

AN2187

Authors: Chris and Vincent Paiano

Associated Project: Yes

Associated Part Family: CY8C21xxx, CY8C23x33, CY8C24xxx, CY8C27x43, CY8C29x66

[GET FREE SAMPLES HERE](#)

Software Version: PSoC Designer™ 5.0

Associated Application Notes: None

Application Note Abstract

This application note describes an audible clock which uses a PSoC®, a 16-bit counter, an external 32 kHz crystal, and an 8-bit pulse width modulator. An encoding scheme is also included.

Introduction

When recording audio, it is useful to know the time of recording by listening to an unassuming audible signal in the background. The PSoC keeps standard 12-hour am/pm time through a counter interrupt. A pulse-width modulator (PWM) is connected to a speaker (or an amplifier) to produce an audible representation of the current time of day. An external 32 kHz crystal is used to keep the PSoC's timing accurate enough for use as a clock.

A rhythmic encoding scheme enables the listener to determine the time of day within a few seconds. To our knowledge, there is no audible time encoding scheme. The encoding scheme described in this application note uses rhythm and simple tones. See section [Encoding Scheme](#) on page 2 and the illustration in the [Appendix](#) on page 4.

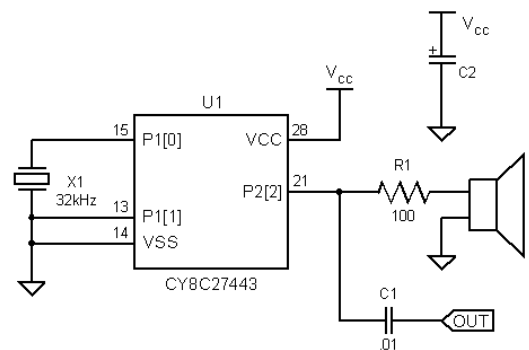
An audible clock is also useful as background noise. Instead of searching for a clock, just listen to the signal for a few seconds to determine the time of day.

Hardware Implementation

The PSoC implementation of the audible clock consists of a 16-bit counter to generate the one-second interrupt, clocked by an external 32 kHz crystal, and an 8-bit PWM connected to the speaker to generate the tones.

A schematic for the example project is shown in [Figure 1](#). The 32 kHz crystal connected to P1[0] and P1[1] can be sensitive to the PWM and incurs timing errors. Moving the PWM signals to P2[2] solves this problem. C₁ blocks the DC for coupling to an audio amplifier. R₁ limits the output current for direct speaker drive. C₂ is a decoupling capacitor for V_{DD}.

Figure 1 Example Project Schematic



Software and PSoC Implementation

All other operations are carried out in the software. Everything happens on the 16-bit counter's interrupt. One second elapses between interrupts. The software updates the current time, and then continues wherever it is on the time output. Up to three tones are produced each second to indicate the current hour, tens of minutes, and minutes. For example, 6:59 is represented by two sets of three tones for the 6, a pause, then one set of three and one set of two for the 5, a pause, then three sets of three for the 9, and finally a long pause before the next reading is produced. The time of day is measured in standard 12-hour am/pm format. Morning (am) is signaled by ascending tones, while afternoon and evening (pm) are signaled by descending tones.

Additionally, uncomment the sleep timer in the included project to put the processor to sleep at all times except when servicing the one-second interrupt for low power operation – making a battery-powered Audible Clock possible.

PSoC code is shown in Example Code 1 and demonstrates how all this is achieved in practice. Figure 2 shows a flowchart of the code. Figure 3 on page 3 shows the User Module configuration and Figure 4 on page 3 shows screen shots of the User Module parameter settings. Figure 5 shows screen shots of the global resource settings.

Encoding Scheme

The encoding scheme used in this application note is illustrated in the Appendix on page 4. Each second is split into triple notes. Up to three tones can be played rhythmically on each count corresponding to hours, minutes' ten digits, and minutes' one digit. Because triple notes are played with a different tone for each of these three variables, the listener can first count the hour of the day, say the number out loud, count the minutes' ten digit

and say it out loud, then count the minutes' one digit and say it out loud. The listener knows that it is morning (am) if the tones are ascending, and afternoon (pm) if the tones are descending.

The longest number to count out is 12:59, which takes 11 seconds to represent with this encoding scheme. After the clock has signaled 12:59, it chirps out the next eight seconds to indicate the passing of time and for a definite separation between readings. Therefore, the entire time of day is known by the tones generated in any 19-second block of recorded information. Much less time is required for smaller number readings. Decrease the number of second chirps between readings by eliminating some of the repeated "case" statements in the code and adjusting the last case to be in sequence with the case directly before it.

Figure 2 Program Flowchart

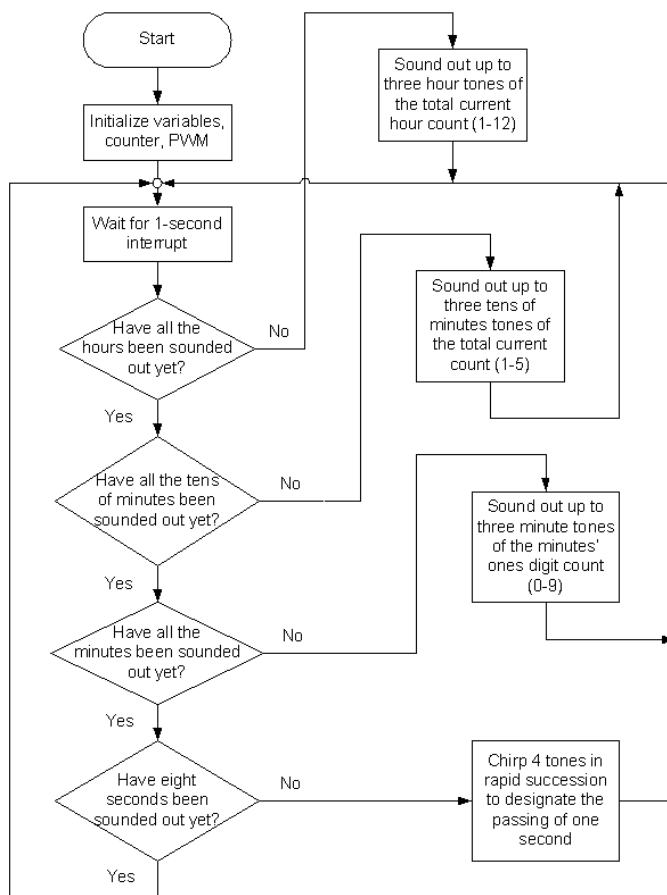


Figure 3 PSoc Digital Block Routing in Chip View

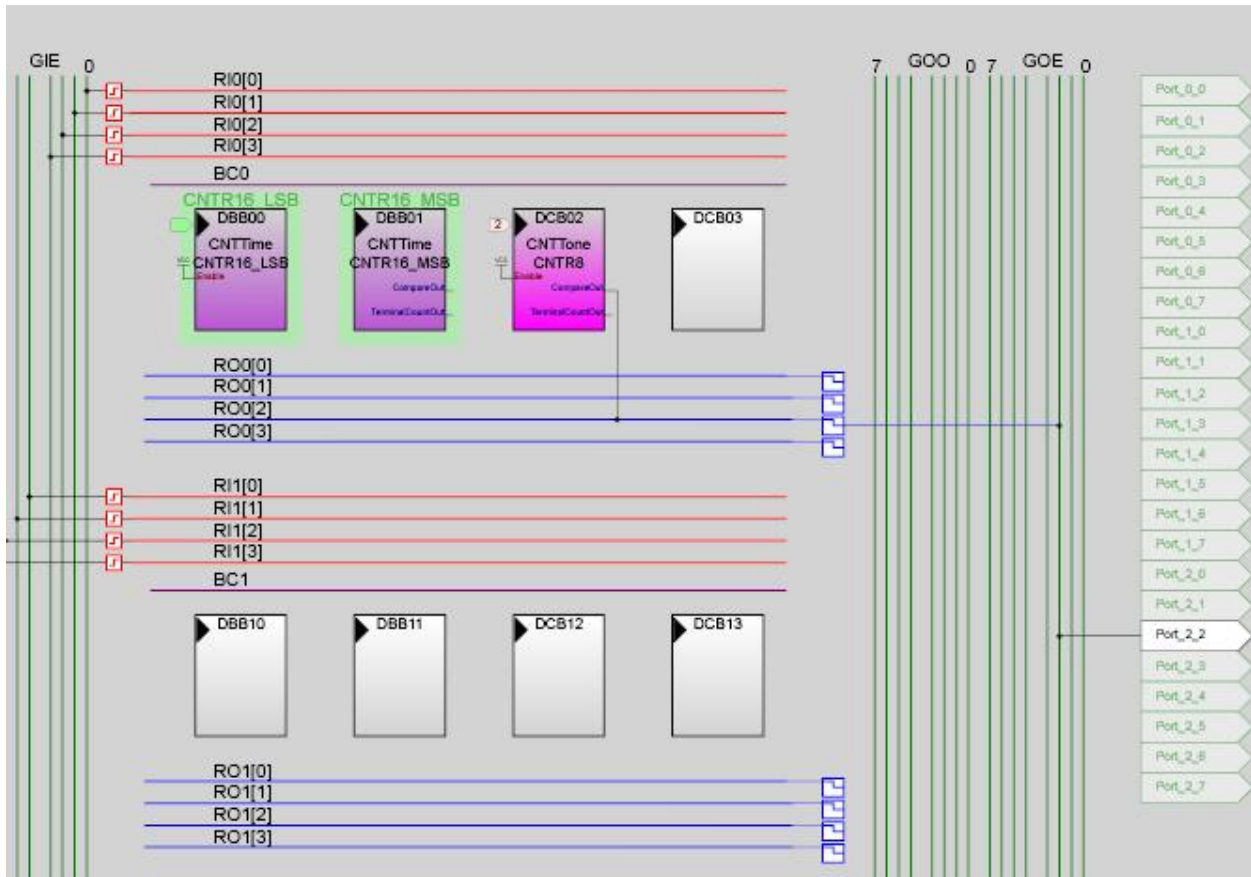


Figure 4 User Module Parameter Settings

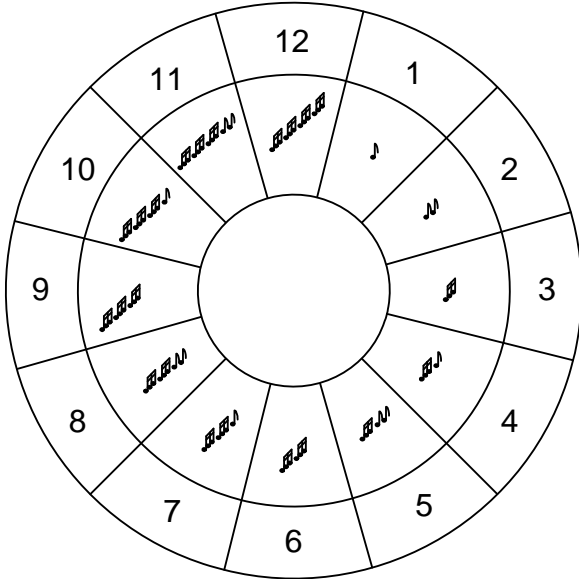
Properties - CNTTime		Properties - CNTTone	
Name	CNTTime	Name	CNTTone
User Module	Counter16	User Module	Counter8
Version	2.5	Version	2.5
Clock	CPU_32_KHz	Clock	VC2
Enable	High	ClockSync	Unsynchronized
CompareOut		Enable	High
TerminalCountOut		CompareOut	Row_0_Output_2
Period	31999	TerminalCountOut	None
CompareValue	10	Period	74
CompareType	Less Than Or Equal	CompareValue	37
Interrupt Type	Terminal Count	CompareType	Less Than Or Equal
ClockSync		Interrupt Type	Terminal Count
InvertEnable	Normal	InvertEnable	Normal

Figure 5 PSoC Global Resource Settings

Global Resources - an2187		Pinout - an2187	
CPU_Clock	93.75_KHz (SysClk/256)	P0[0]	Port_0_0, StdCPU, Pull Down, DisableInt
32K_Select	External	P0[1]	Port_0_1, StdCPU, Pull Down, DisableInt
PLL_Mode	Disable	P0[2]	Port_0_2, StdCPU, Pull Down, DisableInt
Sleep_Timer	1_Hz	P0[3]	Port_0_3, StdCPU, Pull Down, DisableInt
VC1= SysClk/N	16	P0[4]	Port_0_4, StdCPU, Pull Down, DisableInt
VC2= VC1/N	2	P0[5]	Port_0_5, StdCPU, Pull Down, DisableInt
VC3 Source	SysClk/1	P0[6]	Port_0_6, StdCPU, Pull Down, DisableInt
VC3 Divider	1	P0[7]	Port_0_7, StdCPU, Pull Down, DisableInt
SysClk Source	Internal 24_MHz	P1[0]	Port_1_0, StdCPU, High Z, DisableInt
SysClk*2 Disable	No	P1[1]	Port_1_1, StdCPU, High Z, DisableInt
Analog Power	All Off	P1[2]	Port_1_2, StdCPU, Pull Down, DisableInt
Ref Mux	(Vdd/2)+/-BandGap	P1[3]	Port_1_3, StdCPU, Pull Down, DisableInt
AGndBypass	Disable	P1[4]	Port_1_4, StdCPU, Pull Down, DisableInt
Op-Amp Bias	Low	P1[5]	Port_1_5, StdCPU, Pull Down, DisableInt
A_Buff_Power	Low	P1[6]	Port_1_6, StdCPU, Pull Down, DisableInt
SwitchModePump	ON	P1[7]	Port_1_7, StdCPU, Pull Down, DisableInt
Trip Voltage [LVD (SMP)	4.81V (5.00V)	P2[0]	Port_2_0, StdCPU, Pull Down, DisableInt
LVDThrottleBack	Disable	P2[1]	Port_2_1, StdCPU, Pull Down, DisableInt
Supply Voltage	5.0V	P2[2]	Tones, GlobalOutEven_2, Strong, DisableInt
Watchdog Enable	Disable	P2[3]	Port_2_3, StdCPU, Pull Down, DisableInt
		P2[4]	Port_2_4, StdCPU, Pull Down, DisableInt
		P2[5]	Port_2_5, StdCPU, Pull Down, DisableInt
		P2[6]	Port_2_6, StdCPU, Pull Down, DisableInt
		P2[7]	Port_2_7, StdCPU, Pull Down, DisableInt

Appendix

Figure 6. Audible Time Encoding Scheme



- AM denoted by ascending notes
- PM denoted by descending notes
- Sparse notes have ascending or descending trails
- 15-second (quadrant) gives time of day
- Constant second ticks
- One quadrant identifier note
- Two (4 second) measures
- Three (2 second) buffers

About the Authors

Name: Chris and Vincent Paiano
Title: B.S., Computer Engineer and
 Electronic Engineer

Background: 22+ years programming/computer
 experience.
 40+ years electronics/design and
 troubleshooting experience.

Contact: psoc@chrispaiano.com
engineering@chrispaiano.com

Document History

Document Title: Consumer/Industrial – Audible Clock

Document Number: 001-34564

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	1505943	MKEA	09/26/07	New publication of existing application note.
*A	2641070	JVY	01/20/09	Updated content. Added part numbers CY8C21xxx, CY8C23x33, CY8C24xxx, CY8C27x43, and CY8C29x66. Updated software version to PSoC Designer 5.0.

In March of 2007, Cypress recataloged all of its Application Notes using a new documentation number and revision code. This new documentation number and revision code (001-xxxxx, beginning with rev. **), located in the footer of the document, will be used in all subsequent revisions.

PSoC is a registered trademark of Cypress Semiconductor Corp. "Programmable System-on-Chip," PSoC Designer, and PSoC Express are trademarks 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
<http://www.cypress.com/>

© Cypress Semiconductor Corporation, 2004-2009. 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.