Application Note

AN_433

FT90x USB Audio Device

Version 1.1

Issue Date: 2017-04-04

This application note describes implementing a USB Audio Device on the FT900. The Wolfson/Cirrus WM8731 Digital to Audio Convertor on the MM900EVx modules are used to convert an audio data stream from a host PC to an analog output.

Bridgetek Pte Ltd (BRT Chip)
178 Paya Lebar Road, #07-03, Singapore 409030
Tel: +65 6547 4827 Fax: +65 6841 6071
Web Site: http://www.brtchip.com

Table of Contents

1 Introduction

The FT900/FT901 and FT905/FT906 devices include an I2S interface which is typically used to stream digital audio data to a Digital to Audio Convertor. This application note describes how to bridge this interface to a host PC.

1.1 Overview

The FT90x contains both an I^2S interface and I^2C interface. These can be used to stream digital audio data to a DAC. The audio stream is sent *via* I2S and the DAC is normally controlled *via* I^2C.

The USB device interface may be configured to report a USB Audio Device and receive a suitable audio stream from a host PC. This can be received by the FT900 and subsequently streamed to the DAC.

There is a large choice of suitable DACs available. The MM900EV2A module has a Wolfson/Cirrus WM8731 device on-board which is connected to the I^2S and I^2C busses. The main target for this application note is the WM8731 although sections of code which are specific to this device are clearly identifiable.

Third-party open source code is used to implement a TCP/IP stack in this application note:

- Printf – tinyprintf.

Links to resources for these libraries are in Appendix A – References.

1.2 Scope

The firmware application for the USB Audio Device assumes a fixed audio resolution and sample rate. There is no mechanism for the host PC or user to change the audio streaming properties dynamically.

There is no hardware volume or mute control. Host operating systems must be able to manage the output by manipulating the data in the output stream.

1.2.1 Features

The application note highlights the use of I^2S to stream audio data and I2C for controlling a DAC. Simple buffering is demonstrated but an unbuffered option is shown to provide a low latency data path from USB to the DAC.

1.2.2 Enhancement

Enhancements to this application might include:

- Add support for 24 bit audio and 96 kHz sampling.
- Allow multiple interfaces to be selected by the host PC to change the audio resolution or sampling rate.
- Reduce the overall size of the application code.

This USB Audio Device example application should be treated as an example. Full source code is provided allowing users to build and modify if required:

http://brtchip.com/wp-content/uploads/FT90x/AN_433_FT90x_USB_Audio_Device_V1.1.zip

See section 6 Importing into the FT9xx Toolchain for more information on modifying the project using Eclipse.

2 Project Overview

The project files for the application are divided into the following folders.

Folder	**Description**
Source	Application source code and abstraction files.
lib	Library files.
lib\tinyprintf	tinyprintf library.

Table 2.1 Project Files Overview

The application source code is contained within the "Sources" folder.

In the code, the main() function and the high-level functions of the application are in the "main.c" file.

2.1 Main Program

The server main program is responsible for detecting a connection from the FT900 to the host PC and instantiating a USB Audio Class Device when enumerated by the host. It will also initialise the WM8731 DAC to receive an audio stream in the format which the USB descriptor requested from the host PC.

2.2 I^2S Code

When buffering is enabled the I^2S on the FT900 is used as an interrupt driven interface. There are interrupts when the on-chip buffer is half empty and fully empty. The algorithm used in this application receives and buffers packets of data from the USB interface and writes then in a larger transfer to the DAC. A circular buffer is used for the grouping of data.

If buffering is disabled then data received from the USB interface is written directly to the I2S interface. The interrupt service routine for the I^2S is used to pad any missing data with zero data.

2.3 USB Interface

A handler has been implemented for USB Audio Device class requests. Only class requests used by the Windows USB audio device driver are implemented.

Configuration descriptors are provided for hi-speed and full-speed operation. USB Audio Device configuration descriptors are included. These describe the capabilities and audio formats used by the USB Audio Device and supported by the DAC.

2.4 Power Management Handler

When moving between suspend and resume states on USB, the power management handler is required to notify the USBD (USB device) library that the bus state has changed.

2.5 Other Features

The DFU-C facility can be enabled for this application. This is called from the macro STARTUP_DFU() in the main() function. It will briefly enable the USB device on the FT90x and allow a DFU utility to update the application code. This can be removed entirely or configured to alter the number of milliseconds it will wait before closing the USB device and continuing with the application.

3 Code Structure

When buffering is enabled: After initialization data received from the USB is buffered by the main task code which updates a circular buffer. The I^2S interrupt level code takes data from the circular buffer and sends it to the I^2S interface where it is converted into analog audio.

The streaming data from the host arrives in a frame pattern, which for stereo 16 bit 44100 Hz samples is either 176 bytes or 180 bytes; and for stereo 16 bit 48000 Hz is 196 bytes. This is grouped into a larger packet of 512 bytes to be transmitted to the DAC. A small latency of around 3 ms is therefore gained. There is, however, a 1 kB buffer in the I^2S interface which needs to be fed data to ensure that there are no gaps in the stream. The re-fill is triggered at the half full point of the buffer which allows the I^2S interrupt routine time to send additional data to the back of the buffer while the data at the front is being consumed. At higher data rates the latency will reduce.

If buffering is not enabled: Data received from the USB interface is directly written into the I^2S buffer. If the I^2S buffer is depleted then the I^2S interrupt level code adds a small amount of data to the buffer to prevent the I^2S bus from stopping due to lack of data.

The FT90x API is used to manage the USB interface and to send data *via* the I^2S interface to the DAC. Various other calls are used to supporting interrupts and a UART monitor/debugging interface.

More information on these FT90x API functions can be found in AN_365 FT9xx API Programmers Manual.

4 USB Configuration

4.1 High-speed Configuration Descriptor

The following configuration descriptor is taken from the FT90x device by "USBView" from Microsoft. It is based on the configuration where:

- The DAC is a WM8731.
- The audio resolution is 16 bit.
- The data format is uncompressed PCM.
- The sample rate is 48000 Hz.

```
Device Power State:               PowerDeviceD0

       ---===>Device Information<===---
English product name: "FT900 TypeI PCM 16 bit 48000 Hz"

ConnectionStatus:
Current Config Value:              0x01  -> Device Bus Speed: High
Device Address:                    0x02
Open Pipes:                           0
*!*ERROR:  No open pipes!

          ===>Device Descriptor<===
bLength:                           0x12
bDescriptorType:                   0x01
bcdUSB:                          0x0200
bDeviceClass:                      0x00  -> This is an Interface Class Defined Device
bDeviceSubClass:                   0x00
bDeviceProtocol:                   0x00
bMaxPacketSize0:                   0x40 = (64) Bytes
idVendor:                        0x0403 = Future Technology Devices International Limited
idProduct:                       0x0FD5
bcdDevice:                       0x0101
iManufacturer:                     0x01
     English (United States)  "FTDI"
iProduct:                          0x02
     English (United States)  "FT900 TypeI PCM 16 bit 48000 Hz"
iSerialNumber:                     0x03
     English (United States)  "FT000000"
bNumConfigurations:                0x01

       ---===>Full Configuration Descriptor<===---

          ===>Configuration Descriptor<===
bLength:                           0x09
bDescriptorType:                   0x02
wTotalLength:                    0x006D  -> Validated
bNumInterfaces:                    0x02
bConfigurationValue:               0x01
iConfiguration:                    0x00
bmAttributes:                      0x80  -> Bus Powered
MaxPower:                          0xFA = 500 mA

          ===>Interface Descriptor<===
bLength:                           0x09
bDescriptorType:                   0x04
bInterfaceNumber:                  0x00
bAlternateSetting:                 0x00
bNumEndpoints:                     0x00
bInterfaceClass:                   0x01  -> Audio Interface Class
```

```
bInterfaceSubClass:                0x01  -> Audio Control Interface SubClass
bInterfaceProtocol:                0x00
iInterface:                        0x00

          ===>Audio Control Interface Header Descriptor<===
bLength:                           0x09
bDescriptorType:                   0x24
bDescriptorSubtype:                0x01
bcdADC:                          0x0100
wTotalLength:                    0x001E
bInCollection:                     0x01
baInterfaceNr[1]:                  0x01

          ===>Audio Control Input Terminal Descriptor<===
bLength:                           0x0C
bDescriptorType:                   0x24
bDescriptorSubtype:                0x02
bTerminalID:                       0x01
wTerminalType:                   0x0101 (USB streaming)
bAssocTerminal:                    0x00
bNrChannels:                       0x02
wChannelConfig:                  0x0003
iChannelNames:                     0x00
iTerminal:                         0x00

          ===>Audio Control Output Terminal Descriptor<===
bLength:                           0x09
bDescriptorType:                   0x24
bDescriptorSubtype:                0x03
bTerminalID:                       0x02
wTerminalType:                   0x0301 (Speaker)
bAssocTerminal:                    0x00
bSourceID:                         0x01
iTerminal:                         0x00

          ===>Interface Descriptor<===
bLength:                           0x09
bDescriptorType:                   0x04
bInterfaceNumber:                  0x01
bAlternateSetting:                 0x00
bNumEndpoints:                     0x00
bInterfaceClass:                   0x01  -> Audio Interface Class
bInterfaceSubClass:                0x02  -> Audio Streaming Interface SubClass
bInterfaceProtocol:                0x00
iInterface:                        0x00

          ===>Interface Descriptor<===
bLength:                           0x09
bDescriptorType:                   0x04
bInterfaceNumber:                  0x01
bAlternateSetting:                 0x01
bNumEndpoints:                     0x02
bInterfaceClass:                   0x01  -> Audio Interface Class
bInterfaceSubClass:                0x02  -> Audio Streaming Interface SubClass
bInterfaceProtocol:                0x00
iInterface:                        0x00

          ===>Audio Streaming Class Specific Interface Descriptor<===
bLength:                           0x07
bDescriptorType:                   0x24
bDescriptorSubtype:                0x01
bTerminalLink:                     0x01
bDelay:                            0x01
```

```
wFormatTag:                       0x0001 (PCM)

          ===>Audio Streaming Format Type Descriptor<===
bLength:                            0x0B
bDescriptorType:                    0x24
bDescriptorSubtype:                 0x02
bFormatType:                        0x01
bNrChannels:                        0x02
bSubframeSize:                      0x02
bBitResolution:                     0x10
bSamFreqType:                       0x01
tSamFreq[1]:                      0x00BB80 (48000 Hz)

          ===>Endpoint Descriptor<===
bLength:                            0x09
bDescriptorType:                    0x05
bEndpointAddress:                   0x01  -> Direction: OUT - EndpointID: 1
bmAttributes:                       0x05  -> Isochronous Transfer Type, Synchronization Type
= Asynchronous, Usage Type = Data Endpoint
wMaxPacketSize:                   0x00C4 = 1 transactions per microframe, 0xC4 max bytes
wInterval:                        0x0004
bSyncAddress:                       0x82

          ===>Audio Streaming Class Specific Audio Data Endpoint Descriptor<===
bLength:                            0x07
bDescriptorType:                    0x25
bDescriptorSubtype:                 0x01
bmAttributes:                       0x80
bLockDelayUnits:                    0x00
wLockDelay:                       0x0000

          ===>Endpoint Descriptor<===
bLength:                            0x09
bDescriptorType:                    0x05
bEndpointAddress:                   0x82  -> Direction: IN - EndpointID: 2
bmAttributes:                       0x01  -> Isochronous Transfer Type, Synchronization Type
= No Synchronization, Usage Type = Data Endpoint
wMaxPacketSize:                   0x0004 = 1 transactions per microframe, 0x04 max bytes
wInterval:                        0x0A0A
bSyncAddress:                       0x00
```

4.2 Full-speed Configuration

The full-speed Configuration Descriptor is identical to the high-speed descriptor.

5 Testing

This section describes how to test the audio device on Windows. Two options are available:

- An MM900EV2A module with a headphone jack.

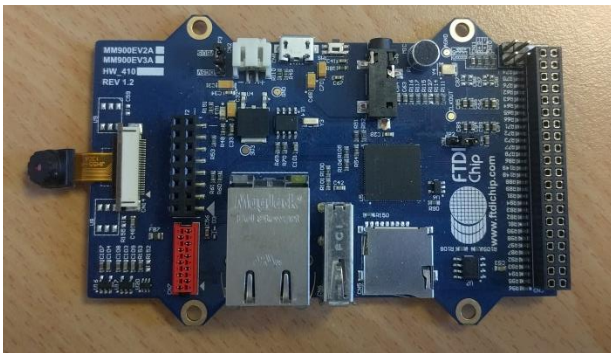

Figure 5.1 MM900EV2A Module

- An MM900EV2A module with speakers connected. JP2 must be set to the ON position and speaker connected to CN10 and CN11 for left and right channels respectively.

Each module was connected to and powered by a PC running Windows 7 or Windows 10.

To program the example code onto the FT900 device on the MM900EV2A board, refer to the application note AN_325 FT9xx Toolchain Installation Guide. The binary file name is "AN_433_FT90x_USB_Audio_Device.bin" and it is found in the "Debug" directory.

5.1 Playback

Once the USB Audio Device has been enumerated by Windows it will then install a device driver automatically. The application is designed to be identical to a standard audio device and use the standard audio device drivers from Windows.

After the device drivers have been installed then the Windows system controls can be used to select the device as the default audio device.

6 Importing into the FT9xx Toolchain

The AN_433 Firmware found at the following link can be easily imported into the FT9xx Toolchain:

http://brtchip.com/wp-content/uploads/FT90x/AN_433_FT90x_USB_Audio_Device_V1.1.zip

Once installed, select File --> Import --> General --> Existing Projects into Eclipse, and point to the downloaded and extracted project directory.

The project will appear in Eclipse Project Explorer as shown in Figure 6.1.

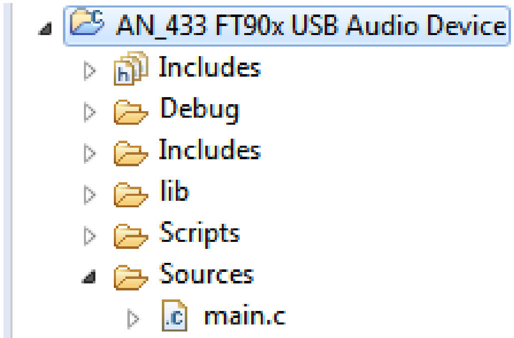

Figure 6.1 Eclipse Project Structure

6.1 Changing the Application Software

The application software provided can be altered and changed if required. The FT9xx Toolchain is a free tool to enable code development and debug for the FT90x series and is based on plug-ins for the free popular IDE using the GCC compiler.

With each software change, the project should be rebuilt and reprogrammed into the FT90x IC. Please refer to AN_325 FT9xx Toolchain Installation Guide for further information.

7 Contact Information

Head Quarters – Singapore

Bridgetek Pte Ltd
178 Paya Lebar Road, #07-03
Singapore 409030
Tel: +65 6547 4827
Fax: +65 6841 6071

E-mail (Sales) sales.apac@brtchip.com
E-mail (Support) support.apac@brtchip.com

Branch Office – Taipei, Taiwan

Bridgetek Pte Ltd, Taiwan Branch
2 Floor, No. 516, Sec. 1, Nei Hu Road, Nei Hu District
Taipei 114
Taiwan , R.O.C.
Tel: +886 (2) 8797 5691
Fax: +886 (2) 8751 9737

E-mail (Sales) sales.apac@brtchip.com
E-mail (Support) support.apac@brtchip.com

Branch Office - Glasgow, United Kingdom

Bridgetek Pte. Ltd.
Unit 1, 2 Seaward Place, Centurion Business Park
Glasgow G41 1HH
United Kingdom
Tel: +44 (0) 141 429 2777
Fax: +44 (0) 141 429 2758

E-mail (Sales) sales.emea@brtichip.com
E-mail (Support) support.emea@brtchip.com

Branch Office – Vietnam

Bridgetek VietNam Company Limited
Lutaco Tower Building, 5th Floor, 173A Nguyen Van Troi,
Ward 11, Phu Nhuan District,
Ho Chi Minh City, Vietnam
Tel : 08 38453222
Fax : 08 38455222

E-mail (Sales) sales.apac@brtchip.com
E-mail (Support) support.apac@brtchip.com

Web Site

http://brtchip.com/

Distributor and Sales Representatives

Please visit the Sales Network page of the Bridgetek Web site for the contact details of our distributor(s) and sales representative(s) in your country.

Appendix A – References

Document References

FT90x Datasheet

tinyprintf http://www.sparetimelabs.com/tinyprintf/tinyprintf.php

AN_325 FT9xx Toolchain Installation Guide

AN_365 FT9xx API Programmers Manual

DFU AN_380 FT900 Bootloader DFU Usage

FT9xx Toolchain

http://brtchip.com/wp-content/uploads/FT90x/AN_433_FT90x_USB_Audio_Device_V1.1.zip

24AA02E48T-I/OT EEPROM http://www.microchip.com/wwwproducts/en/24AA02E48

MM900EVxA http://brtchip.com/ft9xx-development-modules-landing/

Acronyms and Abbreviations

Terms	Description
DFU	Device Firmware Update
EEPROM	Electrically Erasable Programmable Read Only Memory
I^2C	Inter-Integrated Circuit
I^2S	Integrated Inter-IC Sound Bus
USB	Universal Serial Bus
UART	Universal Asynchronous Receiver and Transmitter (Serial Port)

Appendix B – List of Tables & Figures

List of Tables

List of Figures

Appendix C – Revision History

Document Title: AN_433 FT90x USB Audio Device

Document Reference No.: BRT_000074

Clearance No.: BRT#055

Product Page: http://brtchip.com/m-ft9/

Document Feedback: Send Feedback

Revision	Changes	Date
1.0	Initial Release	2017-01-23
1.1	Updated Release adding no-buffered mode.	2017-04-04