AN2186

Consumer or Industrial: Acoustic Glass Break Detector

Author: Vadym Grygorenko

Associated Project: Yes
Associated Part Family: CY8C22xxx, CY8C24xxx, CY8C27xxx
Software Version: PSoC® Designer™ 5.4
Related Application Notes: AN2099

AN2186 describes a low-cost, three-band acoustic glass break detector. Digital signal processing provides high sensitivity and sufficient resistance to false alarms.

1 Introduction

There are a variety of ways to implement glass break recognition. One method is to detect the vibration made by breaking glass. This technology may be realized using a piezo-film sensor for impact detection. Another method is to detect the sound of breaking glass. The simplest of such detectors are responsive to sound level spikes in the 4-5 kHz frequency band. The main disadvantage of the acoustic method is the high probability of false alarms caused by loud music, tinkling of bells, or other sounds. To increase false alarm immunity, two or more frequency bands are treated and information relating to amplitude and timing of the separated channels is analyzed.

It is possible to make dual technology detectors in which one part detects a very low frequency sound pressure wave created by the flexing of glass before breakage and another part is sensitive to high frequencies of the shattered glass. The most advanced detectors are based on neuronets and pattern recognition. However, these detectors require powerful digital signal processors to implement complex algorithms.

The PSoC® solution for a framed-glass break detector presented in this Application Note is based on timing and amplitude analysis of sound at three frequencies: 35 Hz, 300 Hz, and 5000 Hz. All signal treatments are performed in the software. This allows tuning flexibility and adaptation to different glass types.

2 PSoC Resources

Cypress provides a wealth of data at www.cypress.com to help you to select the right PSoC device for your design, and quickly and effectively integrate the device into your design. In this document, PSoC refers to the PSoC 1 family of devices. To learn more about PSoC 1, refer to the application note AN75320 - *Getting Started with PSoC 1*.

The following is an abbreviated list for PSoC 1:

- **Overview:** PSoC Portfolio, PSoC Roadmap
- **Product Selectors:** PSoC 1, PSoC 3, PSoC 4, or PSoC 5LP. In addition, PSoC Designer includes a device selection tool.
- **Datasheets:** Describe and provide electrical specifications for the PSoC 1 device family.
- **Application Notes and Code Examples:** Cover a broad range of topics, from basic to advanced level. Many of the application notes include code examples.
- **Technical Reference Manuals (TRM):** Provide detailed descriptions of the internal architecture of the PSoC 1 devices.
- **Development Kits:**
    - CY3215A-DK In-Circuit Emulation Lite Development Kit includes an in-circuit emulator (ICE). While the ICE-Cube is primarily used to debug PSoC 1 devices, it can also program PSoC 1 devices using ISSP.
    - CY3210-PSOCEVAL1 Kit enables you to evaluate and experiment Cypress's PSoC 1 programmable system-on-chip design methodology and architecture.
    - CY8CKIT-001 is a common development platform for all PSoC family devices.
- The MiniProg1 and MiniProg3 devices provide an interface for flash programming.

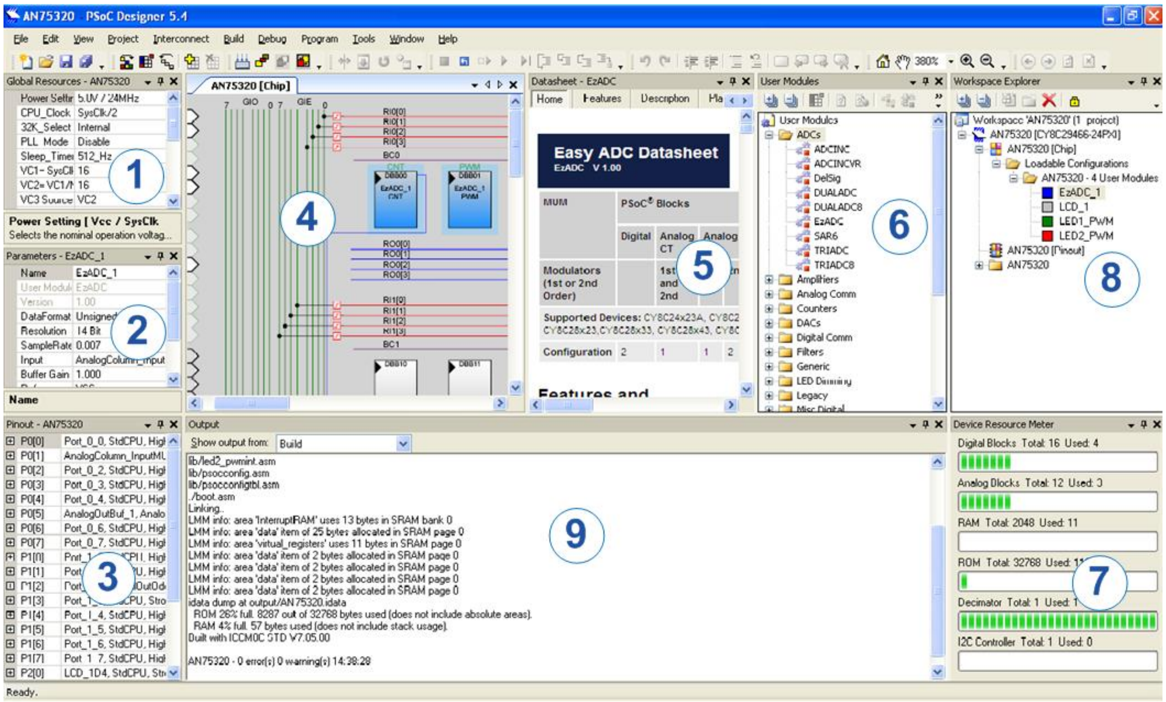
2.1 PSoC Designer

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, go to **PSoC® Designer > Help > Documentation > Designer Specific Documents > IDE User Guide**.

Figure 1. PSoC Designer Layout



Power Settir 5.0V / 24MHz
CPU_Clock SysClk/2
32K_Select Internal
PLL Mode Disable
Sleep_Timer 512_Hz
VC1= SysClk 16
VC2= VC1/N 16
VC3 Source VC2

Power Setting [Vcc / SysClk
Selects the nominal operation voltag...

Name EzADC_1
User Module EzADC
Version 1.00
DataFormat Unsigned
Resolution 14 Bit
SampleRate 0.007
Input AnalogColumn_Input
Buffer Gain 1.000

Show output from: Build

```
lib/led2_pwmint.asm
lib/psocconfig.asm
lib/psocconfigtbl.asm
./boot.asm
Linking..
LMM info: area 'InterruptRAM' uses 13 bytes in SRAM bank 0
LMM info: area 'data' item of 25 bytes allocated in SRAM page 0
LMM info: area 'virtual_registers' uses 11 bytes in SRAM page 0
LMM info: area 'data' item of 2 bytes allocated in SRAM page 0
LMM info: area 'data' item of 2 bytes allocated in SRAM page 0
LMM info: area 'data' item of 2 bytes allocated in SRAM page 0
LMM info: area 'data' item of 2 bytes allocated in SRAM page 0
idata dump at output/AN75320.idata
  ROM 26% full. 8287 out of 32768 bytes used (does not include absolute areas).
  RAM 4% full. 57 bytes used (does not include stack usage).
Built with ICCM8C STD V7.05.00

AN75320 - 0 error(s) 0 warning(s) 14:38:28
```

Digital Blocks Total: 16 Used: 4
Analog Blocks Total: 12 Used: 3
RAM Total: 2048 Used: 11
ROM Total: 32768 Used: 11
Decimator Total: 1 Used: 1
I2C Controller Total: 1 Used: 0

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/go/PSoC1Code 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

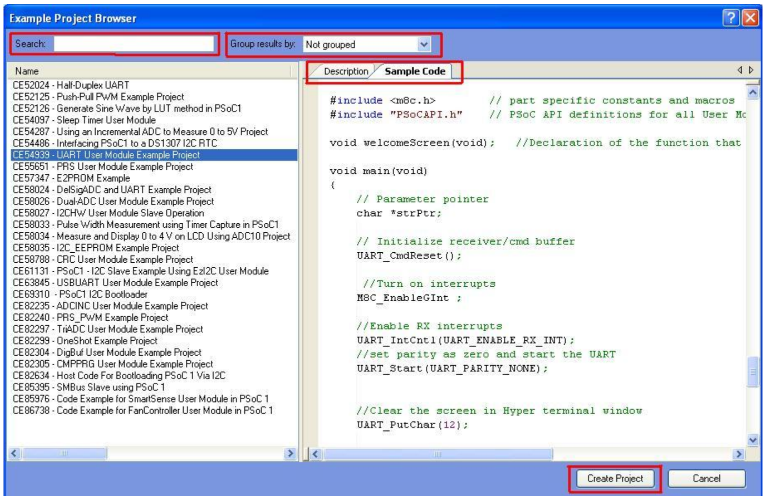
Proximity_CY3280_21x34
CY3280_21x34_SLM
CY3280_20x34_SLM
AN2141
HSSP_CY8Cxxxx
AN2272_F14

CE52125 - Push-Pull PWM Exar
CE54287 - Using an Incrementa
CE54939 - UART User Module E
CE82235 - ADCINC User Modul
CE63845 - USBUART User Moc
CE82240 - PRS_PWM Example
CE55651 - PRS User Module E›

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



CE52024 - Half-Duplex UART
CE52125 - Push-Pull PWM Example Project
CE52126 - Generate Sine Wave by LUT method in PSoC1
CE54097 - Sleep Timer User Module
CE54287 - Using an Incremental ADC to Measure 0 to 5V Project
CE54486 - Interfacing PSoC1 to a DS1307 I2C RTC
CE54939 - UART User Module Example Project
CE55651 - PRS User Module Example Project
CE57347 - E2PROM Example
CE58024 - DelSigADC and UART Example Project
CE58026 - Dual-ADC User Module Example Project
CE58027 - I2CHW User Module Slave Operation
CE58033 - Pulse Width Measurement using Timer Capture in PSoC1
CE58034 - Measure and Display 0 to 4 V on LCD Using ADC10 Project
CE58035 - I2C_EEPROM Example Project
CE58788 - CRC User Module Example Project
CE61131 - PSoC1 - I2C Slave Example Using EzI2C User Module
CE63845 - USBUART User Module Example Project
CE69310 - PSoC1 I2C Bootloader
CE82235 - ADCINC User Module Example Project
CE82240 - PRS_PWM Example Project
CE82297 - TriADC User Module Example Project
CE82299 - OneShot Example Project
CE82304 - DigBuf User Module Example Project
CE82305 - CMPPRG User Module Example Project
CE82634 - Host Code For Bootloading PSoC 1 Via I2C
CE85395 - SMBus Slave using PSoC 1
CE85976 - Code Example for SmartSense User Module in PSoC 1
CE86738 - Code Example for FanController User Module in PSoC 1

```
#include <m8c.h>        // part specific constants and macros
#include "PSoCAPI.h"    // PSoC API definitions for all User Mc

void welcomeScreen(void);   //Declaration of the function that

void main(void)
{
    // Parameter pointer
    char *strPtr;

    // Initialize receiver/cmd buffer
    UART_CmdReset();

     //Turn on interrupts
    M8C_EnableGInt ;

    //Enable RX interrupts
    UART_IntCntl(UART_ENABLE_RX_INT);
    //set parity as zero and start the UART
    UART_Start(UART_PARITY_NONE);


    //Clear the screen in Hyper terminal window
    UART_PutChar(12);
```

2.3 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.

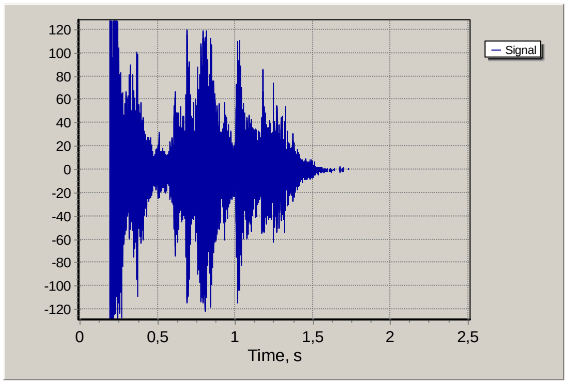
You can also use the following support resources if you need quick assistance.

- Self-help
- Local Sales Office Locations

3 Detecting Algorithm

A typical waveform for the sound of breaking glass is shown in Figure 4.

Figure 4. Waveform for Breaking Glass

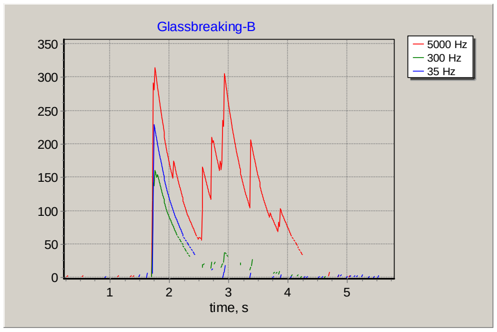


The initial spike created by the shock waves that flow outward after impact is easily seen.

To recognize the sound of breaking glass and to distinguish it from another source, three band pass filters (BPFs) are applied and their timing and output amplitudes are analyzed.

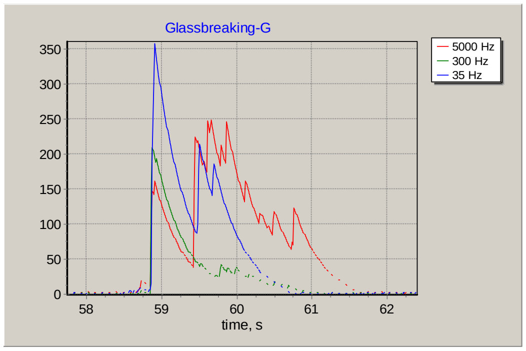
The first filter is a low-pass filter (LPF) with a 35 Hz cutoff frequency. Its role is to detect the initial shock wave. The second filter is a band-pass filter with a 5 kHz central frequency and a Q-factor of 5. This filter responds to the sound of shattering glass. The third filter is a 300 Hz BPF and serves to eliminate false alarms. Figure 5 shows the filters' outputs when a test signal is applied.

Figure 5. Filter Output (Glassbreaking-B.wav)



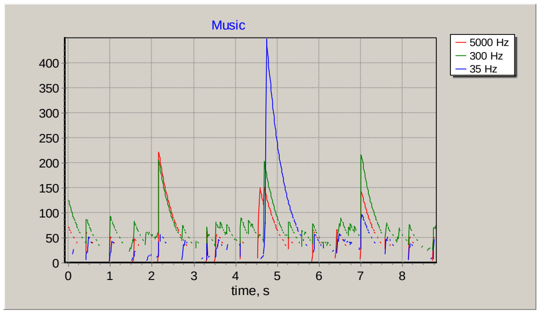
Another test signal response is shown in Figure 6.

Figure 6. Filter Output (Glassbreaking-G.wav)



For comparison, output caused by a rock music signal is shown in Figure 7.

Figure 7. Filter Output (Music)

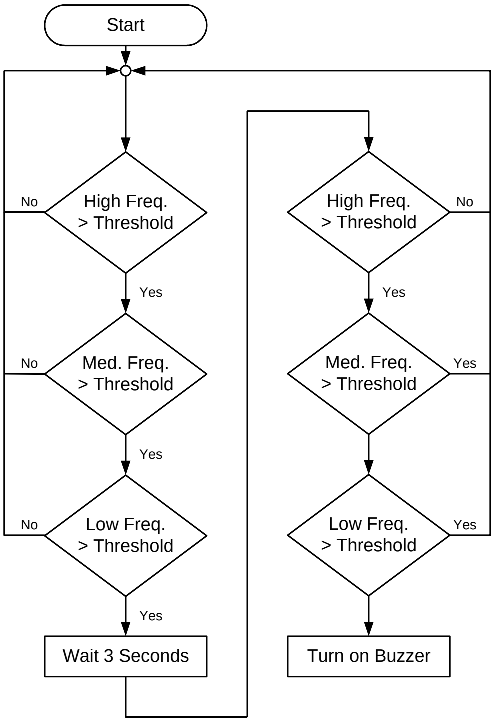


A comparative analysis of the presented waveforms allows one to make the following conclusions:

- Peculiarities in the sound of breaking glass are present in the amplitude spike for all filters' outputs and,
- Music differs from the sound of breaking glass in the amplitude of the middle-frequency filter output.

Therefore, the algorithm shown in Figure 8 may be employed to detect the sound of breaking glass.

Figure 8. Sound Break Detection Algorithm



4 PSoC® Implementation

A high-level device block diagram is shown in Figure 9.

The acoustic signal is received by the microphone and amplified by the programmable gain amplifier (PGA). The signal is digitized using an 8-bit delta-sigma analog-to-digital converter (ADC). All subsequent signal processing occurs in the software.

Such uncomplicated hardware architecture allows implementation of the simplest and least expensive 8-pin PSoC, the CY8C22xxx series. Using the more complex CY8C24xxx or CY8C27xxx PSoC family devices yields additional features such as wide range sensitivity adjustment and communication with a computer for tuning purposes.

Figure 9. High-Level Device Block Diagram

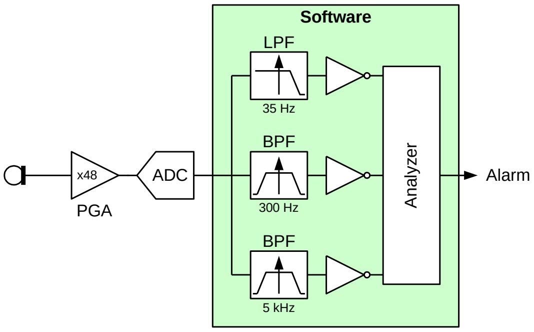
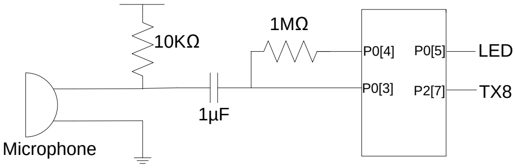


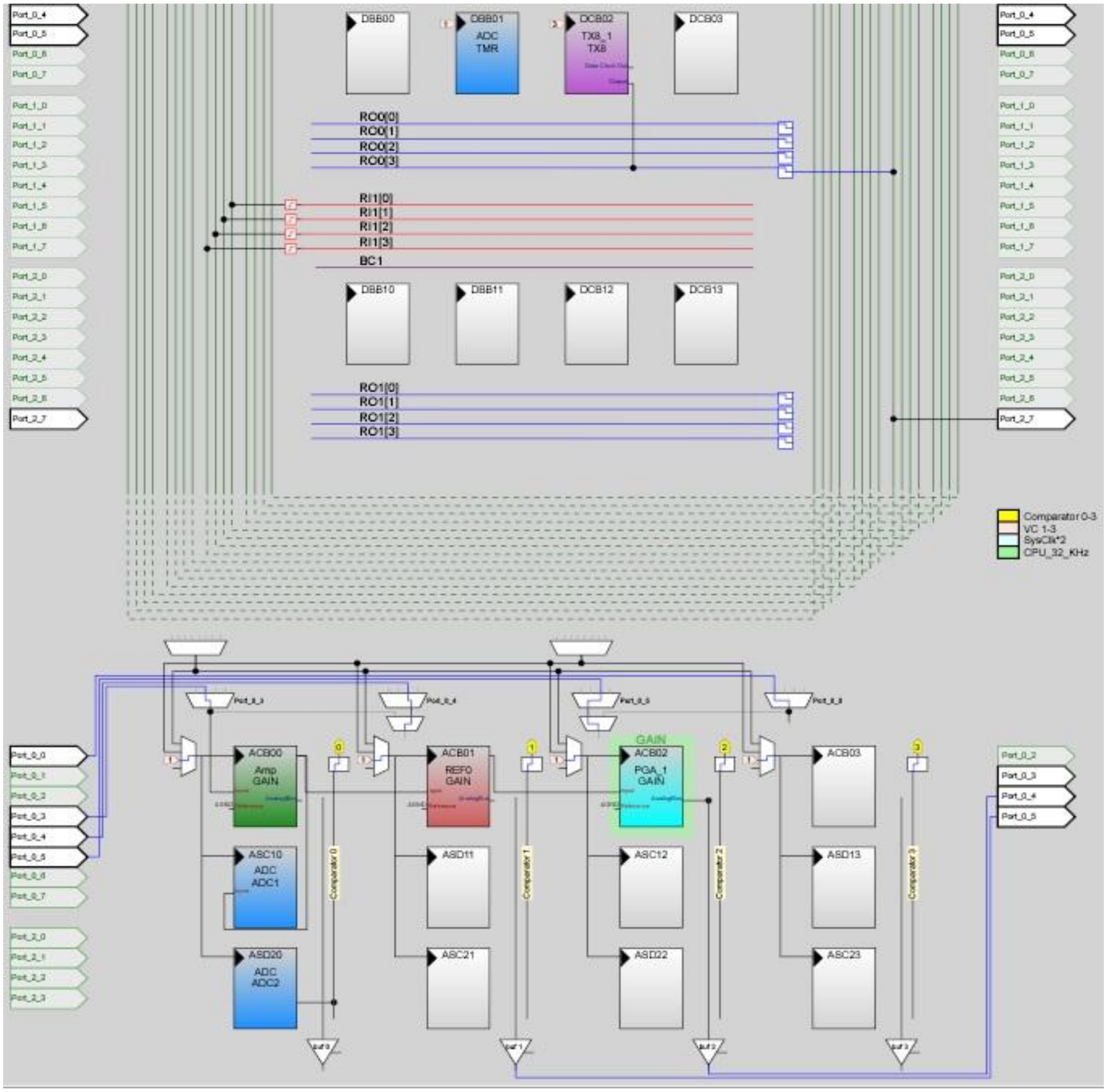
Figure 10 shows the PSoC CY8C27443 interface schematic. Pin P0[3] is the analog input. Pin P0[4] is the PGA buffer output. The TestMux routes the analog ground signal on P0[5]. P0[5] is the open drain alarm output. P0[5] powers the LED in the test circuit.

Figure 10. Interface Schematic for CY8C27443



To perform various tests with different sounds and visualize device output on the computer, the CY8C27xxx PSoC family may be used. Three user modules are added: two PGAs and a serial transmitter, TX8. One PGA is used to extend the gain range, the other to deliver the amplified signal to a test point external to pin P0[4]. The output of TX8 is connected to P2[7] and may be passed to a computer COM port through an RS232 line driver. The user module configuration is shown in Figure 11.

Figure 11. User Module Placement for CY8C27443



5 Software

The following tasks are completed in the software:

- Perform 3-channel digital filtration.
- Rectify filter outputs, emulating a peak detector.
- Compare calculated amplitudes with predefined and calculated real-time threshold levels.
- Perform timing analyses and decision making about alarm turn on.

The 5 kHz IIR band pass filter is defined by:

$$y_n = \frac{1}{4}(x_n - x_{n-2}) - \frac{3}{4}y_{n-2}, \qquad \text{Equation 1}$$

y is output and x is input.

The filter described in Equation (1) has a frequency response function with a maximum of 2 at one-fourth the sampling frequency and a Q-factor of approximately 5. In our case, the ADC sample rate is chosen to be 18750 Hz, $\left(\frac{24\,MHz}{5\cdot 256}\right)$, so the filter is tuned to a frequency or approximately 4.7 kHz.

It is important that coefficients in Equation (1) avoid multiplication and division operations. Addition and arithmetical shift operations are sufficient.

The 300 Hz BPF is also defined by Equation (1). However, it is necessary to reduce the input data sample rate. This is accomplished using another LPF and a decimator. A simple equation is used:

$$x_n = \frac{1}{16}\sum_{i=1}^{16} x_i. \qquad \text{Equation 2}$$

In other words, input data to the 300 Hz BPF is an average of the 16 ADC samples.

The LPF is defined by:

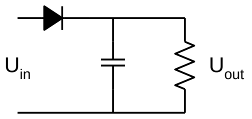
$$y_n = \frac{1}{2}(x_n + 2x_{n-1} + x_{n-1}) - \frac{1}{8}y_{n-2} \quad \text{Equation 3}$$

This filter has a cut-off frequency approximately 0.23 times the input sample rate. The gain is 1.78. Input data is measured as the average of the 128 ADC samples to achieve the 35 Hz band.

The averaged values are passed through a first order IIR high pass filter (HPF) with a cut-off frequency of approximately 0.36 Hz to generate offset canceling. This additional filter is based on Application Note AN2099 “Single-Pole IIR Filters. To Infinity And Beyond!”

Rectification is the next stage of signal treatment. A simple peak amplitude detector is simulated as shown in Figure 12.

Figure 12. Simple Peak Amplitude Detector



This circuit is emulated by the following:

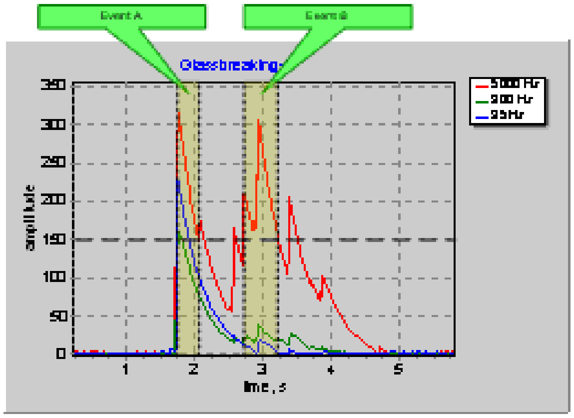
$$U_{out}[n] = \max\left(U_{out}[n-1]\cdot\left(1-\frac{1}{32}\right), U_{in}[n]\right). \quad \text{Equation 4}$$

Equation (4) is equivalent to the amplitude detector with a characteristic time of 0.5s.

Rectifier outputs are compared to threshold levels. These levels are defined by the analysis in Figures 2, 3 and 4. The threshold for the 5 kHz channel is set to 150 and the other channels’ threshold levels are calculated dynamically as a quarter of the current 5 kHz amplitude.

The timing analysis of filters’ outputs is shown in Figure 13.

Figure 13. Timing Analysis



There are two events that must be detected to ascertain glass breakage. The first event, marked as “Event A,” is that all channel output amplitudes exceed their threshold levels.

The second event, marked as “Event B,” occurs when high frequencies are above their threshold and middle frequencies are less than their corresponding threshold values. Event A is considered valid only if its duration is less than 1s. Event B must be longer than 0.25s in duration. Timeout for all events is set to 3s.

The CY8C27xxx implementation allows all channel output amplitudes to be transmitted through the TX8 User Module to a computer’s COM port. The source code for a Borland Delphi amplitude visualization application is included with the associated project. Note that this code uses freeware components. For more information, go to TurboPower Async Professional.

Broken glass test sounds are also included. It is possible to pass these sounds from a computer sound card’s linear output directly to C1 without a microphone, as shown in Figure 10.

6 Summary

This application note presents a simple acoustic glass break detector design. The design demonstrates the basic principles of glass breakage sound recognition. Development of the end-user devices may require additional adjustment of threshold levels and timing analysis parameters, depending on glass types, microphones used, or other parameters. Industrial glass break detector testers may be used for additional adjustments.

About the Author

Name:	Vadym Grygorenko.
Title:	Systems Engrg Sr MTS
Background:	Vadym earned a radiophysics diploma in 1986 from Ivan Franko National Lviv University and his Ph.D in 1992. His interests involve embedded systems design and application programming.

Document History

Document Title: AN2186 - Consumer or Industrial: Acoustic Glass Break Detector

Document Number: 001-33780

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	1499983	YARD_UKR	09/24/2007	OLD APP. NOTE: Obtained spec. # for note to be added to spec system.
*A	3201517	BIOL_UKR	03/21/2011	Updated BOOT.TPL file and UM versions in attached associated project.
*B	4365315	RICA	04/29/2014	Updated to new template. Completing Sunset Review.
*C	4664203	ASRI	03/23/2015	Updated Software Version as “PSoC® Designer™ 5.4” in page 1. Updated PSoC® Implementation: Updated description. Removed figure “User Module Placement for CY8C22113”. Updated Figure 10. Updated Figure 11. Updated attached associated project to PSoC Designer 5.4. Completing Sunset Review.
*D	4788165	ASRI	06/05/2015	Added PSoC Resources section Sunset review Updated template

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, 2007-2015. 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.