

## USB FM RADIO

### 1. Introduction

This USB FM Radio design is intended as a reference for incorporating FM radio functionality into a USB product. The design consists of two major components, the Si4701 FM radio receiver and the C8051F321 microcontroller with a built-in universal serial bus (USB) peripheral. Due to the high level of system integration of both of these components, the total design is very small with fewer external components than many other solutions.

The software, firmware, schematic, and layout source for this design are located in AN264SW.zip. The latest version of these files as well as this document itself can be found at the following URL:

<http://www.silabs.com/usbradio>

### 2. System Overview

The system consists of a Windows® application that communicates with the C8051F321 using the USB connection. The C8051F321 microcontroller controls the Si4701 using the serial peripheral interface (SPI). The Si4701 audio outputs are sampled using the C8051F321's analog-to-digital converter (ADC) and sent to the host across the USB interface. The Windows application plays the audio using the PC speakers or headphones.

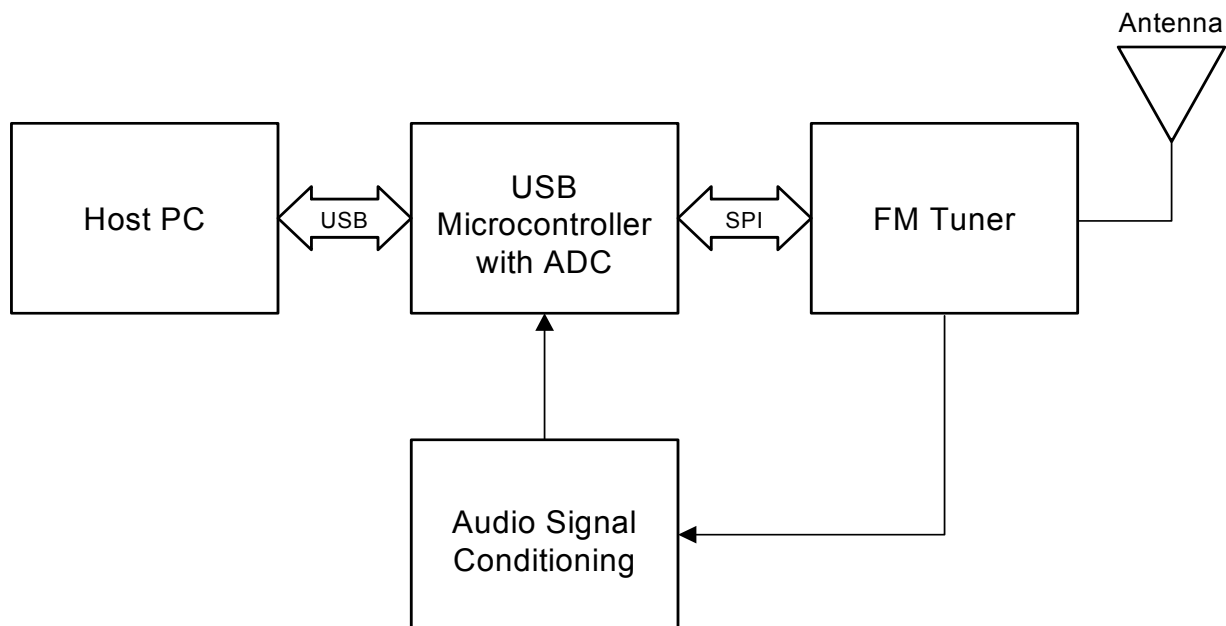


Figure 1. System Block Diagram

## 3. Hardware

The schematic and layout for the USB FM Radio design are located in "Appendix A—Schematics" on page 4 and "Appendix B—Layout" on page 6. The board consists of two layers with parts on one side only.

**ESD Protection**—The design includes electrostatic discharge (ESD) protection on both the antenna and USB connection using Littlefuse's SP0503BAHT and California Micro's CM1213, respectively.

**Analog Audio Filter**—The Si4701 audio outputs are amplified and filtered before the C8051F321's ADC samples them. The dual package op-amp, ON Semiconductor's MC33202, is used to create a single-pole low-pass filter with a corner at 23405 Hz and a gain of 3.

**Antenna Input**—The antenna signal is conditioned using C1, C20, and L1 as recommended in "AN231: Si4700/01 Headphone and Antenna Interface."

**Reference Clock Generation**—The 32.768 kHz reference clock for the Si4701 is generated using a watch crystal. The C8051F321 drives the crystal to oscillate and runs this signal through an asynchronous comparator creating the required reference clock.

## 4. Firmware

The firmware listings for the C8051F321 can be found in AN264SW.zip, under the "Firmware" directory. The provided firmware has been developed using the Keil C compiler, assembler, linker, and the Silicon Laboratories IDE.

The overall organization of the firmware is straightforward. It begins by initializing all necessary device peripherals and the Si4701 FM tuner. Once this is complete, the microcontroller operates in one of two modes, command or audio. In command mode, the device can accept control commands from the USB host for operations such as seeking, tuning, or saving presets. In audio mode, the microcontroller sends ADC samples and a small amount of status information up to the host. The MCU enters audio mode once the host has started polling for audio data, and exits when it receives any command from the host.

**USB Profile**—This device presents four different interfaces to the host when it enumerates. Three of these are active at any given time. The audio control interface is used by the host to control different parts of the audio functions. There is also a zero-bandwidth streaming interface and a full-bandwidth streaming interface. Only one of these interfaces is used at a time. Finally, there is a Human Interface Device (HID) interface used for all other required device communication, such as controlling the Si4701 and upgrading the C8051F321 flash image.

**ADC Sampling**—The 10-bit ADC on the C8051F321 is used to sample both the audio channels. The ADC sample rate is 192 kHz, and it switches between the right and left channel between successive conversions using the on-chip ADC multiplexer. The audio sample rate for stereo samples is consequently 96 kHz, and each audio sample is 4 bytes in size. These are sent to the host in signed two's complement format.

**USB Transfer Type**—The audio streaming interface uses isochronous USB transfers to send audio data to the host. Since the audio sample rate is 96 kHz and each sample is 4 bytes in size, the device transfers 384 bytes per millisecond across the bus.

**Microcontroller Flash Upgrade**—The firmware includes a command interpreter that is capable of upgrading the device's flash image. This code is capable of parsing HID commands to read/write and check integrity of flash pages. However, it is not a true boot loader because it not functional without a valid external code image to handle USB enumeration. This means that if the flash upgrade process fails, the device will be inoperable.

## 5. Windows Software

The graphical user interface (GUI) software is located in AN264SW.zip, under the “GUI” directory. This software was developed in C++ using Visual Studio 6.0.

When the GUI is first launched, it begins by looking for a USB FM Radio device. Once a device is found, it is opened up as a USB Audio Device for audio streaming and as a HID device for FM Tuner control. In addition, the default WAV output device is opened to playback the audio stream.

**Audio Stream**—The host application uses a circular buffer of 20 32 kB blocks to accept audio from the USB FM Radio device. Initially, 19 of the buffers are filled with audio data. Next, this data is sent to the WAV output device, and as streaming continues, pointers to the first free block and first playable block are incremented around the circular buffer to provide a constant and seamless flow of audio.

Audio will stream until a user action occurs such as tune, seek, or an alteration of radio settings. When this occurs, the audio stream will stop, and the commands for the requested action are sent down through the HID interface. Once the acknowledgement of the command comes back, the GUI is updated and stream playback will continue.

**Device Status**—In addition to the audio streaming, the device is also polled for its current attributes such as the current station, the received signal strength, and a stereo/mono indicator. As this information changes, it will also change on the display of the GUI.

**Device Removal**—Audio streaming and polling will stop if the device is disconnected from the USB. In this case, all of the device handles are closed, and the host application will begin to poll for connected devices.

**USB Device Driver**—Because Windows has built-in drivers for USB audio and HID class devices, the USB FM Radio does not require an OEM supplied driver disk for operation. The player executable is the only required user software. A similar approach is possible for other operating systems, if they have built-in support for these two USB device classes.



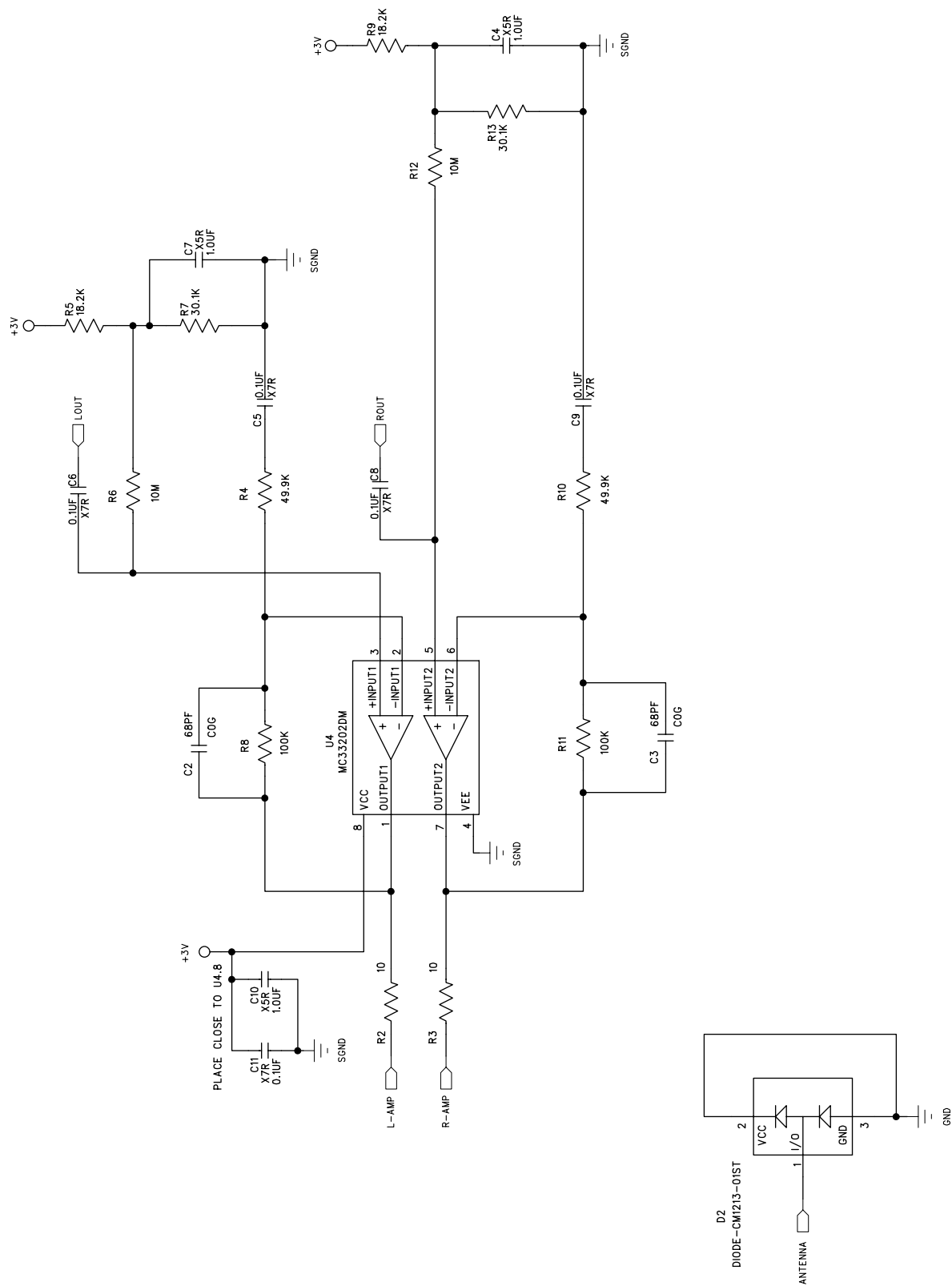


Figure 3. Schematic Page 2