

## APPLICATION NOTE 3914

# Using the MAXQ3212 Microcontroller as an AC-Power Sequencer

*Abstract: The MAXQ3212 microcontroller is ideally suited for real-world control applications because of its small size, low cost, and ready availability of a complete, low-cost development system. This application note describes a system based on the MAXQ3212 that is designed to sequence AC power to a set of stereo components. The system eliminates objectionable and potentially damaging transients during system power-up or power-down. The MAXQ3212 RISC microcontroller uses port pins as I/Os, and generates signals to activate relays to switch the AC power to two separate groups of stereo components.*

## Overview

It is sometimes advantageous to sequence power to a system of individual components in a particular order. It may also be desirable to insert time delays between the application or removal of power to the components to help ensure stability. A specific example of such a situation is a component stereo system illustrated in **Figure 1**. This technique of sequentially applying power and inserting time delays is beneficial because it can eliminate potentially damaging transients from reaching sensitive components like the speakers' high-frequency drivers. The components in Figure 1 are separated into two groups: the signal sources and pre-amplifier, referred to generally as the pre-amp, represent one group; the power amplifier and speakers, referred to generally as the speakers, represent the other. In such a system, it is desirable to apply power to the pre-amplifier first (AC Power A) and delay a period of time so their outputs stabilize before power is applied to the speakers (AC power B). By doing this, objectionable audible pops and/or potentially damaging transients can be eliminated. For power-down, identical concerns apply, but the sequence should be reversed to remove power from the speakers before power is removed from the pre-amplifier.

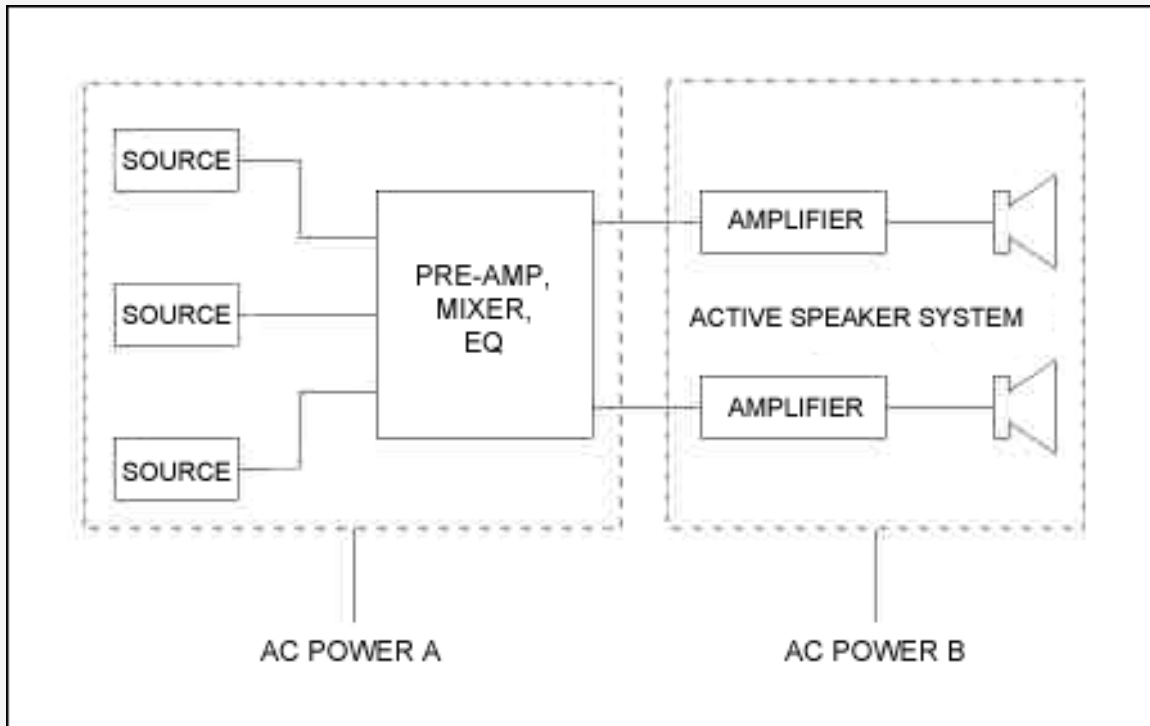


Figure 1. Typical component stereo system.

This application note describes a simple system based on the [MAXQ3212](#) RISC microcontroller that sequences power to the component stereo system illustrated in Figure 1. The MAXQ3212 was chosen for this project because of its

small size and low cost. It was also important that a complete, low-cost development system be readily available. This latter requirement was met by the MAXQ3210 evaluation (EV) kit. The MAXQ3212 was selected over the MAXQ3210 for this project because the MAXQ3210's voltage regulator and the piezoelectric horn driver are not used in this application. To keep the design simple and minimize potential operational issues, small power relays are used to switch the AC power. The system software senses user inputs from single-pole, single-throw (SPST) switches and uses processor port pins configured as outputs to control the relays. Light-duty power FETs interface the processor's port pins to the relays' coils.

The complete development environment for this example code is version 1.0 of the [MAX-IDE](#) and revision A of the [MAXQ3210 kit board](#).

## System Implementation Details

The implementation for this system is quite simple. To avoid any issues with reactive loading on the AC power lines, relays with 5V coils are used to switch the AC power instead of solid-state devices such as triacs. The system implemented here uses an external, regulated, 5V supply, and this voltage is connected directly to the relays' coil. The current to the relays' coil is switched using small (TO92 package) power FETs, and the control for the FETs is supplied by a standard, output port pin on the MAXQ3212 processor. A schematic of the system is shown in **Figure 2**.

## User Inputs

In the prototype of this system, a single switch allowed the user to turn the entire system on and off. It became evident, however, that having the ability to apply power to the pre-amplifier but not the speakers would be desirable. This flexibility would allow the use of headphones, which are driven by the pre-amplifier, without activating the speakers. To accomplish this, a speaker-disable switch is added to the system. This system tests this switch immediately before power is applied to the speakers. If the switch is active, no power is applied to the speakers; if this switch is not active, power is applied to the speakers as usual.

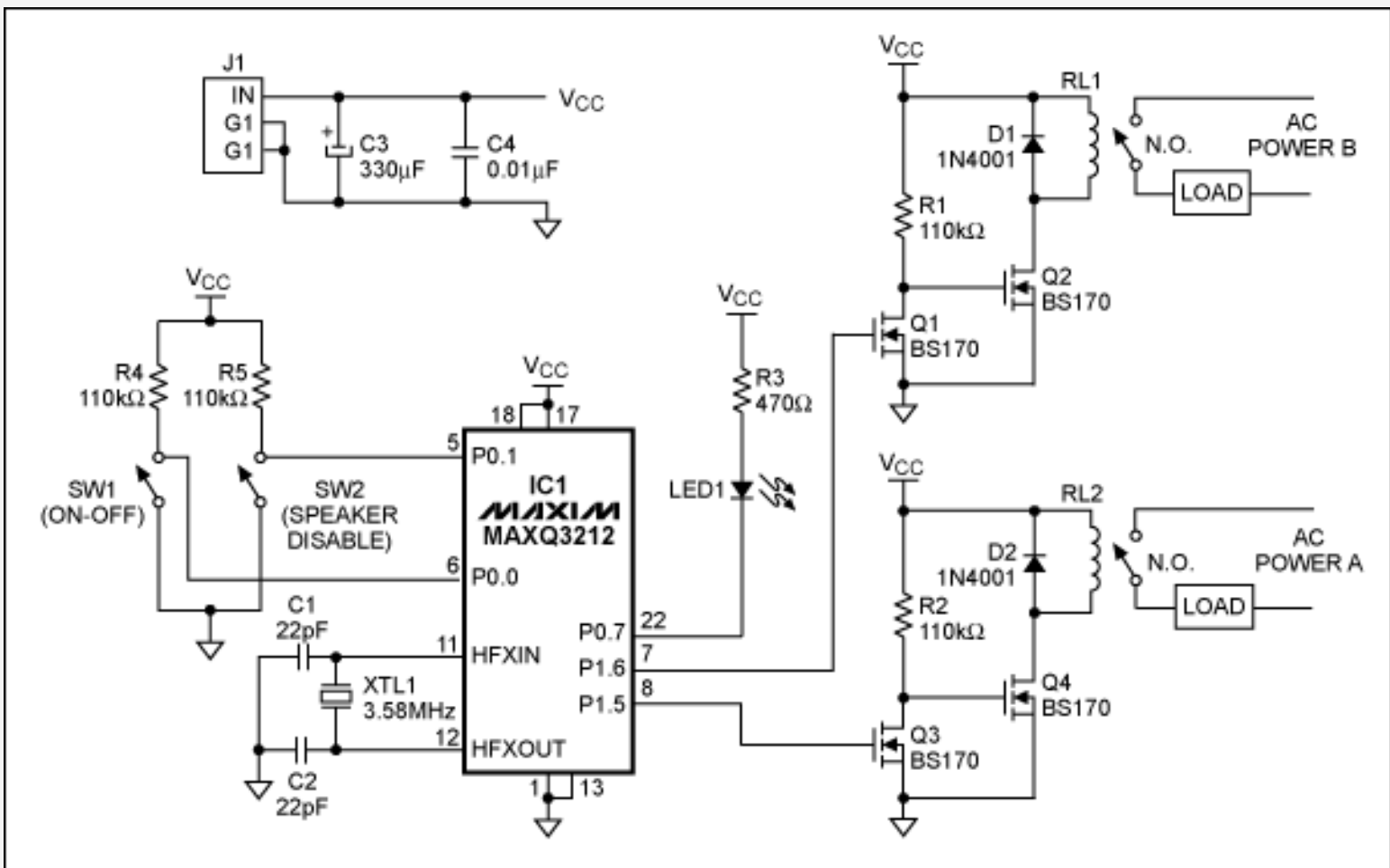


Figure 2. MAXQ3212 system schematic.

## LED Status Indicator

In this system, the direct LED drive capability of MAXQ3212's port pin, P0.7, provides a system status indicator. As the system software passes through its main loop, this port pin is toggled from its previous state. Because this pin is connected directly to an LED pulled high by a 470Ω current-limiting resistor, this polarity change of the port pin effectively flashes the LED every other pass through the main program loop. In an idle system (i.e., waiting for a user input), the LED flashes approximately twice per second. When the software enters a delay loop, the LED remains in its current state until the delay loop is exited. The system status can, therefore, be interpreted from the behavior of the LED.

## Dual-FET Coil Drive

In the initial prototype of the system, a single FET was used between the port pin and the relay coil. As DC power was applied to the microcontroller and it reached stability, the relay would activate briefly as the processor's port pin passed from its reset-default high state to the firmware-directed low state required to deactivate the relay. By adding a second FET, the reset-default condition of the port pin keeps the relay deactivated without any firmware intervention. Thus undesirable transient behavior is eliminated.

## System Status Byte

In the system's software, a data byte is used to reflect the current system status. Bits within this byte are used to indicate the status of individual input and output signals. Therefore, this byte must be read-, write-, and bit-addressable. In the MAXQ3212, all registers in modules 0 through 2 are bit-addressable. The Wake Up Timer register (WUT) is located in module 0 (M0[15]), but this timer is not used in this particular application. This register is, therefore, selected as the location for system status byte. In the software, bits within this byte are assigned labels.

The bits are read or written individually so they can be set or cleared as necessary to reflect the status of the port pins which they represent.

## Time-Delay Generation

To generate known time intervals in this system, the MAXQ3212's Timer 2 is used. As a 16-bit timer with a 3.58MHz main system crystal, this timer can generate intervals up to 2.34 seconds. For simplicity, a basic time interval of one second is used, and a system variable, TIC, is used to count the one-second intervals. The timer's reload value is given by the equation below. It can be seen that for an interval of one second, a reload value of 37583 (rounded to the nearest integer) is needed.

$$\text{RELOAD} = 65536 - (\text{Time} \times 3.578\text{MHz}/128)$$

The timer is configured to cause an interrupt at the end of the one-second interval; the TIC variable is incremented every interrupt. Therefore, comparing TIC with the desired value allows the generation of time intervals much longer than the timer's maximum interval. Changing the delay time is a simple matter of changing the parameters passed to the delay subroutine, and reassembling the code.

## Selectable Delay

The delay period is designed to be a modifiable system parameter to accommodate situations where different delays are required for power-up versus power-down. In the prototype of this system, it was observed that the time required for the pre-amplifier's outputs to stabilize following power-up was a significantly shorter time than the period required for the power to "bleed off" the speakers when powered down. Therefore, in the software related to this document, a 5-second delay is used for the speaker power-up, and a 10-second delay for the pre-amp power down.

## Example Code

The software accompanying this application note is contained in the file an3914\_sw.zip that is available on the Maxim ftp web site at: [ftp://ftp.dalsemi.com/pub/microcontroller/app\\_note\\_software/](ftp://ftp.dalsemi.com/pub/microcontroller/app_note_software/). This file contains the source code file, `Sequencer.asm`, and the file `maxq3210.inc` that contains the MAXQ3210 and MAXQ3212's register address definitions. Also included in this .zip file is the MAX-IDE project file `Sequencer.prj` and the loadable hex file, `Sequencer.hex`. By extracting these files to a common directory, the software can be assembled and executed on the MAXQ3210 EV kit.

## Conclusion

The MAXQ3212 is a high-performance RISC processor suitable for a wide range of applications. Because of its low cost and low power, it is particularly well-suited to battery-powered, portable and environmental sensor/alarm applications. In the application presented here, the MAXQ3212's low-cost and low-pin-count package made it attractive for this simple AC power-control application. Also, because of the low-cost and ready availability of a complete development environment, i.e., the MAXQ3210-KIT, the firmware could be developed and tested quickly and effectively.

---

Application note 3914: [www.maxim-ic.com/an3914](http://www.maxim-ic.com/an3914)

### 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**

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

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

AN3914, AN 3914, APP3914, Appnote3914, Appnote 3914

Copyright © by Maxim Integrated Products

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