Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

AN60639

PSoC® 1 Piezo Keypad Implementation

Author: Andriy Ryshtun
Associated Project: Yes
Associated Part Family: CY8C21X34
Software Version: PSoC Designer™ 5.4 SP1 or later
Related Application Notes: None

To get the latest version of this application note, or the associated project file, click here.

AN60639 describes the implementation of keypad using piezo sensors with PSoC® 1. The solution described here provides reliable operation and an active time of up to two minutes.

Contents

1 Introduction....1
2 Getting Started....2
2.1 PSoC Designer....2
2.2 Code Examples....4
2.3 PSoC Designer Help....6
2.4 Technical Support....6
3 Theory of Operation....7
3.1 Positive Baseline Update....12
3.2 Negative Baseline Update....12
4 High Voltage Sensor Scan....13
5 Matrix Keypad Scan....15
6 Summary....15
A Appendix A....16
A.1 Board Schematic....16
A.2 Example Designs....17
Document History....18
Worldwide Sales and Design Support....19
Products....19
PSoC® Solutions....19
Cypress Developer Community....19
Technical Support....19

1 Introduction

Piezo sensors enable touch detection in the presence of metal surfaces, under water, in industrial environments with strong electrical fields, and in the chemical industries. Piezo sensors convert applied force into voltage. Therefore, they are not sensitive to external electrical and magnetic fields, water drops, or grounded surfaces. Because of these characteristics, piezo sensors are suitable to implement keypads in industrial environments.

This application note discusses a simple and cost effective solution to scan a piezo keypad without external components.

2 Getting Started

Cypress provides a wealth of data at www.cypress.com to help you to select the right PSoC device for your design, and to help you to quickly and effectively integrate the device into your design. For a comprehensive list of resources, see the knowledge base article How to Design with PSoC® 1, PowerPSoC®, and PLC – KBA88292. Following is an abbreviated list for PSoC 1:

- Overview: PSoC Portfolio, PSoC Roadmap
- Product Selectors: PSoC 1, PSoC 3, PSoC 4, PSoC 5LP
- In addition, PSoC Designer includes a device selection tool.
- Application notes: Cypress offers a large number of PSoC application notes covering a broad range of topics, from basic to advanced level. Recommended application notes for getting started with PSoC 1 are:
  - AN75320 - Getting Started with PSoC® 1.
  - AN2094 - PSoC® 1 - Getting Started with GPIO.
  - AN74170 - PSoC® 1 Analog Structure and Configuration with PSoC Designer™
  - AN2041 - Understanding PSoC® 1 Switched Capacitor Analog Blocks
  - AN2219 - PSoC® 1 Selecting Analog Ground and Reference

Note: For CY8C29X66 devices related application notes, click here.

- Development Kits:
  - CY3210-PSoCEval1 supports all PSoC 1 mixed-signal array families, including automotive, except the CY8C25/26xxx devices. The kit includes an LCD module, potentiometer, LEDs, and breadboarding space.
  - CY3214-PSoCEvalUSB features a development board for the CY8C24x94 PSoC device. Special features of the board include USB and CapSense development and debugging support.

Note: For CY8C29X66 devices related development kits, click here.

The MiniProg1 and MiniProg3 devices provide interfaces for flash programming and debug.

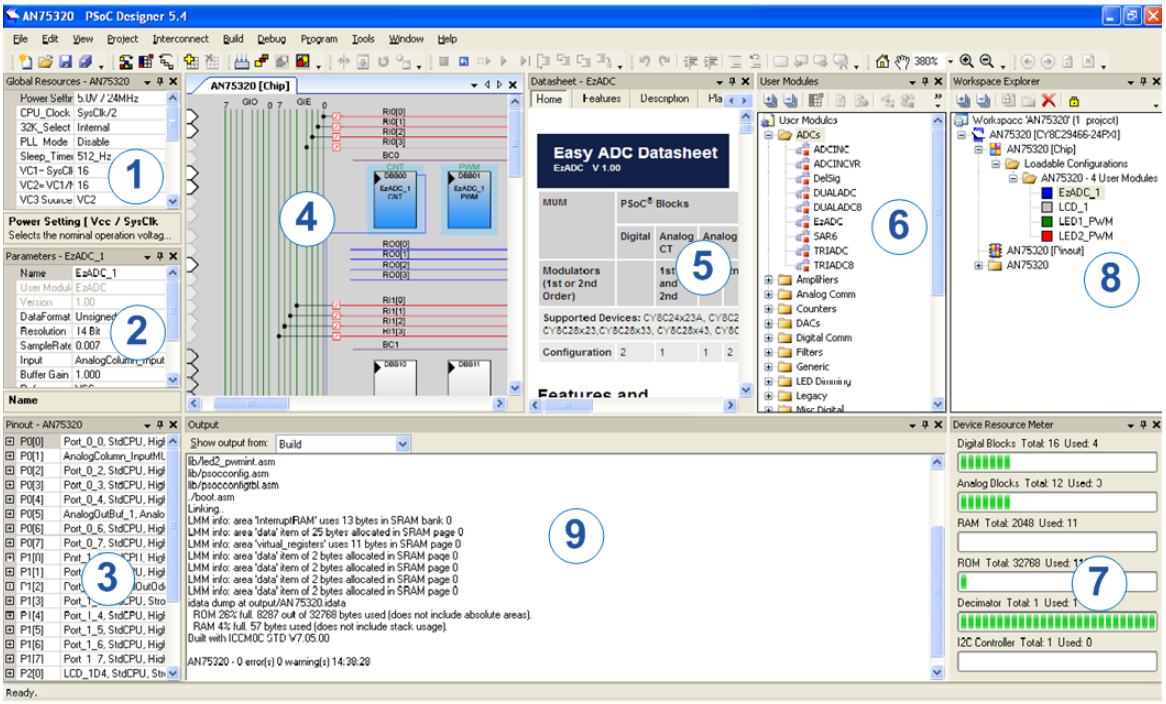
2.1 PSoC Designer

PSoC Designer is a free Windows-based Integrated Design Environment (IDE). Develop your applications using a library of pre-characterized analog and digital peripherals in a drag-and-drop design environment. Then, customize your design leveraging the dynamically generated API libraries of code. Figure 1 shows PSoC Designer windows. **Note:** This is not the default view.

1. Global Resources – all device hardware settings.
2. Parameters – the parameters of the currently selected User Modules.
3. Pinout – information related to device pins.
4. Chip-Level Editor – a diagram of the resources available on the selected chip.
5. Datasheet – the datasheet for the currently selected UM
6. User Modules – all available User Modules for the selected device.
7. Device Resource Meter – device resource usage for the current project configuration.
8. Workspace – a tree level diagram of files associated with the project.
9. Output – output from project build and debug operations.

Note: For detailed information on PSoC Designer, open PSoC Designer IDE, go to **Help** > **Documentation**, open the "Designer Specific Documents" folder, and open the "IDE User Guide.pdf" document.

Figure 1. PSoC Designer Layout



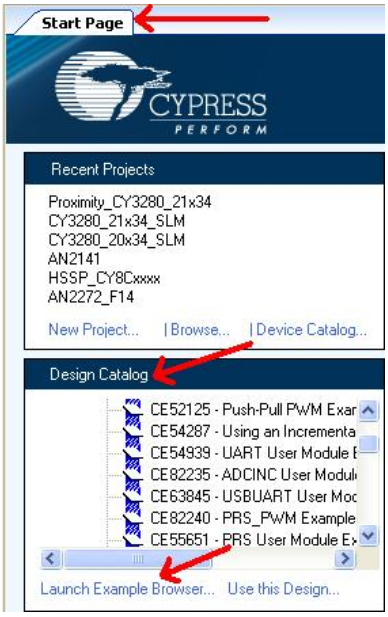
2.2 Code Examples

The following webpage lists the PSoC Designer based code examples. These code examples can speed up your design process by starting you off with a complete design, instead of a blank page, and also show how PSoC Designer User Modules can be used for various applications.

http://www.cypress.com/documentation/code-examples/psoc-1-code-examples

To access the code examples integrated with PSoC Designer, follow the path **Start Page > Design Catalog > Launch Example** *Browser* as shown in Figure 2.

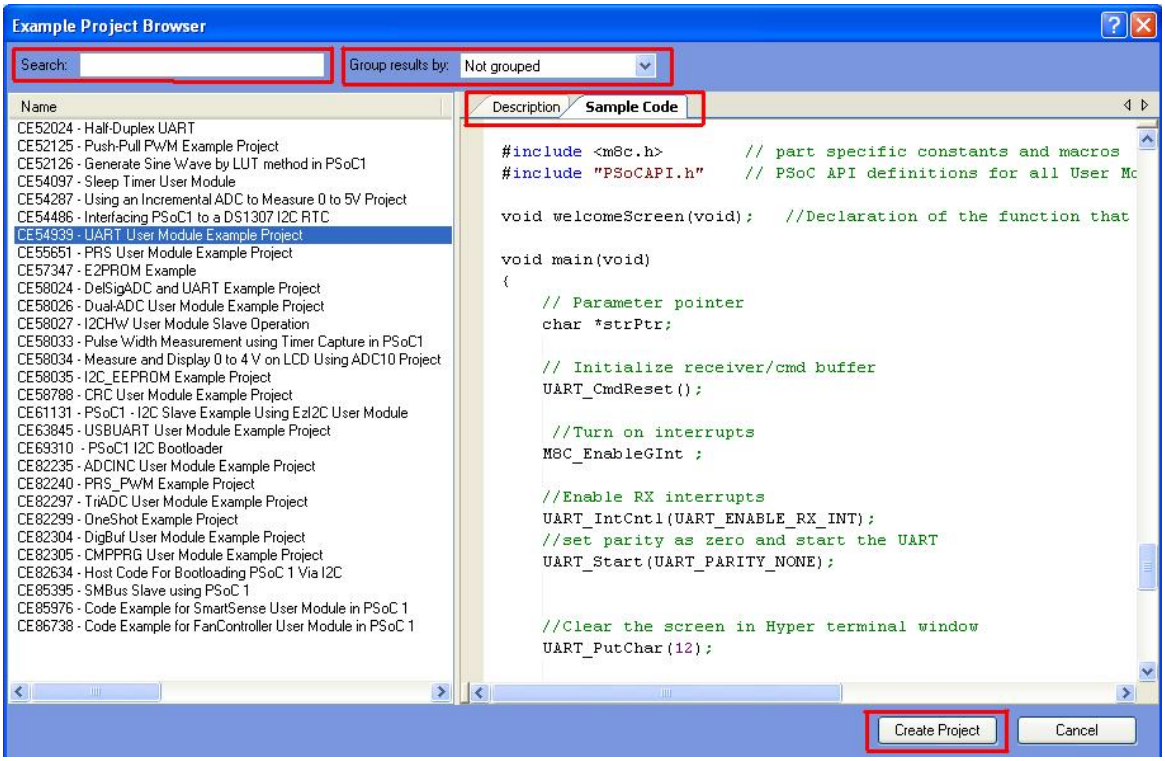
Figure 2. Code Examples in PSoC Designer



In the Example Projects Browser shown in Figure 3, you have the following options:

- Keyword search to filter the projects.
- Listing the projects based on category.
- Review the datasheet for the selection (on the Description tab).
- Review the code example for the selection. You can copy and paste code from this window to your project, which can help speed up code development, or
- Create a new project (and a new workspace if needed) based on the selection. This can speed up your design process by starting you off with a complete, basic design. You can then adapt that design to your application.

Figure 3. Code Example Projects with Sample Codes



2.3 PSoC Designer Help

Visit the PSoC Designer home page to download the latest version of PSoC Designer. Then, launch PSoC Designer and navigate to the following items:

- **IDE User Guide**: Choose **Help > Documentation > Designer Specific Documents > IDE User Guide.pdf**. This guide gives you the basics for developing PSoC Creator projects.
- **Simple User Module Code Examples**: Choose **Start Page > Design Catalog > Launch Example Browser**. These code examples demonstrate how to configure and use PSoC Designer User modules.
- **Technical Reference Manual**: Choose **Help > Documentation > Technical Reference Manuals**. This guide lists and describes the system functions of PSoC 1 devices.
- **User Module Datasheets**: Right-click a User module and select “Datasheet.” This datasheet explains the parameters and APIs of the selected user module.
- **Device Datasheet**: Choose **Help > Documentation > Device Datasheets** to pick the datasheet of a particular PSoC 1 device.
- **Imagecraft Compiler Guide**: Choose **Help > Documentation > Compiler and Programming Documents > C Language Compiler User Guide.pdf**. This guide provides the details about the Imagecraft compiler specific directives and Functions.

2.4 Technical Support

If you have any questions, our technical support team is happy to assist you. You can create a support request on the Cypress Technical Support page.

If you are in the United States, you can talk to our technical support team by calling our toll-free number: +1-800-541-4736. Select option 8 at the prompt.

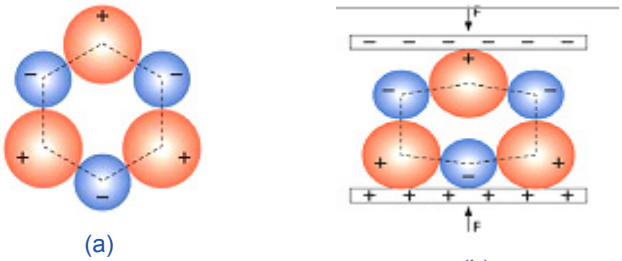
You can also use the following support resources if you need quick assistance.

- Self-help
- Local Sales Office Locations

3 Theory of Operation

Piezo crystal structures are shown in Figure 4.

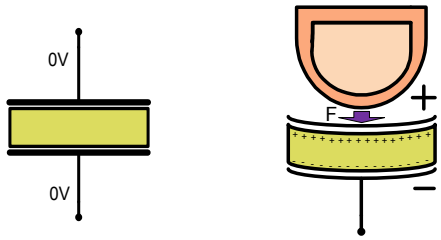
Figure 4. Unpressed and Pressed Crystals



(a) (b)

Figure 4 (a) shows a six-sided crystal structure cell when button is not pressed. Figure 4 (b) shows the charge displacement in the cell when the button is pressed. The pressure on this structural cell causes a shift in the gravity centers of the positive and negative charges against each other. The resulting difference in the charge is evaluated as a voltage or charge on the sensor sides.

Figure 5. Unpressed Crystal and Finger Influence

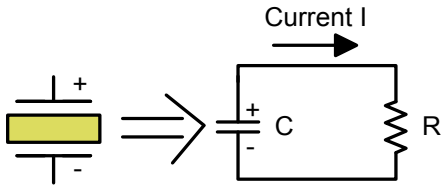


(a) (b)

The charge that is generated is applied across the piezo sensor capacitance. If the force applied to the sensor is constant, then the sensor generates a fixed amount of the charge. The result is voltage which can be measured by any voltage measurement device.

The voltage measurement device has input resistance that causes slow sensor discharge as shown in Figure 6. This charge is lost after some time due to sensor leakage current and input resistance of measurement device.

Figure 6. Sensor Circuit with Load Resistance R



The piezo sensor only responds to change in force. Once the crystal is pressed and the pressure remains constant, no more charge will be produced. This is the reason behind why the self-discharge results in reduction of voltage across the sensor. When designing a piezo keypad, ensure the following:

- High sensitivity to very small pressure forces
- Long button activation times
- High temperature stability

High sensitivity can be achieved using a low noise amplifier or analog-to-digital converter (ADC) with 8 to 10 bit resolution.

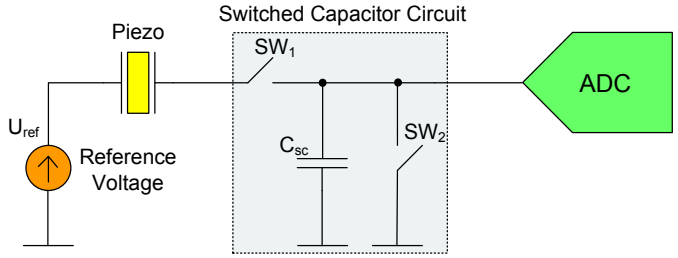
The long activation time is required to generate more charge which is needed for functions such as a clock setting button or a menu slider. The maximum activation time depends on the measurement technique and the piezo keypad's self discharging current. Different measurement principles achieve different times. Most methods provide very small activation times of less than 10 seconds. However, customers often ask for more than two minutes.

Many piezo sensors provide high voltages even from a soft press. This is not a problem for a discrete component solution. When you use a microcontroller without external components however, the input voltage cannot be higher than the power supply voltage. Most low cost, general-purpose microcontrollers use a 1.8 V, 3.3 V, or 5 V power supply. The obvious solution is a resistive divider, but high value resistors of 100 G Ohms or more are needed. These resistors increase input leakage current and decrease the activation time. A solution to this problem is discussed in the section High Voltage Sensor Scan.

Temperature stability is an important parameter for industrial applications. The sensor mechanical deformations or temperature gradients can cause the sensor voltage to shift in a positive or negative direction. The piezo crystal voltage is highly dependent on the temperature. The sensor leakage current also increases with temperature. The solution described in this document has automatic temperature drift compensation and measure both the positive and negative parts of the piezo charge.

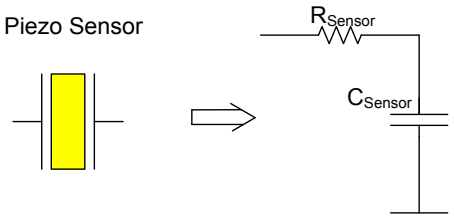
The solution is to use a switched capacitor circuit combined with an ADC for piezo sensor signal measurement, as shown in Figure 7. To understand the operation of Switched Capacitor Circuit, see AN2041 - Understanding PSoC® 1 Switched Capacitor Analog Blocks.

Figure 7. Scanning Principle



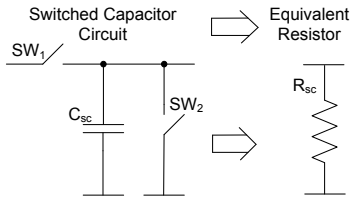
The sensor can be represented in equivalent RC circuit using sensor resistance R_{Sensor} and capacitor C_{Sensor}, as shown in Figure 8. The Switch Capacitor Circuit can be represented as shown in Figure 9.

Figure 8. Piezo Sensor Equivalent Circuit



The sensor is connected to reference voltage on one side and switched capacitor circuit on the other. The positive reference source is needed because the sensor can generate negative voltage after finger release. Both positive and negative signal parts should be correctly measured. There are no specific requirements for this reference source, it can be simple resistive divider or two microcontroller pins configured to pull-up and pull-down modes and split together.

Figure 9. Switched Capacitor Circuit Equivalent Resistor

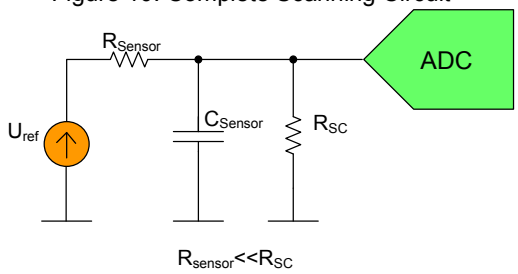


The sensor signal measurement flow is as follows:

- A switched capacitor circuit operates cycles to charge capacitor C_{Sensor} to reference voltage as shown in Figure 10.
- When switches SW1 and SW2 are opened, then capacitor C_{SC} holds the piezo sensor's voltage.
- Finally, the ADC processes the data.

When the data is complete, the firmware connects the next sensor to the switched capacitor circuit input and measures it. The sensor pins that are not scanned are in high impedance state. This allows you to detect long lasting activation on the button without any compensation circuits and external components.

Figure 10. Complete Scanning Circuit



The advantages of this design are:

- No high value resistors. The design uses a switched capacitor circuit and provides the same functionality as high value resistors. The switched capacitor circuit is less expensive and easy to integrate in the chip.
- Achieves more than two minutes of activation time. The switched capacitor circuit does not have a continuous discharge current when compared to a resistor and provides a long activation time limited by the sensor's internal leakage current.
- Decreases the influence of the ADC leakage current on the piezo sensor. This is because the ADC input is disconnected from the sensor while data is being processed.
- Sets the ADC input voltage to a reference value if no force is applied. As switched capacitor circuit is used, it increases the sensitivity and does not waste charge to connect traces.
- Compensates for temperature drift. Temperature changes and gradients can cause some sensor parameters to change. However, switched capacitor circuits always provide a stable reference voltage on the ADC input without depending on other sensor parameters.

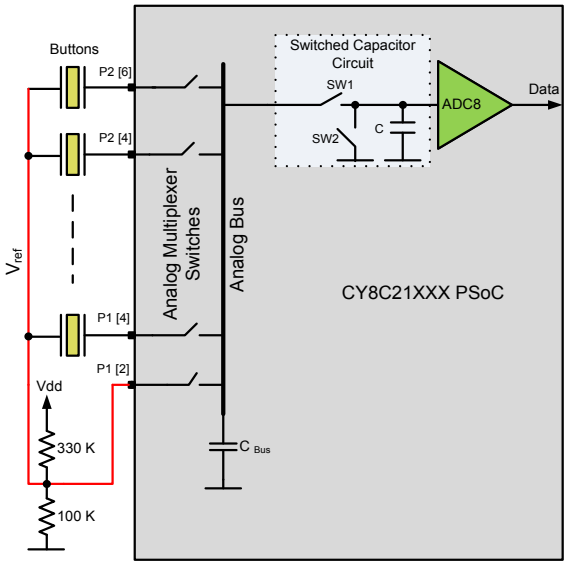
- Compensates for the sensor's parasitic capacitance. These capacitances are always charged to a reference voltage.
- Can be used on general purpose microcontrollers without external components. There are three blocks needed:
  - A 10-bit or more ADC
  - A switched capacitor circuit
  - A sensor multiplexer

  All modern microcontrollers have an ADC. Many of them have an internal multiplexer. The switched capacitor circuit can be built using the multiplexer and a GPIO. A PSoC device has all of them in a single chip.
- No RF radiation. Unlike a charge balancing principle there are no short pulses on the sensor that cause radiation.

This solution can be used in CMOS technology; it is already implemented in existing Cypress PSoC microcontrollers.

A multiple sensor scanning scheme is shown in Figure 11. One microcontroller pin is required for every sensor. Each sensor is connected to a pin on one side and to a common reference point on the other. The same reference is routed to the microcontroller for analog multiplexer (AMux) bus capacitance pre-charge and to prevent charge loss. AMux bus is used to connect none, one, or more analog signals to a different common analog signal. The practical need of the pre-charge is to decrease the current leakage from the source by slowing down the discharge rate *dV*/*dT* of the input supply voltage. There are no specific requirements for this reference source – a resistive divider or two microcontroller pins configured in pull-up and pull-down modes can be used.

Figure 11. Multiple Sensors Scanning Principle

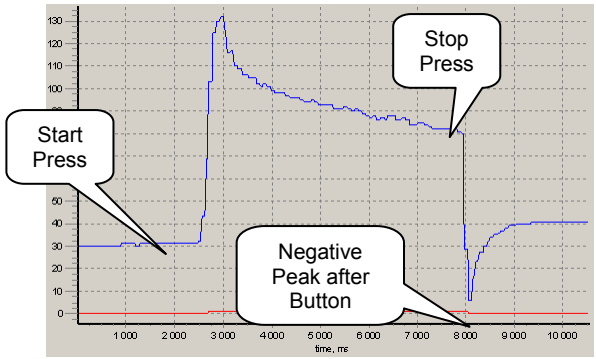


Only one ADC and one switched capacitor circuit is needed to scan all sensors. Connect the C_{BUS} (chip internal AMux bus capacitance) to a reference source before the sensor-to-bus connection to prevent sensor charge loss.

Only one sensor should be connected to the analog bus at a time. The switched capacitor circuit is connected to the same bus and provides correct measurements. After scanning, the sensor disconnects from the analog bus and is in high impedance state to save charge.

The button press graph is shown in Figure 12.

Figure 12. ADC Data Waveform



To know what button is pressed, filter the ADC signal using a low pass filter (LPF) and compare the value not filtered with the filtered value. In this case, a fast finger press force causes a high difference between the two values. The slow signal changes caused by the environment are not detected and pass through the LPF. Traditionally, the LPF value is called the baseline. Baseline update algorithms are widely used in many systems including CapSense.

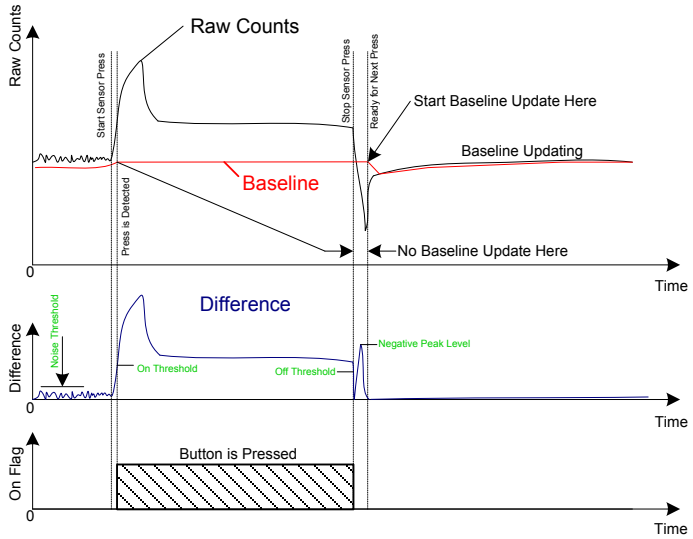
A piezo sensor signal with a high negative peak needs a special asymmetric baseline update algorithm. The algorithm for the piezo keypad application is shown in Figure 14. The main advantages are:

- Tracks positive and negative baseline parts at different speeds.
- Tracks positive and negative baseline parts with different threshold settings.
- Skips baseline updates while a strong negative peak is present.

The algorithm works as follows: At controller start, the baseline value is automatically set to the same level as the sensor signal. The baseline updates run only after all sensors are scanned. Then, the current sensor value is compared to the previous baseline (LPF) value. If the raw counts are higher than the baseline the algorithm performs a positive baseline update. Otherwise, it performs a negative baseline update.

The baseline update illustration is shown in Figure 13.

Figure 13. Baseline Update Illustration



3.1 Positive Baseline Update

If the difference between the current raw count value and the baseline is higher than the “On threshold” value, then the “On” mask is set. The algorithm proceeds to the next sensor baseline update. If the difference is less than the “Off threshold” value then the “On” mask is cleared. Some hysteresis is realized by setting different values for the “on” and “off” thresholds.

If the difference between the current raw count value and the baseline value is higher than the “noise threshold” value and lowers than the “On threshold" value, then baseline update is enabled. This means that the baseline is updated if a button press is not detected, and not updated if a button press is detected. This allows you to detect long lasting button presses. If a long press is not needed and temperature stability is more important, the baseline can be updated without checking the “On” threshold.

Baseline updates are made using the “bucket” method. The bucket is a one byte variable in memory that increases with each scan. If the bucket is higher than the “baseline update threshold” then some value is added to the current baseline value and the bucket is cleared. This value defines the minimum baseline increase.

3.2 Negative Baseline Update

The baseline is updated if the difference between the baseline and raw counts is higher than the “noise threshold” and less than the “negative peak threshold”. This threshold should be adjusted to prevent the negative baseline from falling beyond the signal. This fall can cause a false “On” condition when the signal starts rising faster than the negative baseline update speed.

Negative baseline updates are also made using the “bucket” method. If the bucket is higher than the “baseline update threshold” then some value is subtracted from the current baseline value and the bucket is cleared.

Figure 14. Asymmetrical Baseline Update Algorithm



4 High Voltage Sensor Scan

Some piezo sensors can generate up to 30 V of output voltage. Typical microcontroller power supply voltages are 1.8 V, 3.3 V, or 5 V. Sensor voltages applied to the microcontroller pin opens an internal body diode that causes fast sensor discharge, see Figure 15.

The obvious solution is to use a resistive divider and move the voltage down to power supply. But this is not a good idea. The Resistive divider continuously discharges the sensor and dramatically decreases the length of the activation time. Also, two high value resistors are needed for each sensor.

Another option is to use capacitors in parallel with the sensor, as shown in Figure 16. Sensor charge is distributed across the sensor capacitances and an external capacitance that decreases resulting in a sensor output voltage. The advantages of this method are:

- No continuous discharge current
- Increases the long activation time
- Only one capacitor is needed for each sensor
- Decreases sensor noise
- Increases EMC immunity

Figure 15. High Voltage Sensor Discharge Path

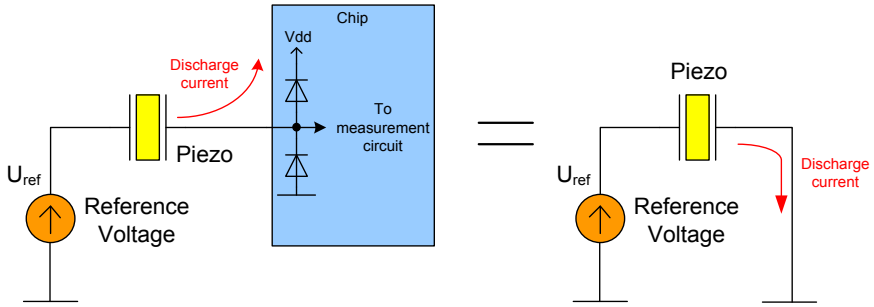
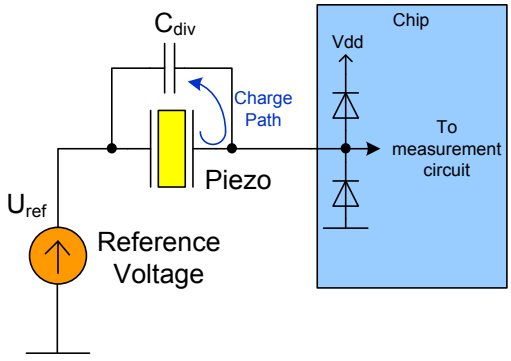


Figure 16. Capacitor to Decrease Sensor Voltage



A capacitor can be connected in parallel with the sensor or from the microcontroller pin to ground depending on the measurement circuit. The sensor charge is distributed across C_{DIV} and the sensor capacitance (C_p). The resulting output voltage is reduced on C_{DIV} to C_p and the resulting sensor voltage does not exceed V_{DD} level. The advantage of this is that the external capacitor does not waste charge, but holds it. All of the charge resulting from pressure on the piezo sensor is accumulated on C_{DIV} and C_p and used to measure the circuit. The resulting activation time and noise parameters are greatly improved.

The example schematic of the piezo keypad using PSoC is shown in the Board Schematic. This schematic is similar for single buttons and matrix keypad scanning. It includes the following components:

- PSoC – U1
- ISSP/data connector – J2
- Piezo keypad connector – J1
- Reference resistors – R8 and R9
- Buzzer – LS1
- Capacitor – C2

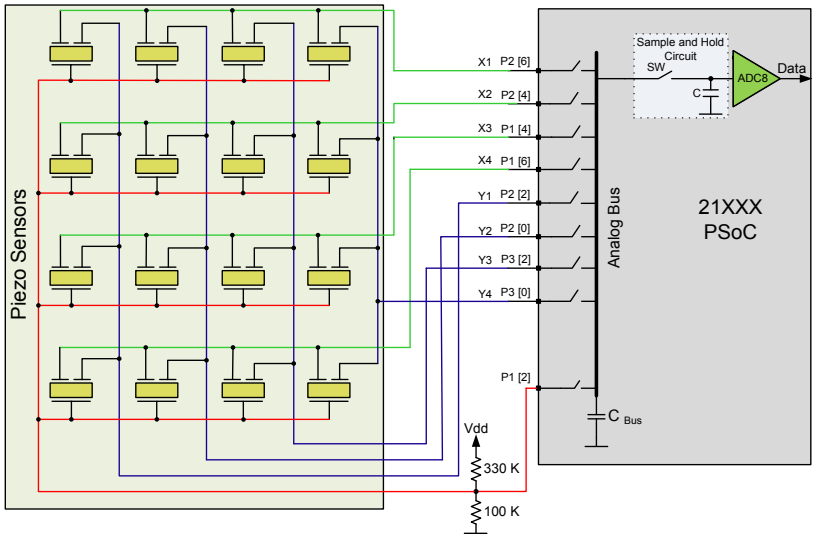
The LS1 buzzer is used to generate sound when the button is pressed. The PSoC generates sound on port P0[3] using a pulse width modulator (PWM). Note that generating the sound signal through the buzzer causes the power supply voltage to fluctuate and noise on the ADC8 to increase. The C2 capacitor limits the buzzer current and decreases the power supply fluctuation due to generating the sound.

Connector J2 is used as for in system serial programming (ISSP) and for I^2C data communication. Connector J1 is used to connect the piezo keypad (both matrix and single buttons) to the board. The resistors R8 and R9 set the reference voltage V_{ref} to about 1.5 V.

5 Matrix Keypad Scan

The matrix keypad implementation is shown in Figure 17. The board connected to the matrix keypad is similar to a traditional keypad. As shown in the figure, each sensor has three pins, one of which is connected to the reference voltage V_{ref}, the other two to the X and Y button coordinates. Therefore when a button is pressed the voltage on the two conductors increases simultaneously (X and Y respectively).

Figure 17. Matrix Keypad Hardware Implementation



6 Summary

The technique described in this document can be used to reliably scan both single and matrix piezo buttons. It allows a continuous button press for two minutes. The button press period can be modified as follows:

- Change the reference voltage. Smaller reference allows more effective use of ADC range and therefore provides higher sensitivity.
- Change ADC resolution. Higher resolution provides higher sensitivity.
- Change piezo button size. Larger size provides higher signal.
- Tune firmware parameters to increase the sensitivity by 10 to 20 percent.

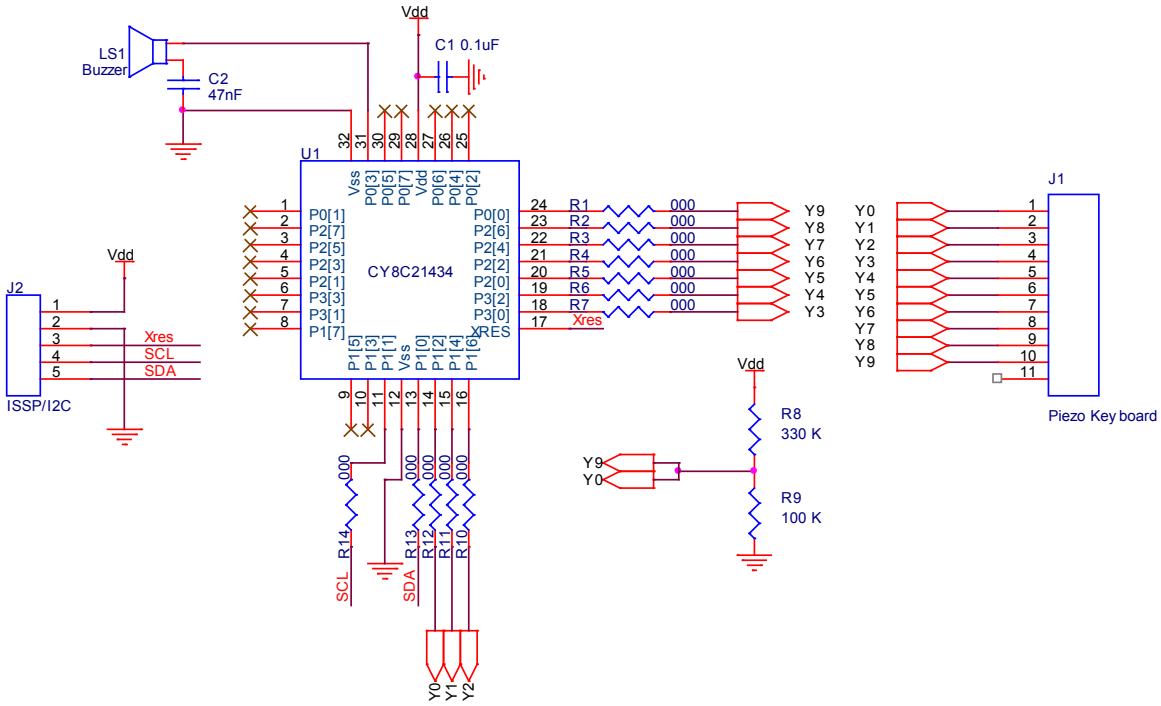
Higher sensitivity also provides longer maximum activation time. This design was tested in the temperature range from -20 °C to +55 °C. However, above +35 °C, the maximum activation time starts decreasing and at +55 °C it detects only short presses. This is due to the increase of the piezo element self discharge current.

About the Author

Name:	Andriy Ryshtun
Title:	Systems Engr Sr Staff
Background:	Andriy graduated in Radio Electronics from National University, Lviv Polytechnica (Ukraine) in 2006. He works in the Ukraine Solution Center. His interests include software and hardware development, and algorithm investigation.

A Appendix A

A.1 Board Schematic



A.2 Example Designs

Figure 18. Single Buttons Scan

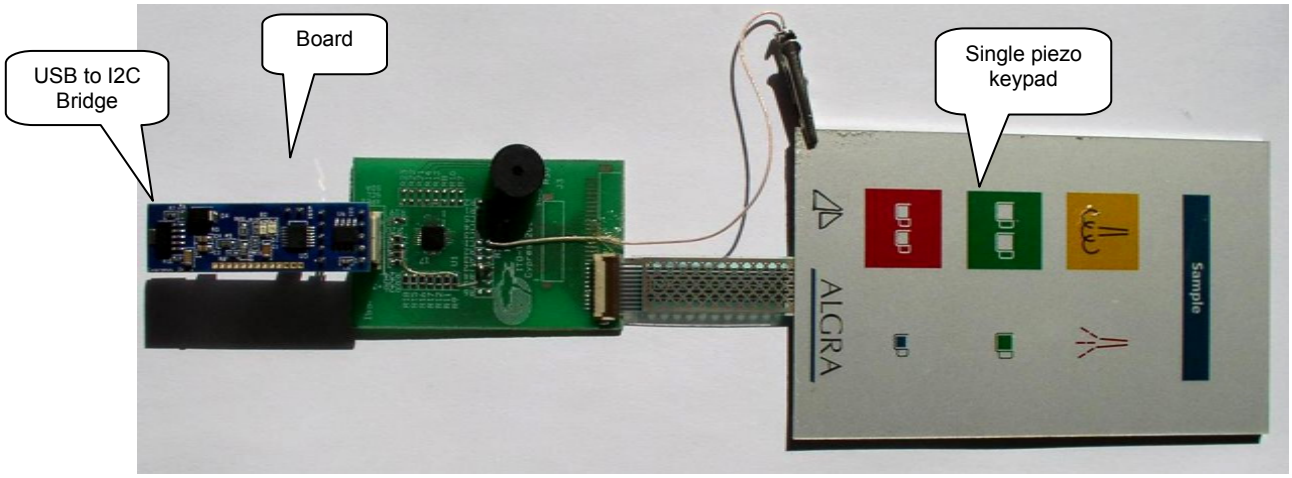
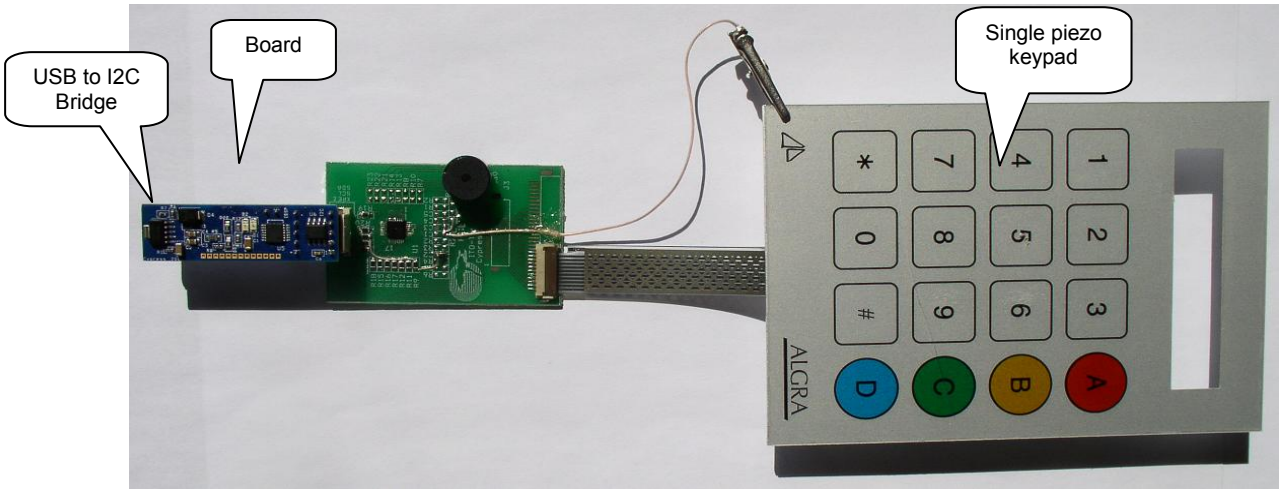


Figure 19. Matrix Keypad Scan



Document History

Document Title: AN60639 - PSoC® 1 Piezo Keypad Implementation

Document Number: 001-60639

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2929976	ANDRIY	05/10/2010	New application note.
*A	3265813	RYAN_UKR	05/25/2011	Document title changed. Abstract updated.
*B	4397603	RICA	06/03/2014	Updated in new template. Completing Sunset Review.
*C	4635426	DIMA	01/22/2015	Updated the Software Version to “PSoC Designer™ 5.4 CP1” in page 1. Updated content for better readability in Introduction. Replaced references to Piezo keyboard with Piezo Keypad in the entire document.
*D	4905862	ASRI	09/10/2015	Updated Abstract in page 1. Updated Introduction, Theory of Operation, High Voltage Sensor Scan and Summary. Updated Figure 8 through Figure 10.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support

PSoC is a registered trademark and PSoC Creator is a trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2010-2015. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.
This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.
Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.
Use may be limited by and subject to the applicable Cypress software license agreement.