

## APPLICATION NOTE 4242

## Using the MAXQ3210 in an Audio Attenuator Circuit

*Abstract: This application note describes a system that implements an automated, stereo audio attenuator. The MAXQ3210 low-power 16-bit RISC microcontroller is the system controller that generates various timing intervals and drives the gradual attenuation of an audio signal ending with a muted condition. The attenuation is created using a DS1801 65-step, audio taper, dual digital potentiometer. Signal buffering is provided by a MAX1467 dual, single supply operational amplifier. Source code for the firmware is available for download from the Maxim website.*

### Overview

Many entertainment devices today are designed with the ability to cease operation automatically without direct human intervention. For instance, it is not uncommon for modern radios or televisions in the home to turn themselves off at a specific time or after a predetermined period. This capability is convenient for implementing a "sleep" mode to allow the device to power-down automatically after the user has fallen asleep.

Unfortunately, not all entertainment devices contain such a capability; component stereo systems generally do not offer this feature. In such cases, it may be desirable to add sleep-mode capability through an external device. Simply turning off the power would not be desirable for such a system. That would abruptly end the audio, and potentially initiate a disruptive transition to silence. It would be preferable for the audio level to decline over a relatively long period, thus producing a gradual, nearly imperceptible elimination of the audio. The inexpensive device described in this application note provides this capability.

This device is designed to be inserted into an audio stream of a component stereo system somewhere between the audio source and the final power amplifier. The device expects a line-level signal in and out, and can, therefore, be placed anywhere in the system where this signal level is conveniently accessible. When initially powered up, this device passes the audio signal without any attenuation, and continues in this way until its "start" button is pressed. This action begins an initial time delay that keeps the device in this "full output" condition for a preestablished countdown interval. After this countdown interval, the device begins to increase the audio attenuation in smooth, equal steps until the signal is completely muted.

### System Hardware

A schematic of the system hardware is shown in **Figure 1**. As can be seen, the system consists of only a few active components: the [MAXQ3210](#) RISC microcontroller, the [DS1801](#) dual audio taper potentiometer, and the [MAX4167](#) dual single-supply operational amplifier. Each of these components and its function within the system is described in the following sections.

### MAXQ3210 Microcontroller

The MAXQ3210 controls the system. The MAXQ3210 is a 16-bit, RISC machine that runs at a maximum clock rate of 3.58MHz. Because almost every instruction executes in 1-clock cycle, the processor's performance approaches 1 million instructions per megahertz. Therefore, the processor has a peak performance of nearly 3.58 million instructions per second (MIPS). The MAXQ3210 contains 1K 16-bit words of EEPROM for program storage and 128 words of EEPROM, plus 64 bytes of SRAM for data storage. The EEPROM memory is configured to be programmed in-system using the device's built-in bootstrap loader or in-application with user-provided code. The microcontroller also contains a 9V to 5V voltage regulator, a piezoelectric horn driver, and up to 15 general-purpose I/O (GPIO) pins.

A version of the MAXQ3210 processor, the [MAXQ3212](#), is identical except that it does not contain the piezoelectric horn driver or the 9V to 5V regulator. The MAXQ3212 would also be a good choice for this application since neither the horn driver nor the regulator are required. The MAXQ3212, therefore, represents an even lower-cost solution.

Ultimately, the MAXQ3210 was chosen for the prototype in this application to allow for maximum system flexibility and possible future enhancements.

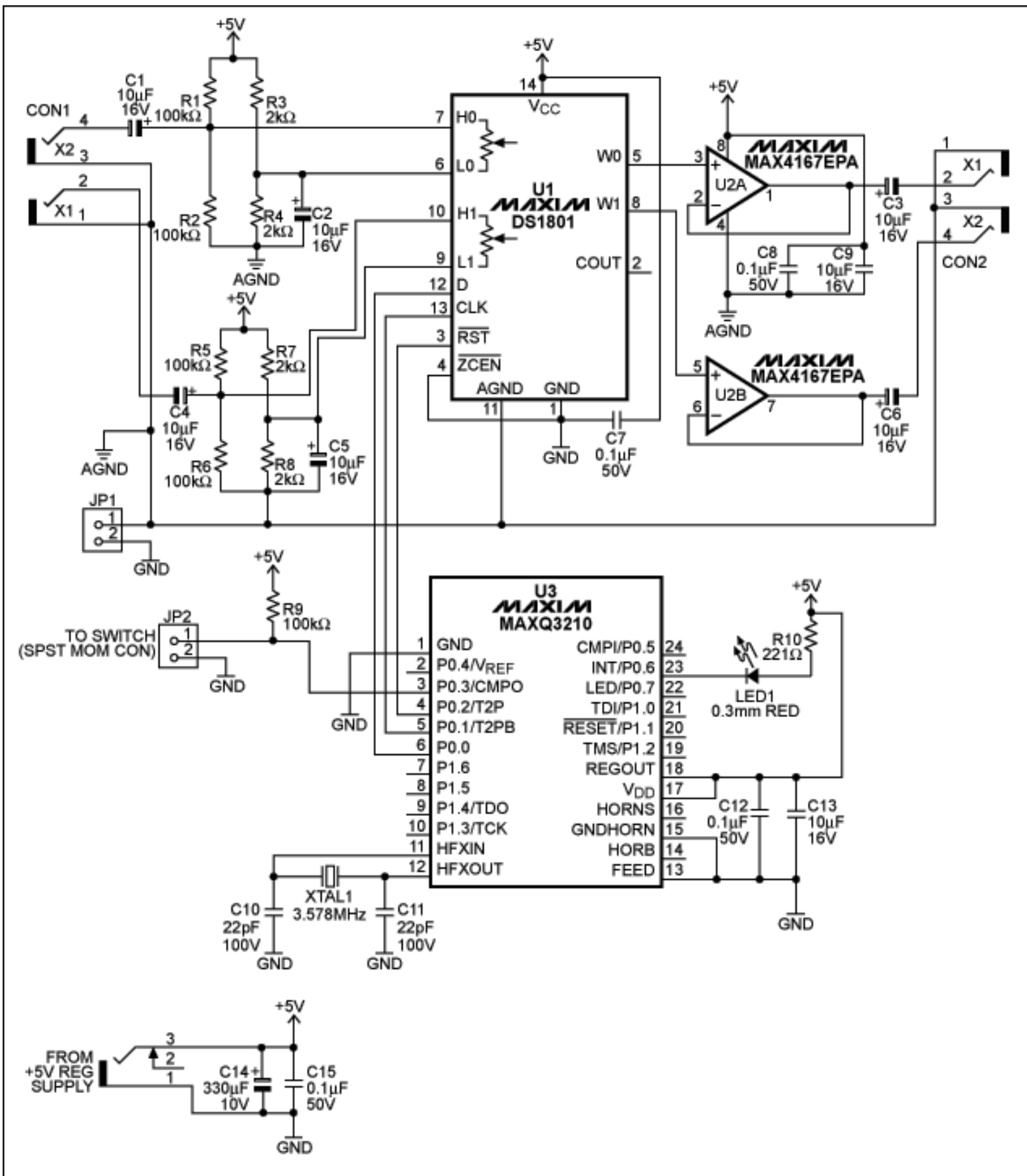


Figure 1. System schematic.

The MAXQ3210 also contains one I/O pin that can directly drive an LED. This application uses the direct-LED drive capability to flash an LED at different rates to indicate system status. This capability is described in more detail in the System Status Indication section below.

## DS1801 Digital Potentiometer

Maxim offers a wide selection of digital potentiometers. The DS1801 was chosen for this application for several reasons. This device contains two, audio taper digital potentiometers, each of which has 65 discrete wiper positions. Positions 0 through 63 represent that level of attenuation (i.e., from 0dB to -63dB attenuation). Position 64 is a special "mute" position which offers greater than -90dB attenuation.

Each of the potentiometers uses an 8-bit shift register to hold the wiper position setting. This is illustrated in **Table 1**. Of these 8-bits, the lower 6 represent the pot's wiper position from 0 to 63. When set to 1, the seventh bit forces the mute condition, regardless of the other bit values. The eighth bit of each shift register is a "don't care," and has no effect on the potentiometer's setting. To set the wiper position of either or both pots, all 16 bits must be shifted into these registers. The arrangement of the shift register and their bit definitions is also given in Table 1.

**Table 1. DS1801 Digital Pot Registers**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Don't Care	Mute	Pot 1 Value MSB	Pot 1 Value	Pot 1 Value	Pot 1 Value	Pot 1 Value	Pot 1 Value LSB	Don't Care	Mute	Pot 0 Value MSB	Pot 0 Value	Pot 0 Value	Pot 0 Value	Pot 0 Value	Pot 0 Value LSB

The DS1801 was also selected because it minimizes noise resulting from large position changes of the potentiometer's wiper. This device contains a "zero crossing detector" that allows the wiper to change positions only when the signal across the potentiometer is zero or after a 50ms timeout. By limiting the wiper transition to the point at which the signal is zero, sudden amplitude changes caused by wiper transitions can be eliminated. If the DS1801 has not detected a zero crossing at 50ms after a new wiper position is written, the wiper will change positions regardless of the state of the input signal. As shown in the schematic of Figure 1, the digital pot is connected so that the DC bias will be applied equally to both ends of the pot. This allows the AC signal to appear to pass through zero volts, even though it is riding on a DC bias of half the supply voltage. Therefore, the zero-crossing detector of the digital potentiometer will operate as intended. The active-low ZCEN pin of the DS1801 is tied low to enable the zero-crossing detector circuitry.

Finally, the DS1801 was selected because of its control interface, a 3-wire serial interface. Designed for CPU-controlled applications, this interface allows the potentiometer's wiper position to be written into the two 8-bit registers described above. Three of the microcontroller's GPIO pins are configured as outputs to provide this interface. The DS1801 3-wire interface consists of three inputs: active-low RST, CLK, and D. The active-low RST signal is used to enable 3-wire serial-port write operations. The CLK is a clock signal that provides synchronization for data transfers. The D signal is data which communicates potentiometer wiper position settings into the device. With its 15 GPIO pins, the MAXQ3210 easily supports this 3-wire interface.

## MAX4167 Operational Amplifier

The MAX4167 is a dual version of the MAX4166 operational amplifier. It is a single-supply (+2.7V to +6.5V) device that offers rail-to-rail inputs and outputs. In this application, it operates from a single 5.0V supply. It has a typical gain-bandwidth product of 5MHz and a typical slew rate of 2V/ $\mu$ s.

The MAX4167 provides signal buffering to the audio signal. In the configuration shown, the device provides unity gain so it acts as a simple buffer.

The line-level audio signal is AC-coupled into and out of the system to eliminate the single-supply-induced DC-bias voltage from the audio signal.

## System Software

The software for this application was developed and tested using the MAXQ® Integrated Development Environment, or MAX-IDE, and the [MAXQ3210 Evaluation \(EV\) Kit](#). The MAX-IDE contains all the tools necessary to develop code for various microprocessors, including the MAXQ3210. It supports code development in assembly language using a project-based interface. Projects can be assembled for the target device, downloaded, and executed line by line. When using the MAXQ3210 EV kit, the state of the system can be examined by viewing the contents of its memory, the items contained in the call stack, the values of all its registers, and the values of all the variables accessible in the source code. Breakpoints can also be set within the source code to halt execution at predetermined locations. The MAX-IDE is a free development tool available from the [Maxim web site](#).

The software is straightforward. The application code is implemented in one main assembly language program, Attenuator.asm, which contains initialization functions, a main program loop, several subroutines, and one interrupt service routine. All

source code is available for [download](#) (ZIP, 8kB). This .ZIP archive contains the MAX-IDE project file, all the necessary "include files," and the source code file. Also included is the application's .HEX file that can be loaded and executed on the MAXQ3210 EV board or a custom target board.

## Potentiometer Control

The main function of this software is to control the DS1801 digital potentiometer. At power-up the potentiometer is set to its nonattenuating state. Following the user input (i.e., a button press) and the appropriate count-down delay, the software begins stepping the potentiometer's wiper towards its fully muted position.

The actual positioning of the potentiometer's wiper is controlled by two subroutines; Output\_Bytes and Clk\_Bytes. The Output\_Bytes subroutine collects the single-byte value to be stored in both potentiometer wiper position registers; tests them for validity; and sequentially sends them to the Clk\_Bytes subroutine. The Clk\_Bytes subroutine provides the low-level port-pin manipulation and bit serialization necessary to strobe the wiper position data into the potentiometer. The software was written so that, if another potentiometer with a different interface were selected, replacing these two routines would likely be the only software change required.

## System Status Indication

As previously mentioned, a flashing LED is used to indicate system status. A flash rate of approximately 0.25s indicates that the device has been powered up, but has not yet received a user input to start the countdown interval. A flash rate of approximately 1s indicates that the system has started its countdown mode before beginning its attenuation stepping. A flash rate of approximately 2s indicates that the system is in its stepped-attenuation mode. A flash rate of 5s indicates that the system has reached its fully muted position and is holding there until another power-up occurs. A red LED is supplied with 5V through a current-limiting resistor. The cathode of the LED is connected directly to the MAXQ3210's port 0, bit 7 which is capable of sinking the necessary current. At the intervals described, the port pin P0.7 is XOR'ed with itself, causing it to toggle which, in turn, causes the LED to flash.

## Time-Interval Generation

Another primary function for the microcontroller in this application is measuring time intervals for different functions. From a system-level perspective, there should be at least two, variable, time-interval delays for this application. The first interval must provide a countdown delay, beginning when the user presses the start button and ending when the attenuation progression begins. Intuitively, this time-interval should be on the order of 1 hour to 2 hours to allow the user sufficient time to fall asleep. The second delay must establish a time-interval between the discrete steps of the attenuation process. This interval will effectively determine the time required for the audio to decrease from its unattenuated level until it reaches its final muted condition. Consequently, it will determine the smoothness of the attenuation stepping. Again intuitively, the entire attenuation stepping process should be on the order of 1 minute to 60 minutes to achieve a smooth, subtle decrease in level. Therefore, with 65 discrete steps of the potentiometer, this represents a time-interval between steps from approximately 1s to 55s. Finally, there must also be a relatively short delay for debouncing the pushbutton switch, which should be on the order of 200ms.

To achieve this wide range of time intervals, two subroutines are implemented for this application. The subroutine called Delay creates a variable time-interval based on an 8-bit parameter passed to it. This parameter determines the number of passes through a 10ms software timing loop. Since this parameter can have values from 0 to 255, this subroutine produces time intervals up to 2.55s. A software timing loop produces the time-interval at an assumed processor clock frequency of 3.58MHz. This subroutine also produces the debounce delay in the final software implementation.

The second time-interval-generating subroutine is called LDelay, and it is based on counting 1s time intervals produced by the processor's internal Timer 2. The timer is initialized to generate an interrupt at 1s intervals; the interrupt-service routine increments the counter variable TIC. The LDelay subroutine compares TIC to a predetermined value (see next section), and when they are equal, TIC is reset to zero. When TIC is zeroed, the variable TOC is incremented by one. The variable TOC is then compared to another predetermined value. If the values are not equal, the cycle is repeated. If these values are equal, the interval is complete and the timer is stopped. Since both TIC and TOC can be 8-bit values, this subroutine can generate delays up to 18.06 hours ( $255 \times 255 \times 1s = 65,025s = 18.06hr$ ).

## Time-Interval Selection

The software was written so that the values for the two time intervals can be easily changed for the specific application to accommodate varying requirements. Two values are defined in the include file (Attenuator.inc) that allow the user to simply

modify their values to change the two primary system time intervals. The variables TIMEV1b and TIMEV1a hold the 16-bit value (MSB, LSB, respectively) for the countdown time interval (i.e., from button press to beginning the attenuation stepping); TIMEV2b and TIMEV2a hold the 16-bit value (MSB, LSB, respectively) for the time interval between attenuation steps. A simple reassignment of these variables in the .INC file will allow any time interval within the stated ranges to be selected for a given application or environment.

## Conclusion

This document describes a system that can be used to automatically attenuate a line-level audio signal within a component stereo system to provide a sleep-mode function. The audio attenuation begins after a fixed time interval from a user input (pushing a button); the attenuation is automatic, gradual, and evenly stepped. The system is controlled by a MAXQ3210 RISC processor, which manipulates a DS1801 dual audio taper digital potentiometer to attenuate the signal. A single-supply, dual operational amplifier, the MAX4167, is used to buffer the signal. The signal is AC-coupled into and out of the system to separate it from the internal bias voltages. The software is written in assembly language, and tested using the MAX-IDE (MAXQ Integrated Development Environment) and the MAXQ3210 EV kit board. All software necessary to assemble and create a loadable, executable .HEX file is available for download from the Maxim website.

MAXQ is a registered trademark of Maxim Integrated Products, Inc.

---

Application note 4242: [www.maxim-ic.com/an4242](http://www.maxim-ic.com/an4242)

### More Information

For technical support: [www.maxim-ic.com/support](http://www.maxim-ic.com/support)

For samples: [www.maxim-ic.com/samples](http://www.maxim-ic.com/samples)

Other questions and comments: [www.maxim-ic.com/contact](http://www.maxim-ic.com/contact)

---

### Automatic Updates

Would you like to be automatically notified when new application notes are published in your areas of interest? [Sign up for EE-Mail™](#).

---

### Related Parts

DS1801: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

MAX4167: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

MAXQ3210: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)

MAXQ3212: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)

AN4242, AN 4242, APP4242, Appnote4242, Appnote 4242

Copyright © by Maxim Integrated Products

Additional legal notices: [www.maxim-ic.com/legal](http://www.maxim-ic.com/legal)