Power Over Ethernet Voice Transmission Reference Design

Relevant Devices

This application note applies to the following devices:
Si3400, C8051F340, and CP2201.

1. Introduction

Power Over Ethernet is used in many applications where it is not economical or convenient for separate power cables to exist. It allows 48 V power to be delivered using the same CAT5 Ethernet Cable used for carrying data. Many security cameras, intercoms, and Voice Over IP telephones benefit greatly by having both power and data delivered on a single Ethernet cable. This reference design demonstrates the following functions:

- Using Power Over Ethernet to power an embedded system with Ethernet connectivity.
- Implementing Automatic Gain Control and G.711 Sound Compression on a C8051F340.
- Transmitting sound over an Ethernet network.
- Implementing a web browser interface for displaying embedded sensor data.
- Sending E-Mail from the embedded system.
- Remotely updating firmware using a TFTP bootloader.

This application note describes the design of the PoE-VOICE-RD hardware and firmware. Refer to the “PoE Voice Transmission Reference Design Kit User’s Guide” for step-by-step demonstration instructions.

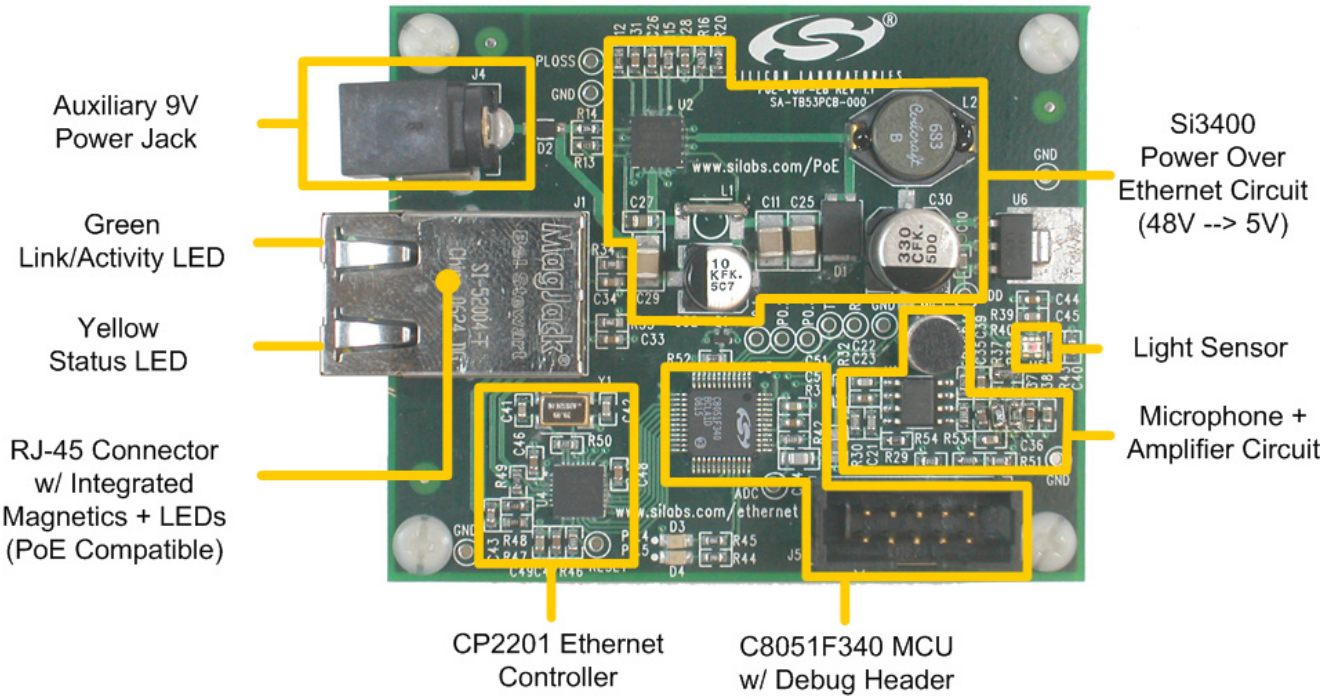

Figure 1. Power over Ethernet Voice Transmission Reference Design Board

2. Hardware Implementation

2.1. Power Over Ethernet

The reference schematic for the PoE-VOICE-EB is shown at the end of the “PoE Voice Transmission Reference Design Kit User’s Guide”.

The Si3400 is a highly integrated solution for Power over Ethernet which complies to IEEE 802.3 (formerly Ieee802.3af-2003) requirements for providing power via either the Ethernet data pairs or the spare pair in an Ethernet cable. The Si3400 contains all of the circuitry including input diode bridges, a “hot swap switch” for under voltage lock out, soft start circuitry and a dc to dc converter and switcher FET. It can deal with the full range of power specified in IEEE802.3 and is useful for applications requiring as much as 10 watts of regulated power. It can be adapted for most practical output voltage and current requirements by changing one or two resistors. More detailed information can be found in the Si3400 data sheet and in Application Note “AN296: Using the Si3400 PoE PD Controller in Isolated and Non-Isolated Designs”. Reference designs, evaluation boards, and design spreadsheets are also available. Power Over Ethernet documentation is available from www.silabs.com/PoE.

If you would like to adapt this reference design for other applications, it is lowest risk to use a reference schematic similar to the evaluation platforms available for the Si3400. If you have more specific requirements, please contact your local sales representative for design assistance.

2.2. Ethernet Connector with PoE Support

The integrated Ethernet connector selected is a Belfuse SI-52004-F. This connector is unique in that in provides access to the 4 unused wires in the Ethernet cable (used for delivering power) and has a transmit turns ratio of 2.5:1. This allows both the PoE (power) function and the Embedded Ethernet (data) function to be combined into a single integrated connector.

An alternative to using this connector is to use a standard RJ-45 jack with discrete magnetics. A standard RJ-45 jack provides access to all 8 wires in the Ethernet cable. This alternative method will consume more PCB area than using an integrated connector.

2.3. MCU and Ethernet Controller

The MCU and Ethernet Controller section is identical to the circuit used in the CP2201 Evaluation Kit, except a 10-pin debug header has been added to allow software development. It uses a C8051F340 and CP2201 to provide the embedded system with Ethernet connectivity.

If you would like to adapt this reference design for other applications, we recommend Application Note “AN292: Embedded Ethernet System Design Guide” as a starting point for new Embedded Ethernet designs.

2.4. Embedded Sensors

The PoE-VOICE-EB contains a light sensor, microphone with amplifier, and uses the on-chip temperature sensor on the C8051F340 to demonstrate the ability to remotely access sensor data from a web browser.

3. Software Implementation

The software for this reference design was generated using TCP/IP Configuration Wizard version 3.1. The generated project used to begin application software development contained support for DHCP, Netfinder, TCP, UDP, HTTP and SMTP. Additional Application code was then added to make the embedded system perform the functions in the demonstration. For more information about the design process for implementing Embedded Ethernet, we recommend reading “AN292: Embedded Ethernet System Design Guide”.

3.1. ADC Sampling Engine

The interrupt-based ADC Sampling engine has two modes: low-speed multiplexed and high-speed single-channel. The high-speed mode is used to sample the microphone at 32 kHz, then average the samples for an effective output word rate of 8 kHz. In this mode, timer 3 overflows are used to schedule conversions and read the samples from the ADC data register.

The low speed multiplexed mode is used for sampling the light and temperature sensors. The sampling rate is 300 Hz and oversampling and averaging is used to increase the sample resolution. The output word rate is set to approximately one sample per second.

3.2. Automatic Gain Control

When the ADC sampling engine is sampling the microphone, an automatic gain control stage is enabled which allows the input signal to be digitally amplified by a factor of 2 or 4 based on its peak-to-peak amplitude. A silence detection stage is also enabled to turn off “digital amplification” when no sound is detected. The thresholds for determining the various sound levels are located in the “udpsound.h” header file. The source code for the automatic gain control implementation can be found in “udpsound.c”.

3.3. G.711 Sound Compression

The human ear detects logarithmic changes in sound, therefore, G.711 uses a logarithmic encoding to compress 10–14 bit sound samples down to an 8-bit stream without any significant loss in sound quality. The portion of the audio data lost in the compression are the changes which the human ear cannot easily detect.

The reference design software uses a lookup table approach to convert 12-bit samples down to 8-bits. Since the G.711 lookup table is symmetric for positive and negative values, only 2048 bytes are needed to encode the entire 12-bit range of values. The implementation of G.711 sound compression can be found in “udpsound.c”.

To decode the 8-bit data stream, a reverse lookup table may be used. Please refer to the G.711 and G.191 recommendations available from the Telecommunication Standardization Sector of the International Telecommunication Union http://www.itu.int/ITU-T/ for additional information.

3.4. Sound Transmission over Ethernet

As the ADC sampling engine is capturing a steady stream of samples at 8 kHz, they are being placed into a RAM buffer. This buffer consists of two identical 251 byte buffers. At any given time, one of the buffers is being filled with new samples and the other is being transmitted. The buffers switch roles every 32 ms.

For transmitting the sound buffer over an Ethernet network, the reference design software uses a direct UDP socket. In this application, UDP is more desirable than TCP because packet latency and jitter reduction are more important than guaranteed packet delivery.

The UDP socket on the MCU is at port 50000 and accepts the following single-byte ASCII commands:

- **“S”** - Start packet which will cause the UDP socket to start sending sound data.
- **“A”** - Acknowledge packet that should be received every second. If a period of 5 seconds passes without receiving an acknowledge packet, the UDP socket will time out and reset.
- **“P”** - Pause packet which will cause the UDP socket to stop sending sound data. The socket may be un-paused by sending another Start packet.
  Note: Only the application which paused the device may unpause it.
- **“Q”** - Causes the UDP socket to stop sending sound data and reset the socket. A socket reset frees resources and allows other PCs to connect to the device.

3.5. Sending the PLOSS Signal

The PLOSS signal is an output from the Si3400 and indicates when PoE power is lost. The value of PLOSS is appended to each data packet being sent and transmitted as the 251'st byte of the packet. Using the PLOSS signal, the PC application can determine if the embedded system is powered from PoE or from the auxiliary supply.

3.6. Web Server Interface

Since the HTTP protocol was enabled in the TCP/IP Configuration Wizard when the project was being generated, the embedded system will automatically respond to requests from a web browser. The default web page generated by the TCP/IP Configuration Wizard is a simple "Hello World" page. This simple web page was replaced by a set of web pages and a CGI script to allow viewing sensor data and sending e-mail from a web browser. The web pages are very similar to the ones used in the CP2201EK and described in "AN292: Embedded Ethernet System Design Guide".

One difference between this reference design and the method described in AN292 is that some of the web pages are stored in the CP220x Flash memory. This 8 kB Flash memory can store up to 8 kB of "web server content" and is accessible by the virtual file system. An example project showing how to load web pages into the CP220x Flash is provided in the software examples directory. See Section "5. Accessing the Software" for more information.

3.7. TFTP Bootloader

A TFTP bootloader was added to the system to allow remote firmware updating. This is the same bootloader used for the CP2201EK and turns the embedded system into a TFTP server. Refer to the readme.txt in the bootloader source code for information on how to use the bootloader in your Embedded Ethernet application.

4. Using the PoE-VOICE-EB for Software Development

The PoE-VOICE-EB Reference Design Board can be used as a hardware platform for complete software development. The output of the TCP/IP Configuration Wizard is targeting the standard Target Board + AB4 Ethernet Development Board hardware. To use the output of the TCP/IP Configuration Wizard with the PoE-VOICE-EB, four initialization routines need to be modified. The simplest method is to replace the following four routines with their equivalent routines found in the *PoE_VOICE_Init.txt* file:

- PORT_Init()
- EMIF_Init()
- ether_reset_low()
- ether_reset_high()

To start developing code, you will need to obtain a USB Debug Adapter and Keil Evaluation CD available from the Silicon Laboratories Website at www.silabs.com. Note that the Keil evaluation CD is limited to 4 kB object code generation. The TCP/IP Library does not count towards the 4 kB evaluation limit.

The USB Debug Adapter and Keil Evaluation CD are also included in any MCU development kit.

5. Accessing the Software

Once the software is installed, a new directory named PoE_VOICE_RD will be created in the main "C:\Silabs\MCU" installation directory. This directory will contain the following:

- Documentation
  - AN312 Power Over Ethernet Voice Transmission Reference Design
  - PoE Voice Transmission Reference Design Kit User's Guide
  - PoE Voice Transmission Reference Design Kit Quick Start Guide
  - Datasheets:
    - C8051F34x MCU Family Datasheet
    - CP220x Ethernet Controller Datasheet
    - Si3400 Power Over Ethernet PD Controller Datasheet
  - Power Over Ethernet Design:
    - AN296 - Using the Si3400 PoE PD Controller in Isolated and Non-Isolated Designs
    - Si3400 Switching Power Supply Calculation Spreadsheet
  - Embedded Ethernet Design:
    - AN292: Embedded Ethernet System Design Guide
    - AN237: TCP/IP Library Programmer's Guide
    - Embedded Ethernet Development Kit User's Guide
    - CP2201 Evaluation Kit User's Guide
    - Ethernet Solutions Guide
- Firmware Image
  - .img format for use with the bootloader
  - .hex format for restoring the device to the original state
- Firmware Source Code
  - PoE Voice Transmission Demo Source Code (source code for C8051F340 MCU)
  - CP220x Flash Programmer Source Code (use for restoring CP220x Flash to original state)
  - 'F34x Ethernet Bootloader Source code (can be used as-is in your embedded system)
  - Initialization routines for the PoE-VOICE-EB (PoE_VOICE_Init.txt).
- PC Demo Application Installer
- Helpful Utilities and Command Line Scripts
  - SET_DHCP and SET_STATIC batch files for quick PC configuration
  - Link to software DHCP server
  - HEX2IMG utility for use with the Ethernet Bootloader

CONTACT INFORMATION

Silicon Laboratories Inc.

4635 Boston Lane
Austin, Texas 78735
Email: MCUinfo@silabs.com
Internet: www.silabs.com

SILICON LABORATORIES