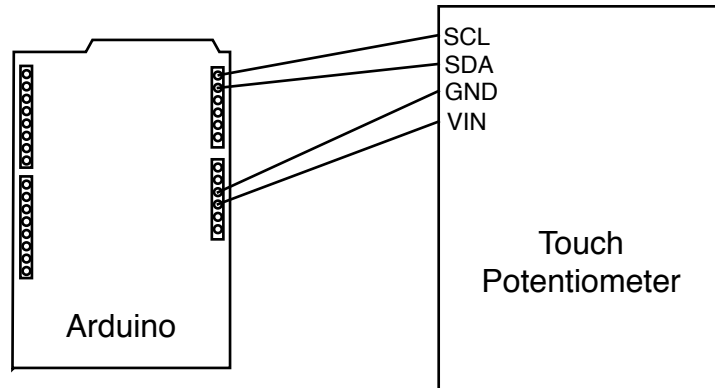
Arduino Connections

The Touch Potentiometer may be connected to a 5V Arduino I^{2}C peripheral (A4/A5) directly. Level translators should be used for 3.3V Arduino boards (or other 3.3V micro-controllers). The Touch Potentiometer activates weak pull-ups on its I^{2}C signals so pull-up resistors are not necessary for short connections (a few inches). Pull-up resistor values of 4.7 k-ohm to 10 k-ohm may be used.

Simple access routines are used to communicate with the Touch Potentiometer as shown in the following demonstration program. The following code segment may be cut&paste into the Arduino IDE. See Command List for more information about the Command interface.

```
/*
 * Simple Touch Potentiometer Example with Arduino
 *
 * Reads the pot value and controls the brightness of the Arduino LED on
 * Digital Pin 13.  Also logs new values to the serial port.  Utilizes
 * both the direct and indirect command interface forms.
 * Assumes Touch Pot is at I2C Address 8
 */
#include "Wire.h"

int i2cAddr = 8; // Direct access at i2cAddr, indirect registers at i2cAddr+1

uint8_t prevValue;
uint8_t curValue;

void setup() {
  Serial.begin(115200);
  Wire.begin();
  pinMode(13, OUTPUT);

  // Demonstrate access to Touch Potentiometer registers
  WriteTpReg(1, 128); // set to 50% by writing to register 1
  curValue = ReadTpReg(1); // read back value just set

  // Set Arduino LED PWM to match
  analogWrite(13, curValue);
  prevValue = curValue;
}

void loop() {
  delay(50);  // Read ~20 times/second

  // Demonstrate direct access to Touch Potentiometer value
```

```
    curValue = ReadTpValue(); // faster I2C access than register read
    if (curValue != prevValue) {
      analogWrite(13, curValue);
      Serial.println(curValue);
      prevValue = curValue;
    }
}

// Write a Touch Potentiometer register
void WriteTpReg(uint8_t addr, uint8_t data) {
  Wire.beginTransmission(i2cAddr+1);
  Wire.write('W');
  Wire.write(addr);
  Wire.write(data);
  Wire.endTransmission();
}

// Get the Touch Potentiometer value
uint8_t ReadTpValue() {
  Wire.requestFrom(i2cAddr, 1);
  if (Wire.available()) {
    return Wire.read();
  } else {
    return 0;
  }
}

// Read a Touch Potentiometer register
uint8_t ReadTpReg(uint8_t addr) {
  Wire.beginTransmission(i2cAddr+1);
  Wire.write('R');
  Wire.write(addr);
  Wire.endTransmission();

  Wire.requestFrom(i2cAddr+1, 1);
  if (Wire.available()) {
    return Wire.read();
  } else {
    return 0;
  }
}
```

Analog Interface as Volume Control

The Touch Potentiometer's AD5262 IC provides two separate digitally controlled analog potentiometer circuits. They may be used separately (e.g. for stereo volume controls) or ganged together (to simulate a 10k-ohm potentiometer) as shown in the following simple single-channel amplifier using the common LM386 amplifier IC. The analog outputs may be configured to vary linearly or logarithmically. The Logarithmic output is commonly used for audio level applications. See Analog Output for more information.
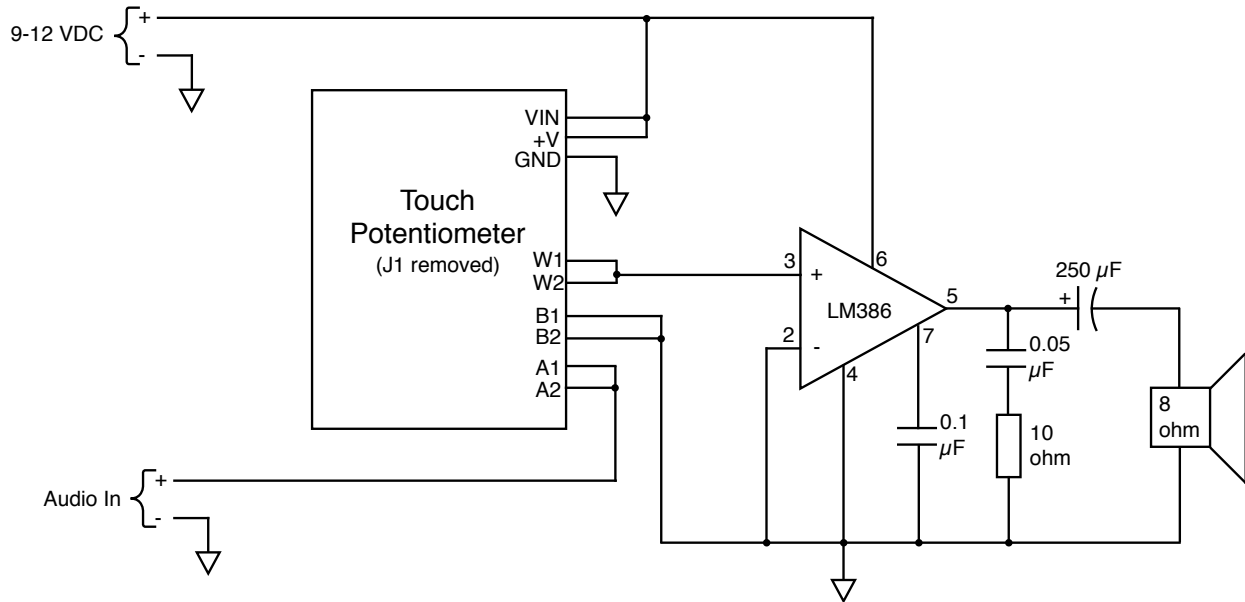
Simple Audio Amplifier

In this example the Touch Potentiometer replaces a logarithmic 10 k-ohm potentiometer to control the volume. In addition to providing a direct user control, a computer or micro-controller can also adjust the volume through the serial or I2C interfaces. Note that J1 is removed so that V+ can be connected directly to the analog power supply. This reduces noise from the digital components in the Touch Potentiometer coupling into the audio signal. Additional filtering circuitry on the input of the VIN voltage regulator may be necessary for high-performance systems.

Operation

System Description

Touch Sensor

The Touch Sensor is implemented using two interleaved copper pads on the top surface of the PCB connected to the on-board micro-controller. They are used as a variable capacitance in an oscillator circuit. An algorithm running on the micro-controller measures the oscillator frequency, which varies as the sensor capacitance varies (due to touch), and computes a touch location along the length of both sensors with a resolution of 8-bits. The frequency of the oscillator decreases as the capacitance increases with touch. The sensors are sampled at about 30.52 Hz and the frequency is represented as a count obtained during a 8.192 mSec sample period.

The algorithm maintains a dynamic baseline corresponding to no touch that is initialized during the initial calibration at power-up. Touch sensor calibration parameters are stored in an on-board EEPROM. Some of the parameters specify a minimum count for the initial calibration. The Touch Potentiometer will remain in initial calibration until the count for both sensors exceeds these parameters (for example, the sensor is being touched during initial calibration). Other parameters specify delta values from the baseline used to identify when a valid touch is present. These parameters may be changed by the user through either the Command Interface or a built-in procedure to recalibrate the Touch Potentiometer parameters for various conditions. See Appendix B for more information.

The Touch Potentiometer is designed to be used without any panel or with a thin panel adhesively attached to the PCB (up to approximately 0.5 mm thick). The sensitivity is reduced as the thickness of a cover increases. The baseline frequency also is reduced. The panel must not be conductive. Higher dielectric materials work better than lower dielectric materials. The default touchpad parameter set has been verified with polystyrene panel material attached using double-sticky carpet tape. Other professional pressure sensitive adhesives such as 3M 467 or 468 should work as well. The adhesive should cover the entire surface of the sensor evenly.
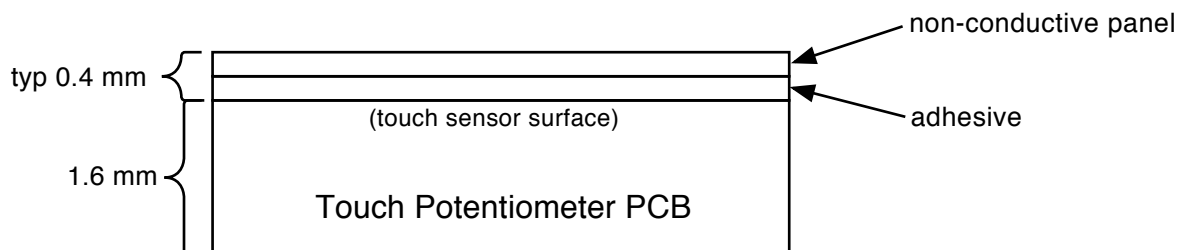
Typical panel stack-up

LED Display

The eight LED display is used to display the current Touch Potentiometer value. It may also be configured to display any pattern loaded by a host system through the Command Interface. The power-up configuration is loaded from EEPROM.

Touch Potentiometer value LED Display

The value may be displayed in three different formats. The format may be changed by writing to the Control register described in the section Control Registers.

1. Linear Display: A bright spot that is next to the current value is simulated by varying the brightness of all LEDs. LEDs near the current value are brightest and LEDs further away are dimmer. The entire display fades to a much dimmer level after 10 seconds of inactivity. It returns to full brightness when the value is changed, either through touch or by a host command. This format is the factory default.

2. Bar Graph: All LEDs from the lowest up to the LED nearest to the current value are lit. LEDs above these are off.

3. Single Dot: The LED nearest to the current value is lit. All other LEDs are off.

The brightness of Bar Graph and Single Dot display formats is configurable to one of four brightness levels.

User LED Display

The LED display may also be controlled manually with an 8-bit pattern written to the UserLedValue register. The brightness of the display is configurable to one of four brightness levels.

Host Interface

The Touch Potentiometer is a slave device. It responds to commands issued to it from a host controller. It interprets commands from either a serial or I^{2}C interface. Both interfaces are simultaneously operational. The Touch Potentiometer is capable of receiving commands 250 mSec after being powered up.

Unknown or illegal commands are ignored. Some commands generate a response. Responses are transmitted back to the host automatically if the command originates on the serial interface. They must be read by the host if the command originates on the I^{2}C interface.

Commands are 1-byte (8-bits in length). They have ASCII representations that are an abbreviation of their function. Commands are followed by zero, one or two arguments representing 8-bit numeric quantities.

Responses are a single 8-bit numeric values.

Commands and responses are encoded slightly differently for serial and I^{2}C communication.

Serial Interface

Commands and arguments sent through the serial interface are encoded as ASCII values. The commands are used to read the current potentiometer value, access a set of control registers and EEPROM storage locations. Commands are a single ASCII character with values between 'A' and 'Z'. Arguments are one to three ASCII numbers "0" through "255" representing an 8-bit value. Commands are issued in Command Sequences. A Command Sequence consists of the command value followed by any required argument terminated with the Carriage Return character (0x0D). Space characters may be included between the command and argument or between the argument and the Carriage Return (<CR>). Space characters must be included between arguments. Any other characters or additional arguments invalidate the Command Sequence (up to the next Carriage Return).

Responses consist of an ASCII number ("0" through "255") terminated with the Carriage Return character.

For example the ASCII character 'G' is used for the Get Current Value command. Sending the command

   G<CR>

or

   G <CR>

generates a response with the form

   128 <CR>

where the value "128" is the current value of the Touch Potentiometer (mid-point).

The Touch Potentiometer can buffer multiple serial commands and arguments in an internal 16-byte FIFO. It processes one command at a time (including transmission of a required response). It will drop commands when the FIFO is full.

Serial Jabber

The Touch Potentiometer may be configured to automatically transmit the potentiometer value through the serial port to host interface when it changes. The value consists of an ASCII number ("0" through "255") terminated with the Carriage Return character. The Jabber function is controlled by the Control register.

The maximum rate of values transmitted is approximately 30 values/second.

Host Interface software must be capable of differentiating automatically transmitted potentiometer values and responses to commands that return data. Typically the Jabber function is disabled before sending commands that generate a response.

I²C Interface

The Touch Potentiometer implements a simple 7-bit slave I²C device. It implements two consecutive addresses. The first address is denoted as the base address and is loaded from EEPROM storage at reset. The base address may be changed via the touch sensor or by a command to change the EEPROM location. The Touch Potentiometer implements only the mandatory I²C slave functionality described in the NXP Semiconductor document "UM10204 I²C-bus specification and user manual". These functions include START condition, STOP condition, Acknowledge and 7-bit slave address. It does not support General Call address, 10-bit slave address, Software reset or Device ID. The maximum bit rate is 100 kHz.

The default base address is 8. The base address provides direct access to the current potentiometer value. The next address provides indirect access to the potentiometer value, control registers and EEPROM storage.

Direct Access (base address)

Reading and writing this address provides direct (1-byte) access to the current potentiometer value.

A write to this location consists of the I²C address byte (with bit 0 low to signify a write) and a data byte to set the value. The Touch Potentiometer changes its internal value to the value of the data byte.

A read from this location consists of the I²C address byte (with bit 0 high to signify a read) and a data byte sourced by the Touch Potentiometer with the current potentiometer value.

Indirect Access (base address + 1)

Reading and writing this address provides indirect (multiple transaction) access to control registers and EEPROM storage for configuration of the Touch Potentiometer. The command encoding is the same as the Serial Interface and the indirect access mechanism also provides a command to read the current potentiometer value.

Command Sequences consist of 1-, 2- or 3-byte I²C write sequences (the command byte followed by any required argument byte) followed by the STOP condition.

The host controller must issue a read of one data byte for commands that generate a response. The read should be issued following the write and before subsequent commands. The Touch Potentiometer may stretch the clock on the SCL line during the read while it fetches data bytes internally to satisfy the read. It will return the value 0x00 for host controller I²C reads that request more data than the is contained in the response or for reads without a prior command.

The example shown in the Serial Interface section above would consist of the following bytes transmitted on the I²C interface for the command.

    <I2cAddress+W><0x47>

The following response data would be read (the host controller reads one byte).

    <I2cAddress+R><0x80>

The Touch Potentiometer can buffer multiple I²C commands and arguments in an internal 16-byte FIFO. However it can only process one command at a time that requires a response. This is because it resets the internal buffer it uses to hold responses each time a command is received from the I²C interface. The host controller must execute a read and obtain the response data for any command that generates a response before issuing another command.

Changing I²C Address
The Base Address may be changed to any even value from 0 to 126 (a total of 64 possible addresses) allowing multiple Touch Potentiometers to operate on the same I²C bus.

The Base Address may be changed by writing to EEPROM location 1 or manually using the touch sensor. The manual method is described in this section.

Press and release the push button PB1 three times quickly (within 2 seconds). This causes the Touch Potentiometer to enter I²C Address set mode at the end of the 2-second period. It blinks all LEDs quickly three times and then displays the current I²C address in binary form on the LED display (the lowest LED is bit 0). For example the default Base Address of 8 will light the 4th LED from the bottom.

Slide a finger up and down the touch sensor. This will change the Base Address displayed on the LED display between the values of 0 and 126. Select the desired address.

Press and release the push button again. This will store the new address in EEPROM and reconfigure the Touch Potentiometer with the address.

Bit Value
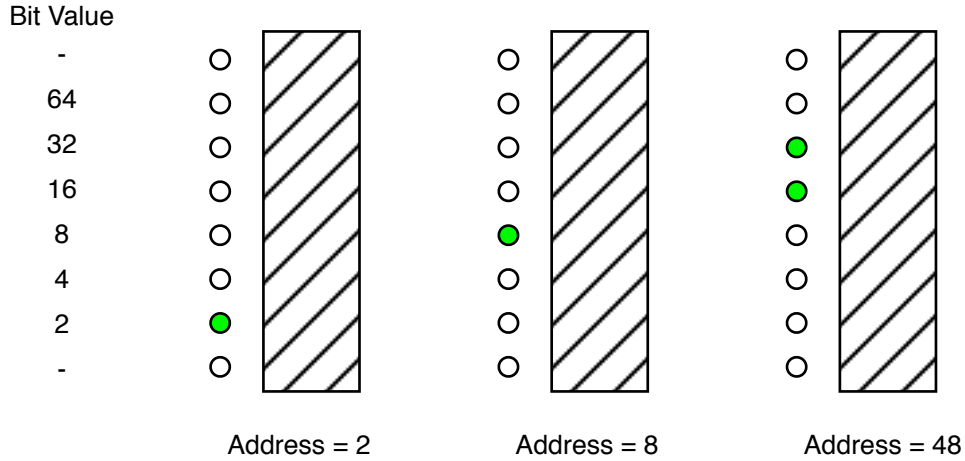
Example I²C Addresses

The I²C Address set mode automatically times-out after 30 seconds of inactivity without changing the I²C Base Address.

PWM Output

The PWM Output has 10-bit resolution. The Touch Potentiometer supports two different transfer functions between the 8-bit potentiometer value and PWM Output.

1. Linear: PWM bits [9:2] are set to the 8-bit potentiometer value. PWM bits[1:0] are set to zero.

2. Non-linear: PWM bits [9:0] are set by the transfer function $y = x^{1.53}$. This transfer function is designed to reduce changes at the low-end. It is optimized for translation between 8 and 10-bits assuring a 1 count change for the first few values at the low-end. It is designed to improve the perceived brightness changes in applications such as LED dimming.
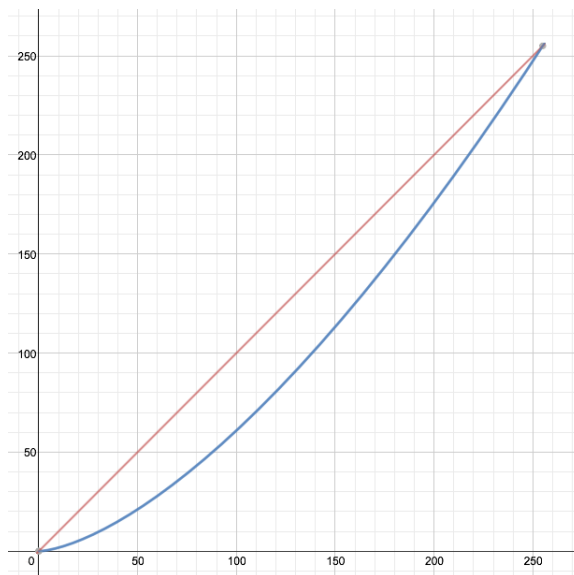
PWM Output Transfer Functions

The transfer function is configured from EEPROM at power-up and may be changed by writing to the Control register described in the section Control Registers.

Analog Output

The analog output has 8-bit resolution. The AD5262 presents a nominal resistance of 20 k-ohms between the A and B terminals. The wiper output may be set to one of 256 values. Each value count is approximately 78 ohms. The wiper circuitry has a typical inherent 60 ohm resistance. Resistance from terminal A to the Wiper decreases with increasing Touch Potentiometer values (typical values: 60-, 138-, 216 ohms, and so on up to approximately 19,982 ohms). Resistance from terminal B to the Wiper increases with increasing Touch Potentiometer values. See the AD5262 specification for more information.

The AD5262 IC is a linear device. The wiper resistance changes linearly as a function of the 8-bit value loaded into the part by the micro-controller. Some analog applications, such as volume control, typically use logarithmic potentiometers ("audio taper") to account for human perception of loudness. The Touch Potentiometer supports the AD5262 in a linear mode as well as a simulated logarithmic mode configured at power-on from EEPROM. This transfer function may also be changed by writing to the Control register described in Control Registers. The simulated logarithmic transfer function takes the 8-bit linear potentiometer value and converts it to a 5-bit (32-position) value with a 41 dB range using the function $y = 8.9125 \times 10^{-3} * e^{(4.7203*x)}$. A mathematical treatment of the algorithm can be found on the web page Dr. Lex Site: Programming Volume Controls. The parameters for this equation were chosen to maximize the dynamic range and utilize as much of the AD5262 resistive range as possible. There are alternative methods for taking the output of a linear potentiometer and simulating a logarithmic device that may suite individual requirements better. One of these is outlined in Appendix C.
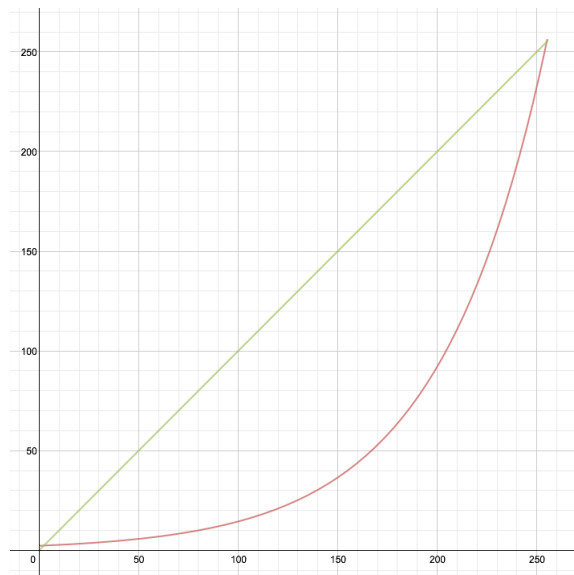
Analog Output Transfer Functions

Control Registers

The Touch Potentiometer implements a set of addressed 8-bit registers accessed via the Serial and I²C interfaces to query status and set operational parameters. Commands to access the registers include the register address. Commands with an address value outside the valid range of addresses are ignored. Some registers are read-only (RO). Write accesses to these registers have no effect.

Address	Register	Access	Description
0	Version	RO	Firmware revision Bits 7:4 - Major revision. Incremented when functionality is added or changed. Bits 3:0 - Minor revision. Incremented for bug fixes.
1	CurPotValue	RW	Current potentiometer value. Power-on value is set to zero.
2	Status	RO	Device Status Bit 7 - Reserved, reads as 0. Bit 6 - Button currently pressed Bit 5 - Touchpad calibration procedure in progress Bit 4 - I²C Address set mode in progress Bit 3 - Sensor 2 (upper) Valid Touch detected Bit 2 - Sensor 1 (lower) Valid Touch detected Bit 1 - Sensor 2 Initial Cal in progress Bit 0 - Sensor 1 Initial Cal in progress
3	Control	RW	Device Configuration Bit 7 - Enable Serial Jabber. Setting this bit enables the Touch Potentiometer to automatically transmit updated values through the serial interface. Bit 6 - Enable Interpolation. Setting this bit enables the Touch Potentiometer to interpolate the potentiometer value between the current value and the value detected upon first touch over a short period of time (50-200mSec). Clearing this bit causes the potentiometer value to be updated immediately. Bit 5 - PWM Output Configuration. Setting this bit configures the non-linear transfer function. Clearing this bit configures the linear transfer function. Bit 4 - Analog Output Configuration. Setting this bit configures the simulated logarithmic transfer function. Clearing this bit configures the linear transfer function. Bit 3:2 - LED Display Mode 00 - Linear Display 01 - User Display 10 - Bar-graph Display 11 - Single-dot Display Bit 1:0 - LED Brightness (User, Bar-graph, Single-dot) 00 - 25% Full Brightness 01 - 50% Full Brightness 10 - 75% Full Brightness 11 - Full Brightness Power-on value is set from EEPROM.
4	UserLedValue	RW	Bit-mask for controlling the LED display when the LED Display Mode is set to User Display. Ignored in other LED Display modes. Power-on value is zero.

Sparkfun Touch Potentiometer User Manual   26

Address	Register	Access	Description
5	Sensor1CountH	RO	High 8-bits of the raw lower capacitive sensor value.
6	Sensor1CountL	RO	Low 8-bits of the raw lower capacitive sensor value.
7	Sensor2CountH	RO	High 8-bits of the raw upper capacitive sensor value.
8	Sensor2CountL	RO	Low 8-bits of the raw upper capacitive sensor value.

EEPROM

The Touch Potentiometer allows access to all 256 8-bit EEPROM locations in the micro-controller. Locations described below contain operational parameters that are used to configure operation at power-up. The remaining locations are accessible through the host interface but unused by the firmware. They may be used to store application-specific data. Future firmware revisions may use additional EEPROM locations so application-specific data should be stored in the middle of the EEPROM range.

Changing the EEPROM values used to configure the capacitive sensors may render these sensors inoperable. However the factory default values, listed below, may be reloaded to restore original operation. In addition, a backup set of parameters for the operational values is stored in the last locations of the EEPROM and may be used to restore the factory default configuration.

Typically the only EEPROM values that are changed are the Control Register default and the I^{2}C Base Address.

The Minimum PowerOn Count and Valid Touch Count Delta are the EEPROM values are usually modified when adjusting the capacitive sensors to account for a specific label. It should not be necessary to change any other operational EEPROM values.

Address	Default Value	Description
0	0x40	Control Register power-on value
1	0x08	I^{2}C Base Address power-on value Bit 7 - Unused - masked off Bit 6:2 - Address bits 6:2 Bit 0 - Unused - masked off
2	0x0D (Note 1)	Sensor 1 Minimum PowerOn Count High Byte The Minimum PowerOn Counts set the initial calibration threshold. Each sensor must generate an average of several values greater than the associated Minimum PowerOn Count to exit the initial calibration and enter normal operation.
3	0xDE (Note 1)	Sensor 1 Minimum PowerOn Count Low Byte
4	0x0E (Note 1)	Sensor 2 Minimum PowerOn Count High Byte
5	0x42 (Note 1)	Sensor 2 Minimum PowerOn Count Low Byte

Address	Default Value	Description
6	0x00 (Note 1)	Sensor 1 Valid Touch Count Delta High Byte The Valid Touch Count Delta values specify the difference in counts accumulated during a sensor sample period from the untouched average to signify a valid touch. These values may vary with different label materials.
7	0x80 (Note 1)	Sensor 1 Valid Touch Count Delta Low Byte
8	0x00 (Note 1)	Sensor 2 Valid Touch Count Delta High Byte
9	0x78 (Note 1)	Sensor 2 Valid Touch Count Delta Low Byte
10	0x00	Sensor Idle Count Delta High Byte The Idle Count Delta value specifies the different in counts accumulated during a sensor sample period from the untouched average at which point the values are no longer incorporated into the untouched average.
11	0x10	Sensor Idle Count Delta Low Byte
12	0x06	Internal algorithm use. Do not change this without consultation from danjuliodesigns LLC.
13	0x00	Sensor Guard Band. Specifies a region at the top and bottom of the computed linear position to consider "0" at the bottom of the sensor and "255" at the top of the sensor. The remaining portion is rescaled between those values. The Sensor Guard Band has the effect of making the Full Off and Full On touch positions larger at the expense of the linear region in between.
14	0xC4	Internal algorithm use. Do not change this without consultation from danjuliodesigns LLC.
15	0x86	Internal algorithm use. Do not change this without consultation from danjuliodesigns LLC.
16	0x88	Internal algorithm use. Do not change this without consultation from danjuliodesigns LLC.
17	0x16	Internal algorithm use. Do not change this without consultation from danjuliodesigns LLC.
18 - 236	0xFF	Unused. Available for user data.
237	-	Backup value set checksum. This value is the sum of values in locations 238 - 255. It is used by software to verify the integrity of the backup value set. It is unused by the firmware and should not be modified by the user unless the user is modifying one of locations 238 - 255 (not recommended).

| Address	Default Value	Description
238 - 255	-	Backup value set. Contains, in reverse order, a backup set of values for locations 0 - 17 (Location 255 contains a backup value for location 0, and so on). These values generally should not be modified by the user. They are designed to be used to recover the operational values. The arduino demo sketch "tp_test_sketch" and the utility program "tputil" access the locations when restoring the factory default operating configuration.

Notes

1. These values represent calibration data that supports most touch potentiometers. These values are adjusted during a factory calibration procedure and differ from what is listed here. The calibration values in these locations are also stored in the set of backup values at the end of the EEPROM.

Built-in Parameter Calibration Procedure

The Touch Potentiometer includes a built-in calibration procedure to adjust the operational and EEPROM Minimum Power-on Count and Valid Touch Delta Count parameters, for example after adding a label. It is not typically necessary to recalibration the Touch Potentiometer. The built-in calibration procedure is initiated by pressing the button four times quickly (within 2 seconds). It is initiated at the end of the 2 second period. It requires fifteen (15) seconds split into three 5-second phases indicated by patterns on the LEDs. You may exit the procedure at any time without updating by pressing the button once.

The procedure is described in Appendix B "Manual Calibration". The three phases are as follows. The Touch Potentiometer collects a running average set of values from the sensors and uses the average value at the end of each phase to compute the values for that phase.

1. Idle (no-touch) Sensor Counts: The Idle Average values are saved at the end of phase 1. The middle six LEDs are lit at the beginning of this phase indicating the Touch Potentiometer should not be touched during this phase. Make sure that nothing is touching either sensor.

2. Sensor 1 Touched Counts: The values for each sensor when Sensor 1 is touched and Sensor 2 is untouched (touching the bottom end of the slider) are saved at the end of phase 2. The bottom LED is lit at the beginning of this phase indicating that you should immediately touch the Sensor 1 landing area (next to the lit LED) and hold your touch until the phase is complete. You should attempt to only touch Sensor 1 (the bottom sensor) and not touch any of Sensor 2.

3. Sensor 2 Touched Counts: The values for each sensor when Sensor 2 is touched and Sensor 1 is untouched (touching the top end of the slider) are saved at the end of phase 2. The top LED is lit at the beginning of this phase indicating that you should immediately touch the Sensor 2 landing area (next to the lit LED) and hold your touch until the phase is complete. You should attempt to only touch Sensor 2 (the top sensor) and not touch any of Sensor 1.

Phase 1
(Don't Touch)

Phase 2
(Touch Lower)
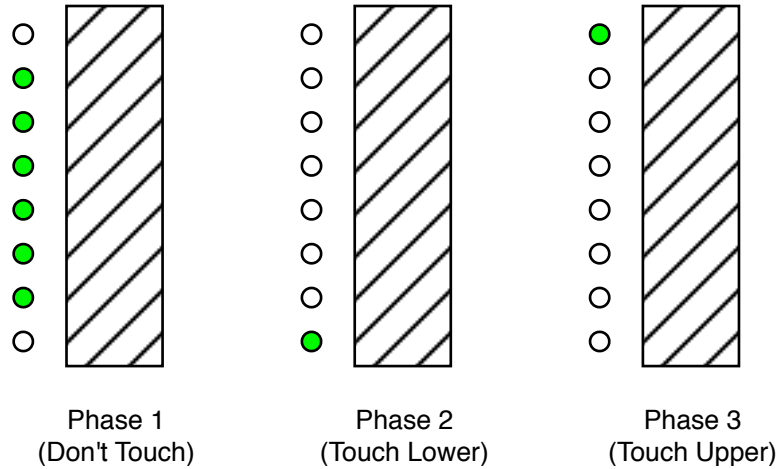
Phase 3
(Touch Upper)

Calibration Procedure Phases

The values collected in all three phases are validated at the end of phase 3. Both the Power-on Count and Valid Touch Delta Count operational parameters and operational EEPROM locations are updated if they pass the validation checks. In this case the Touch Potentiometer enters the initial calibration phase (four LEDs lit) waiting for you to release your touch from the top of the slider. The Touch Potentiometer enters normal operation again without updating any operational parameters or EEPROM location if the validation check fails. The validation checks mail fail for the following reasons. Please see Appendix B for more information about the nomenclature.

1. Sensor2DeltaA is greater than 25% of Sensor1DeltaA. This may occur if Sensor 2 was being partially touched during Phase 2.

2. Sensor1DeltaB is greater than 25% of Sensor2DeltaB. This may occur if Sensor 1 was being partially touched during Phase 3.

3. Any of the Sensor Touched Counts is greater than the Idle Average. This may occur if excessive noise is coupled into the sensor while being touched (for example 50 or 60 Hz noise). This typically happens if there is a ground loop in the system or a power-supply with inadequate filtering is used.

Note that the Backup EEPROM locations are not updated by this procedure and may be used to recover the factory-default configuration.

Command Interface

Command List

The Touch Potentiometer implements the following commands, summarized in the following table and described more fully in the following sections.

Command	ASCII Command Value	Hex Command Value	Command Argument	Response	Description
Get current value	G	0x47	-	8-bit potentiometer value	Gets the current potentiometer value.
Read Control Register	R	0x52	8-bit register address	8-bit register value	Read a control register
Write Control Register	W	0x57	8-bit register address, 8-bit register value	-	Write a control register
Set EEPROM location	S	0x53	8-bit EEPROM address, 8-bit value	-	Write to an EEPROM location
Query EEPROM location	Q	0x51	8-bit EEPROM address	8-bit value	Read an EEPROM location

Touch Potentiometer Command List

8-bit values are encoded as an ASCII string "0" to "255" for the serial interface.

Get Current Value

Description
Request the current potentiometer value.

Serial command
G<CR>

I²C commands
Write: <I²C Base Addr + 1><0x47>

Read: <I²C Base Addr + 1>

Argument
None

Serial response
Returns 1 8-bit byte with the current potentiometer value encoded as an ASCII string "0" to "255" followed by <CR>.

I²C response
Read 1 8-bit byte with the current potentiometer value.

Notes
None

Read Control Register

Description
Read a control register value.

Serial command
R<N><CR>

I²C command
Write: <I²C Base Addr + 1><0x52><N>

Read: <I²C Base Addr + 1>

Argument
<N> is the 8-bit address of the control register to read

Serial response
Returns 1 8-bit byte with the control register value encoded as an ASCII string "0" to "255" followed by <CR>.

I²C response
Read 1 8-bit byte with the control register value.

Notes
1. Read requests with a value of <N> that is outside the valid address ranges are ignored. No data is returned to the serial port and the I²C Read returns a value of 0.

Write Control Register

Description

Write an 8-bit value to a control register.

Serial command

W<N> <V><CR>

I²C command

Write: <I²C Base Addr + 1><0x57><N><V>

Argument

<N> is the 8-bit address of the control register to write.

<V> is the 8-bit data to write.

Serial response

None

I²C response

None

Notes

1. A space character is required between <N> and <V> for commands sent to the serial interface.

2. Write requests with a value of <N> that is outside the valid address ranges are ignored.

3. Write requests to Read Only control registers are ignored.

Set EEPROM Location

Description

Write an 8-bit value to an EEPROM location.

Serial command

S<N> <V><CR>

I²C command

Write: <I²C Base Addr + 1><0x53><N><V>

Argument

<N> is the 8-bit address of the EEPROM location to write.

<V> is the 8-bit data to write.

Serial response

None

I²C response

None

Notes

1.A space character is required between <N> and <V> for commands sent to the serial interface.

2.The host interface should delay approximately 5 mSec after transmitting a Set EEPROM Location command before sending another command to allow the micro-controller to execute the update of EEPROM storage.

Query EEPROM Location

Description

Read an EEPROM location value.

Serial command

Q<N><CR>

I2C command

Write: <I2C Base Addr + 1><0x51><N>

Read: <I2C Base Addr + 1>

Argument

<N> is the 8-bit address of the EEPROM location to read

Serial response

Returns 1 8-bit byte with the EEPROM location value encoded as an ASCII string "0" to "255" followed by <CR>.

I2C response

Read 1 8-bit byte with the EEPROM location value.

Notes

None

Appendix A: Troubleshooting

Symptom	Possible Causes	Action
LEDs are all off	No Power	Make sure the input voltage is between 4.6 and 12 volts, that the polarity is correct and that the power supply can supply enough current for the Touch Potentiometer.
	User Display	The UserLedValue register is cleared at power-up. A Touch Potentiometer that is configured to display the User Value at power-up will clear the LED display after the Initial Calibration is completed.
Device remains in Initial Calibration	An object is interfering with the capacitive sensors	Make sure you are not touching the capacitive sensors at power-up.
		Make sure that a conductive item, such a piece of metal, is not touching the capacitive sensors at power-up.
	Minimum PowerOn Counts are too high	Adjust the Minimum PowerOn Counts to be lower than an average of the sensor readings with no touch. See Appendix B for more information about calibrating the Touch Potentiometer.
AD5262 Integrated Circuit (U2) is hot	No Analog Power	The AD5262 can get hot when certain parasitic circuit paths are enabled when it has logic power but no power connection on V+ or V-. Make sure that V+ and V- are connected to a power supply outside the range of the analog voltage signals.

Symptom	Possible Causes	Action
No communication through the serial interface.	Signals reversed	Make sure that the TX and RX signals connect to the other device's RX and TX signals respectively.
	Incorrect serial interface configuration	Make sure the host system baud rate is set to 9600 and serial port settings are 8N1.
	Incorrect voltage levels	The Touch Potentiometer is designed to work with 5V host systems. It should not be used with 3.3V or RS232C systems without level shifters on the serial signals.
No communication through the Logic-level I^2C interface.	Incorrect I^2C interface configuration	Make sure the host interface is configured for 7-bit addressing, 100 kHz operation and is using the correct I^2C address.
	Missing pull-up resistors	Make sure the I^2C is properly pulled-up. The Touch Potentiometer weak pull-ups are good for short connections. Longer connections may require external pull-ups (4.7 - 10 k-ohm resistors).
	Unsupported I^2C commands	Make sure the host controller issues simple I^2C read and writes with 7-bit addressing. Make sure the writes are for the command and any required argument only. Make sure the reads are for one byte.
Receive unexpected 0x00 when reading the I^2C interface.	Host controller is performing an I^2C read that is larger than the data returned by the previous command.	Make sure the I^2C read only reads one byte.
No response for commands sent through the serial interface.	Extraneous characters in command (for example a <LF> character before the <CR> character).	Make sure the only characters being sent to the Touch Potentiometer include the ASCII command, optional space characters, numeric characters and the <CR> termination character.

| Symptom	Possible Causes	Action
Device becomes unresponsive when touched.	Excessive noise (e.g. 50 or 60 Hz) is being coupled into the sensors. This typically happens with poor quality power supplies (e.g. lacking input filtering) or a ground loop condition (the device is connected both to a power supply and a computer with its own power supply).	Make sure the power supply has adequate filtering, both on the AC input side and DC output side.
Device fails a calibration procedure with sensor count values that are greater than the idle (untouched) count values.		Make sure there are no ground loops in the system. Isolate digital connections to one of the power domains if it is necessary to operate in an environment with two power supplies.
Output varies slowly when being touched in one location.		

Appendix B: Manual Calibration

The default configuration parameters in EEPROM should suffice for many applications of the Touch Potentiometer. The procedure described in this section may be used to adjust the configuration parameters for situations where the default values do not provide adequate performance and the built-in calibration function cannot be used. An example of this situation might occur when the touchpad is enclosed with a thicker label. The default values may make it difficult to sense changes in touch position or the Touch Potentiometer may remain in the Initial Calibration. This procedure sets the Minimum PowerOn Counts and Valid Touch Count Delta values in the EEPROM for both sensors. This is the same procedure that is implemented by the built-in parameter calibration procedure.

The procedure requires reading the raw Sensor Count Registers (Sensor1CountH, Sensor1CountL, Sensor2CountH and Sensor2CountL) and concatenating the high and low halves together to get 16-bit values for Sensor1Count and Sensor2Count. It is ideal to average together several readings although it is possible to execute the procedure with only a single reading at the points where the Touch Potentiometer raw values are required.

Sensor1Count = (Sensor1CountH << 8) | Sensor1CountL

Sensor2Count = (Sensor2CountH << 8) | Sensor2CountL

The registers should be read sequentially. The firmware does not ensure atomic access to these values. It is possible for the values to be updated between access of the high and low halves. For this reason the data should be validated prior to use. A simple mechanism is to read the pair several times, say 5 times, through out the high and low values and average the remaining 3 values.

The 16-bit Sensor Counts are the result of sampling the capacitive sensor oscillator each sample period. As shown in the diagram below, the count is highest in the untouched state and is lower as capacitive loads are added to the sensor. The goal of the manual calibration procedure is to determine the average Sensor Count values for the no-touch and touch conditions and use those to compute the Minimum PowerOn and Valid Touch Count Delta values.

Sensor Operation
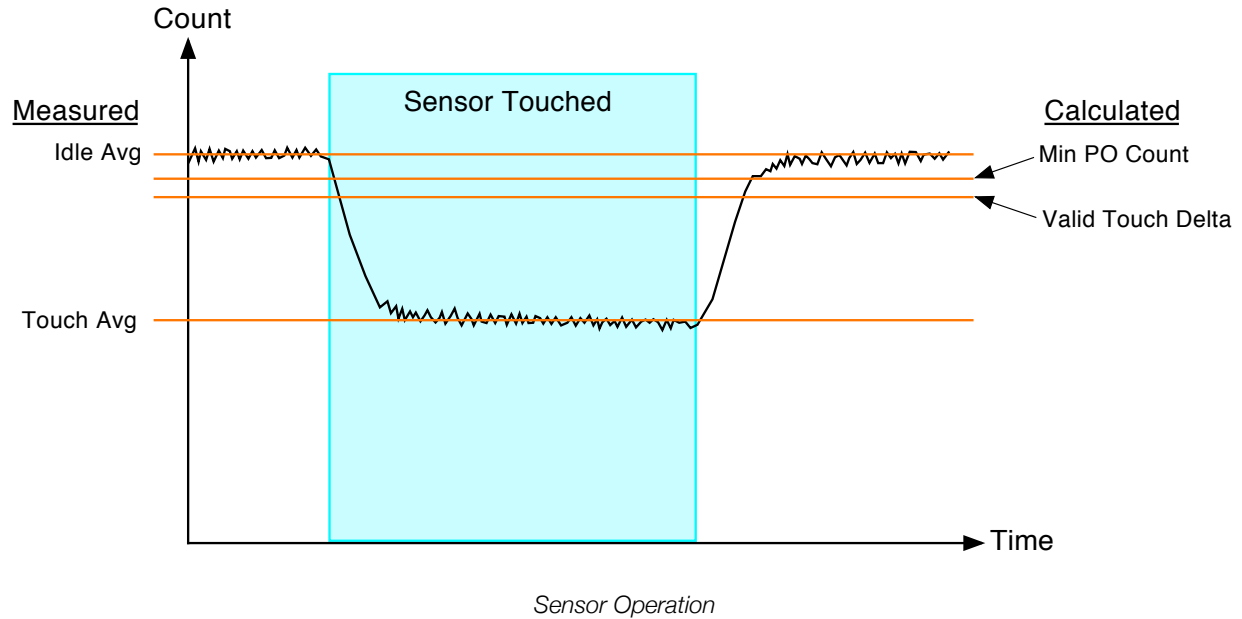
Note: The Arduino Leonardo sketch "tp_test_sketch", available on the danjuliodesigns.com website, implements this calibration procedure and may be used as-is or modified to suit the user's needs.

Procedure

1. Measure the idle (no-touch) Sensor Counts: Sensor1Idle, Sensor2Idle.

    a. Position the Touch Potentiometer so the sensors are not being touched.

    b. Read the raw Sensor Count values as described above. Store the Sensor 1 (low sensor) value as Sensor1Idle. Store the Sensor 2 (high sensor) value as Sensor2Idle.

2. Measure the Sensor 1 touch Sensor Count: Sensor1TouchA, Sensor2TouchA.

    a. Touch the Sensor 1 landing area. The landing area is at the bottom of the touch area, below the interleaved portion of the sensor array (next to the bottom-most LED). It is important to try not to touch the interleaved portion at all because you are trying to provide the maximum capacitive load to Sensor 1 and the minimum capacitive load to Sensor 2.

    b. While touching the Sensor 1 landing area, read the raw Sensor Count values as described above. Store the Sensor 1 value as Sensor1TouchA. Store the Sensor 2 value as Sensor2TouchA.

3. Measure the Sensor 2 touch Sensor Count: Sensor1TouchB, Sensor2TouchB.

    a. Touch the Sensor 2 landing area. The landing area is at the top of the touch area, above the interleaved portion of the sensor array (next to the top-most LED). It is important to try not to touch the interleaved portion at all because you are trying to provide the maximum capacitive load to Sensor 2 and the minimum capacitive load to Sensor 1.

    b. While touching the Sensor 2 landing area, read the raw Sensor Count values as described above. Store the Sensor 1 value as Sensor1TouchB. Store the Sensor 2 value as Sensor2TouchB.

4. Compute intermediate delta values: Sensor1DeltaA, Sensor1DeltaB, Sensor2DeltaA, Sensor2DeltaB

    a. Sensor1DeltaA = Sensor1Idle - Sensor1TouchA

    b. Sensor1DeltaB = Sensor1Idle - Sensor1TouchB

    c. Sensor2DeltaA = Sensor2Idle - Sensor2TouchA

    d. Sensor2DeltaB = Sensor2Idle - Sensor2TouchB

5. Validate the data. Repeat steps 1 - 4 if the data cannot be validated.

    a. Verify Sensor2DeltaA is less than 25% of Sensor1DeltaA

    b. Verify Sensor1DeltaB is less than 25% of Sensor2DeltaB

6. Compute the Minimum PowerOn Count values: MinPowerOn1Count, MinPowerOn2Count.

    a. MinPowerOn1Count = Sensor1Idle - (0.33 * Sensor1DeltaB)

    b. MinPowerOn2Count = Sensor2Idle - (0.33 * Sensor2DeltaA)

7. Compute the Valid Touch Count Delta values: ValidCountDelta1, Valid CountDelta2.

    a. ValidCountDelta1 = Sensor1DeltaB

    b. ValidCountDelta2 = Sensor2DeltaA

8. Store the values in EEPROM.

    a. Sensor 1 Minimum PowerOn Count High Byte = MinPowerOn1Count[15:8] (MinPowerOn1Count >> 8)

    b. Sensor 1 Minimum PowerOn Count Low Byte = MinPowerOn1Count[7:0] (MinPowerOn1Count & 0xFF)

    c. Sensor 2 Minimum PowerOn Count High Byte = MinPowerOn2Count[15:8] (MinPowerOn2Count >> 8)

    d. Sensor 2 Minimum PowerOn Count Low Byte = MinPowerOn2Count[7:0] (MinPowerOn2Count & 0xFF)

    e. Sensor 1 Valid Touch Count Delta High Byte = ValidCountDelta1[15:8] (ValidCountDelta1 >> 8)

    f. Sensor 1 Valid Touch Count Delta Low Byte = ValidCountDelta1[7:0] (ValidCountDelta1 & 0xFF)

    g. Sensor 2 Valid Touch Count Delta High Byte = ValidCountDelta2[15:8] (ValidCountDelta2 >> 8)

   h. Sensor 2 Valid Touch Count Delta Low Byte = ValidCountDelta2[7:0] (ValidCountDelta2 & 0xFF)

Notes about the calibration values:

1.The Minimum Power On Counts should be less than the Sensor Count values when there is any touch and less than the worst case untouched values.

2.The smaller the Valid Touch Count Delta values, the more influence the other sensor has when trying to turn the potentiometer fully on or fully off. The greater that Valid Touch Count Delta then the less sensitive to changes near the end.

Appendix C: Log Taper Circuit

The following circuit example is from an article written by Hank Zumbahlen of Analog Devices in the January 20, 2000 issues of EDN magazine titled "Tack a log taper onto a digital potentiometer". It is provided as an alternative method for converting the linear operation of the AD5262 into a simulated logarithmic device suitable for audio volume control.

A simulated logarithmic transfer function may be created by adding a pad resistor from the wiper of the AD5262 to one end as shown below in (a) with an equivalent circuit (b). The transfer function is shown by the following equation.
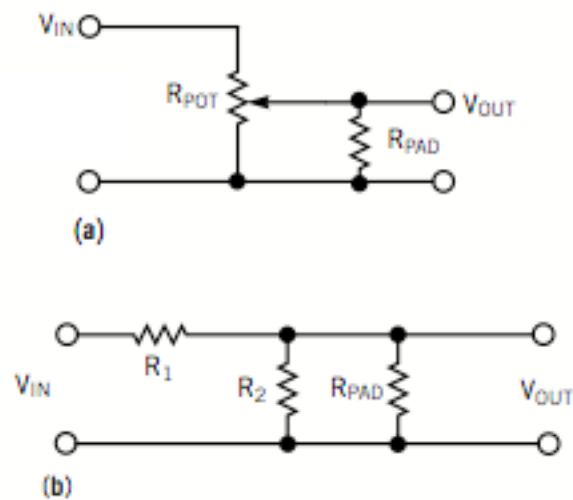
$$\frac{V_{OUT}}{V_{IN}} = \frac{R_2\|R_{PAD}}{R_1 + R_2\|R_{PAD}}$$

Circuit and transfer function

The transfer function is shown in the following graph. R = 20000/R_{PAD} (R=0.25=> R_{PAD} = 80 k-ohms; R=0.1 => R_{PAD} = 200 k-ohms; R=0.025 => R_{PAD} = 800 k-ohms). As can be seen, the taper isn't a precise number of dB per potentiometer count, but it does allow for better low-level performance.
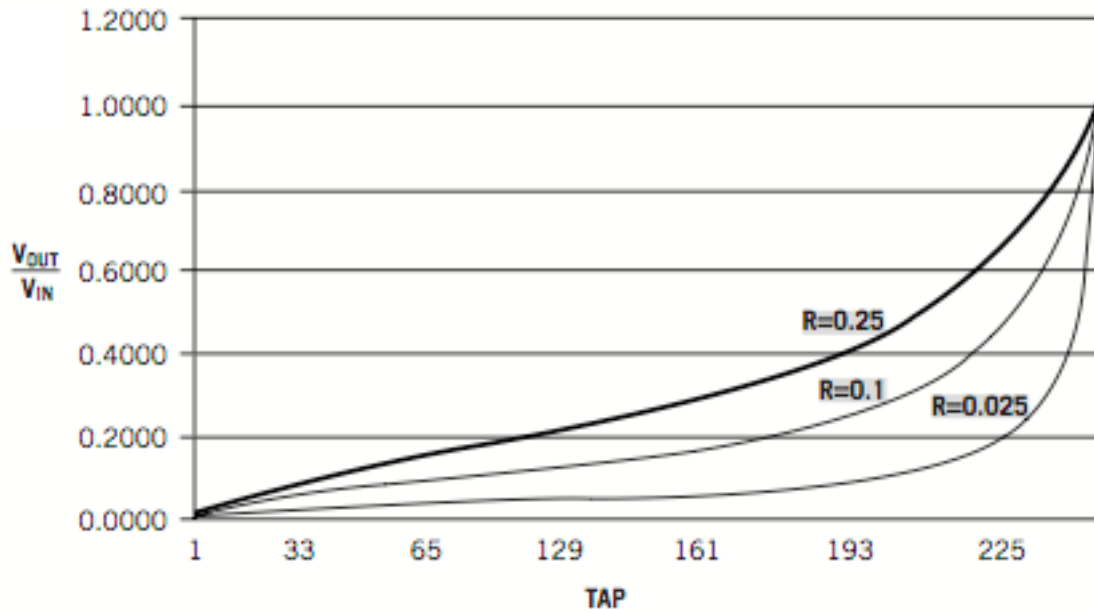
Transfer Function

The following commentary is taken from the article verbatim.

"You must address a couple of issues. The first is that the end-to-end resistance of the potentiometer changes with the digital code. It varies from the potentiometer resistance at one end (with the wiper at the lower end) to the value of the pad resistance in parallel with the potentiometer resistance at the other end. If you configure the circuit as a typical attenuator and drive it from a low-impedance source, the low pad resistance should not present a major problem. If, however, you are try- ing to obtain a set resistance value to determine a time constant (or any other application in which the resistor value is critical), this approach may not work well. The second issue involves over-voltage. The three terminals of the potentiometer can be anywhere within the supply range of the IC, which is 5V for the AD52XX and ±15V for the AD72XX family. If you apply over-voltage to one of the pins, even in a transient condition, the IC could latch up because of a parasitic substrate SCR."

Appendix D: Schematic