Application Note

AN_426

FT90x USB Examples using MCCI Data Pump USB Stack

Version 1.0

Issue Date: 2017-09-13

This application note demonstrates the usage of MCCI USBD examples ported to FT90x.

Bridgetek Pte Ltd (BRTChip)
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

MCCI's Data Pump USB stack has been ported onto the FT90x and the demo examples corresponding to this application note illustrates the common USB class devices using the Stack. This provides an alternative to the default Bridgetek USB stack that is shipped with the FT90x Toolchain. Users those who wish to use the MCCI USB stack, can base their products on the examples in this section. The data pump is available as a **static library** within the examples. Please contact Bridgetek if access to the source code of the Data Pump library is required.

1.1 Overview

The MCCI Data Pump is a comprehensive USB framework that attempts to encapsulate USB hardware, protocol and class level details, allowing the application developer to concentrate more on application programming than having to understand USB details. The framework supports both USB devices and hosts and a large variety of USB classes. The examples in this document cover the common USB device applications.

1.2 Scope

This application note demonstrates the USB device examples utilizing the MCCI Data Pump stack. It explains the project configurations and settings required to build the source code and how to get the demo applications up and running quickly. For more details on the MCCI Data Pump framework and individual USB class specific APIs, please refer to other relevant application notes.

The following application notes from Bridgetek can provide further information:

Related Application Notes	**Description**
AN_400_MCCI_USB_Resource_Compiler_UserGuide	Describes the syntax of the USB Resource Compiler utility that is responsible for generating the USB Descriptors from a text .urc file
AN_402_MCCI_USB_DataPump_UserGuide	Detailed explanations of the Data Pump API and architecture
AN_403_MCCI_USB_DataPump_Mass_Storage_Protocol_UserGuide	Data Pump API specific to the USB Mass Storage Class
AN_406_MCCI-USB-DataPump-Virtual-Ethernet-Protocol-UserGuide	Data Pump API specific to the USB RNDIS

Table 1 MCCI Data Pump Application Notes

2 Prerequisites

2.1 Required Hardware

	Item	Description
1.	MM900EVxA	FT900 MCU Evaluation Module, x=1,2, or 3
2.	UMFTPD2A	MM900EV MCU Programmer / Debugger Module
3.	Micro USB Cables	1 x micro USB cable to interface the MM900EV board to the host PC 1 x micro USB cable to interface the UMFTPD2A to the host PC

Table 2 Hardware Required

2.2 Required Tools and software

	Item	Description
1.	FT90x Toolchain	Utilities required for the MCCI build are available only with FT90x Toolchain v2.2.0 or above.
2.	MCCI Example Projects	The source code corresponding to this application note is available for download on the Bridgetek website.
3.	Terminal Application	Serial Port (COM) terminal emulation program. For example, Br@y's Terminal, HyperTerminal, PuTTY, Tera Term.

Table 3 Tools Required

2.3 Build process and source code structure

The MCCI examples have a different build process and code structure compared to other Bridgetek examples shipped along with the FT90x Toolchain. This section will provide a brief overview of these differences, for further details please refer to the application notes listed in Table 1.

The MCCI build depends on a few external utilities like the USB resource compiler, bsdmake and some standard UNIX coreutils. It also requires a shell environment to execute the build. These utilities and shell are provided as part of the FT90xToolchain installation and are available in the installation directory in the `FT90x Toolchain\MCCI\` directory, if selected as part of the installation options. Note that these tools are only available with the FT90x Toolchain **v2.2.0** and above.

2.3.1 Global Environment Variables

The MCCI build depends on a few environment variables that have been pre-configured in the example projects. The configuration can be viewed in the project settings in Eclipse (`Project | Properties | C/C++ Build | Environment`). When creating a new MCCI project, these variables must be imported. The simplest way is to copy the environment settings from an existing example.

While modifying or adding new paths, avoid spaces and other special characters as much as possible. They are described in Table 4 and shown in Figure 1.

Shell Variable	Description	Default Value
BUILD_TYPE	Variable to help the automated build to choose between Release and Debug configurations	RELEASE or DEBUG depending on the project configuration selected
LD_LIBC_DIR	Path to the standard C library	${FT90X_TOOLCHAIN}/tools/ft32-elf/lib
LD_LIBFT900_DIR	Path to the FT90x peripheral driver library	${FT90X_TOOLCHAIN}/hardware/lib
LD_LIBGCC_DIR	Path to the GCC library	${FT90X_TOOLCHAIN}/tools/lib/gcc/ft32-elf/
MAKE	Path to BSDMAKE executable (available at ${MCCI_TOOLS}/bin)	bsdmake.exe
MAKESHELL	Path to the SHELL to be used for the build (available at ${FT90X_TOOLCHAIN}/../MCCI/msys32/usr/bin)	bash.exe
MCCIUSBRC_INC	Path to USBRC Includes	${MCCI_TOOLS}/i/usbrc
MCCI_TOOLS	Path to MCCI proprietary tools	${FT90X_TOOLCHAIN}/../MCCI/tools/tools
PATH	System PATH to be used for the build	${FT90X_TOOLCHAIN}/tools/bin; ${FT90X_TOOLCHAIN}/../MCCI/msys32/usr/bin; ${MCCI_TOOLS}/bin;
TARGET_BIN_NAME	Name for the output binary file	Same as the project name - ${ProjName}
TMP	Path to a temporary working directory for the SHELL	Same as system %TMP% directory - ${TMP}
XARGS	Path to XARGS executable	"${MCCI_TOOLS}/bin/xargs.exe"

Table 4 MCCI Projects - Environment Variables

Note: The environment variables are configured to **replace the native environment variables**; this is to ensure that the build is not affected by any pre-existing configurations in the user's PC for variables such as PATH.

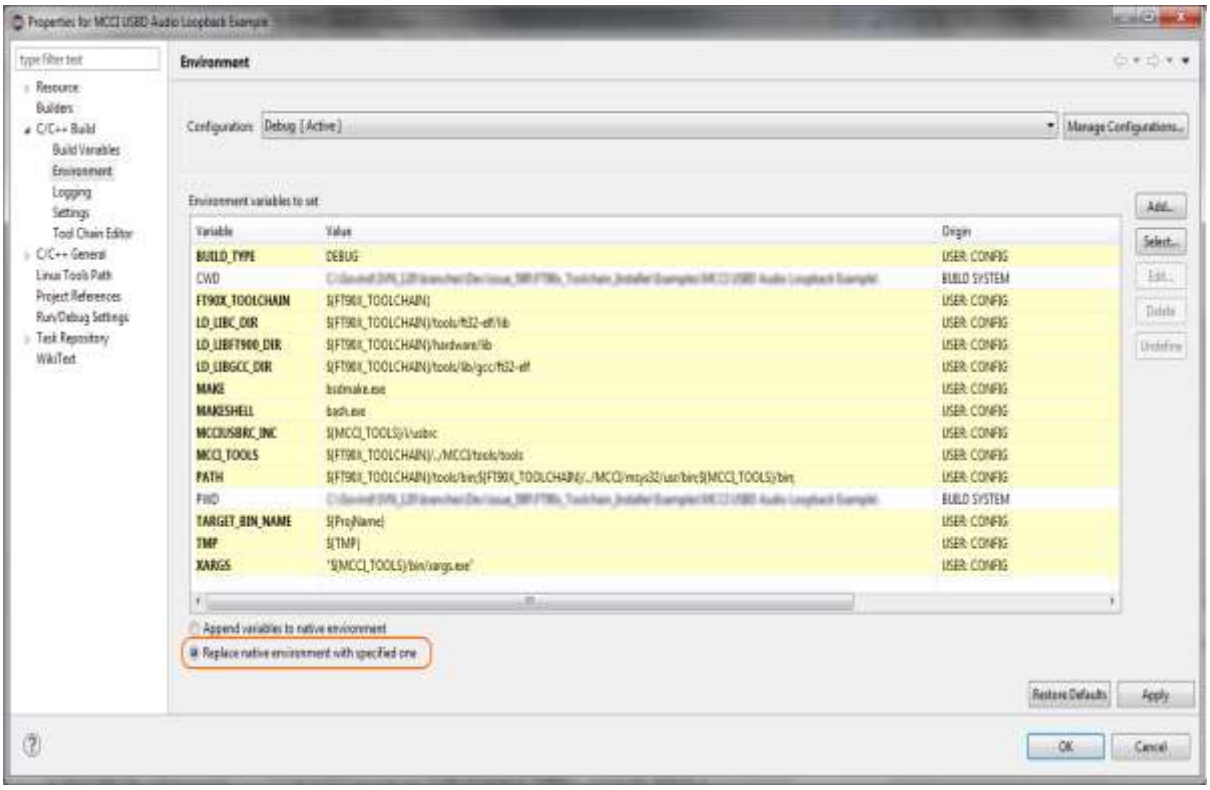

Figure 1 MCCI Projects - Environment Variables

2.3.2 Build Process

The MCCI examples are all *custom makefile* projects within Eclipse. Any changes to the build process (adding new .c files) or changing compiler options must be done by editing the appropriate makefiles in the project and the convenient Eclipse Bridgetek plugin features are also not available. However, Eclipse can still be used to edit, build, and debug the projects and all the CDT extensions for source code navigation and analysis can be used.

The build is controlled by a series of shell scripts and makefiles. The top most makefile is located in the project root directory. This invokes a script `mcci-gen-make.sh` which runs the script `/usbkern/makebuildtree` (generated from `/usbkern/makebuildtree.sh`). This generates a `/usbkern/build` folder with all the dependencies, sources and makefiles necessary for the release or debug build. In order to force a rebuild of the entire source tree (necessary if moving a previously built project to another directory), the user can manually delete the `/usbkern/build` directory along with the `/usbkern/makebuildtree` file. In most situations, issuing a build and/or clean followed by build from the Eclipse GUI should suffice.

2.3.3 Debug and Release Builds

Each MCCI example project can be built with either of two configuration options - Debug or Release. The default setting is a Debug build. The difference between them is that the Debug build has UART logs enabled while the Release build has logs disabled. The UART output is over **UART0** with the baud rate configured to be **115200bps**.

To switch the configuration right click the project in Eclipse and go to `Build Configurations | Set Active | Debug / Release`. Output binary for the Debug and Release builds are located in the path `usbkern/build/ft32/mm900beta/none-gcc-checked` and `usbkern/build/ft32/mm900beta/none-gcc-free` respectively.

2.3.4 Folder Structure

The top level folder is /usbkern, inside of which there are various folders organized functionally. The important folders and files are summarized in Table 5.

Path	Description
usbkern\arch\ft32\os\none\soc\ft900\app\ft900dci_xxx	A large part of the application specific source code is available here. Including initialization sequence customizations and any callbacks registered with the Data Pump layer.
usbkern\arch\ft32\os\none\soc\ft900\app\ft900dci_xxx*.urc	The *.urc file contains the **configurations for USB descriptors** including VID/PID etc. Refer AN_400_MCCI_USB_Resource_Compiler_UserGuide for information on the syntax for defining USB descriptors. To change these configurations, edit this file and rebuild the project.
\usbkern\libport\arch\ft32\mk\gcc\sys.mk	Configurations for compiler/linker options, global settings for the whole project
usbkern\mk\libdesc.mk	Libraries used are specified here
UsbMakefile.inc	This makefile defines the source code files that are to be compiled in each directory. To add a new source file, the UsbMakefile.inc in the appropriate path should be modified.

Table 5 MCCI Projects - paths of important files

2.4 Importing examples to Eclipse

1. Install the FT90x Toolchain v2.2.0 or above.

2. Extract or copy the source code related to this application note to a convenient location. Please make sure the destination directory depth is not very deep, as the MCCI tools are unable to work with long path names (200 characters or more).

3. Launch Eclipse and import the MCCI projects into the current workspace by going to `File | Import | General | Existing Projects into Workspace` as shown in Figure 2

4. Browse and select the root folder of the examples as shown in Figure 3. This will import all the MCCI projects into the workspace.

5. Now the examples can be built and programmed into the FT900 IC. See AN_325 FT900 Toolchain Installation Guide for more information.

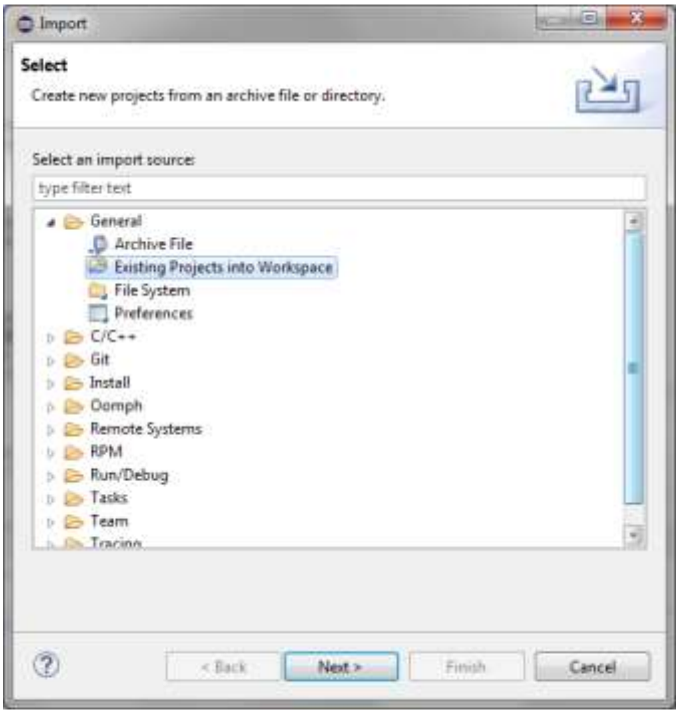

Figure 2 Import Existing Projects to Workspace

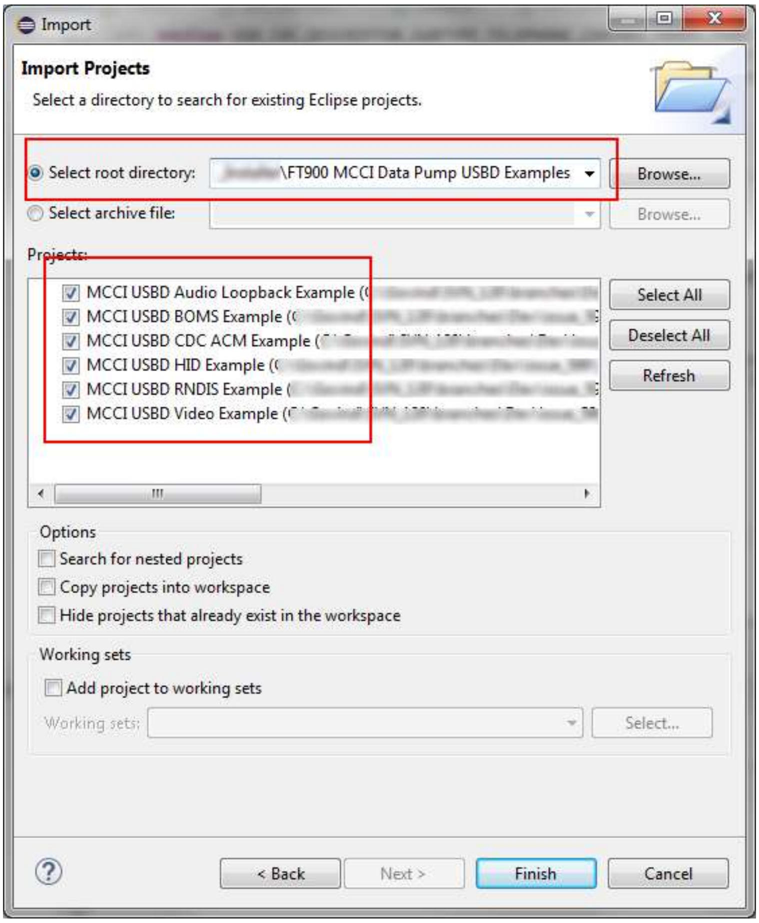

Figure 3 Import all MCCI projects

2.5 MCCI USB Device Examples

2.5.1 MCCI USBD Audio Loopback Example

This example creates a USB Audio device with a virtual USB Speaker and Microphone.

2.5.1.1 Purpose

The example code will enumerate as a USB Audio Class device on the PC with a speaker and microphone. Data is internally looped back from the speaker interface to the microphone interface for illustration.

2.5.1.2 Setup

Compile the example source code and program the "checked" binary (with UART logs enabled) from the `usbkern/build/ft32/mm900beta/none-gcc-checked/` folder onto the FT90x EVM. Connect the EVM to a PC via USB and monitor the UART0 output on a serial port monitor. The "free" binary can be used if UART logs are not desired.

To test the loopback device, an audio playback application can be used to play audio over the USB Speaker device while a recording program such as Windows Sound Recorder can be used to record audio received over the USB Microphone.

2.5.1.3 Execution

1. A welcome message should appear like so (followed by other Data Pump related messages):

```
(C) Copyright Bridgetek Pte Ltd
--------------------------------------------------------------------
Welcome to MCCI USBD Audio Loopback Example...

FT900 will enumerate as an Audio Class Device with both Microphone and Speaker
Any data sent over the USB Speaker interface will be looped back over the USB
Microphone interface.
-------------------------------------------------------------------
```

2. Navigate to `Control Panel | Hardware and Sound | Manage Audio Devices`. MCCI(r) Audio Demo Loopback device should be visible in both the Playback and Recording tabs as shown in Figure 4 and Figure 5.

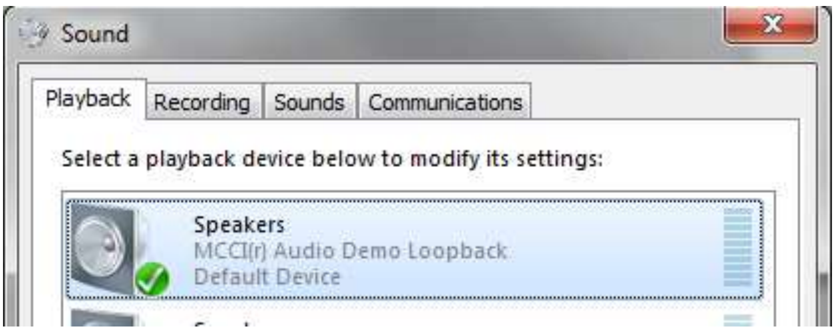

Figure 4 Audio Demo – Speaker

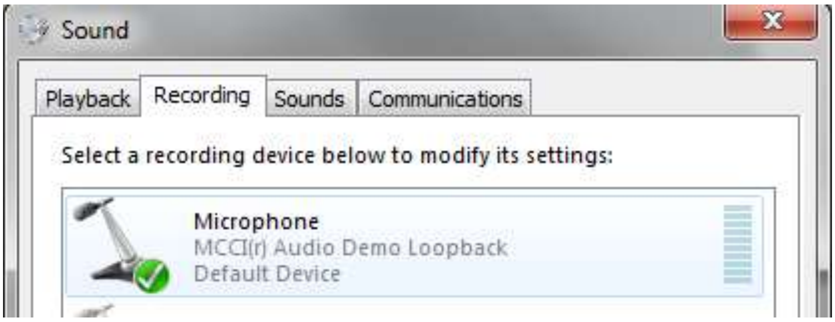

Figure 5 Audio Demo - Microphone

3. In the Audio Devices panel, set the loopback device as the default speaker and default microphone (right click and select `Set as Default Device`)
4. Play some audio in a multimedia application such as Windows Media Player. While audio is being played run the Sound Recorder application (`All Programs | Accessories | Sound Recorder`) and click on the Start Recording button.
5. Stop recording after a while. Disconnect the USB device or change the default speaker setting to the system speaker. Playback the recorded sound clip. The recorded clip should contain the audio that was played in step 4.

Figure 6 Audio demo - Recording from the loopback device

2.5.2 MCCI USBD BOMS Example

This example creates a USB Mass Storage Class Device using Bulk Only transport.

2.5.2.1 Purpose

The example code will enumerate as an USB Mass Storage Class Device on a PC and the user can perform various file operations on an SD card mounted on the FT90x EVM.

2.5.2.2 Setup

Compile the example source code and program the "checked" binary (with UART logs enabled) from the `usbkern/build/ft32/mm900beta/none-gcc-checked/` folder onto the FT90x EVM. Connect the EVM to a PC via USB and monitor the UART0 output on a serial port monitor. The "free" binary can be used if UART logs are not desired.

An SD card should be mounted on the FT90x EVM.

2.5.2.3 Execution

1. A welcome message should appear like so (followed by other Data Pump related messages):

```
(C) Copyright Bridgetek Pte Ltd
--------------------------------------------------------------------
Welcome to MCCI USBD BOMS Example...

FT900 will enumerate as BOMS Mass Storage Device.
On-board SD Card (on FT90x EVM) can be accessed from a PC.
--------------------------------------------------------------------
```

2. Open a Windows Explorer window and navigate to "My Computer". A removable disk corresponding to the SD Card connected to FT90x EVM should be visible as shown in Figure 7.

Figure 7 Mass Storage Device Demo - Removable Disk

3. The drive can be accessed and normal file operations (create file, read, write etc.) can be done on it.

2.5.3 MCCI USBD CDC ACM Example

This example emulates a Communications Device Class (CDC) Abstract Control Model (ACM) device.

2.5.3.1 Purpose

When connected to a host, the operating system on the host can open a Virtual COM Port (VCP) to the CDC ACM device. Data sent from the Host over this VCP is echoed back to the Host via the same VCP.

2.5.3.2 Setup

Compile the example source code and program the "checked" binary (with UART logs enabled) from the `usbkern/build/ft32/mm900beta/none-gcc-checked/` folder onto the FT90x EVM. Connect the EVM to a PC via USB and monitor the UART0 output on a serial port monitor. The "free" binary can be used if UART logs are not desired.

2.5.3.3 Execution

1. A welcome message should appear like so (followed by other Data Pump related messages):

```
(C) Copyright Bridgetek Pte Ltd
--------------------------------------------------------------------
Welcome to MCCI USBD CDC ACM Example...

Connect a terminal application to the Virtual COM Port enumerated.
Data typed into the terminal will be echoed back.
--------------------------------------------------------------------
```

2. A new VCP corresponding to the CDC-ACM device should be enumerated in Windows. In case Windows is unable to install the driver for this device the user must install the .INF file included with this example.

3. Open a new terminal to this VCP and type a message (e.g. "Hello World"). The entered string will be echoed back to the terminal.

2.5.4 MCCI USBD HID Example

This example creates a USB Human Interface device.

2.5.4.1 Purpose

The example code will enumerate as a USB Human Interface Device (HID) on a Host PC.

2.5.4.2 Setup

Compile the example source code and program the "checked" binary (with UART logs enabled) from the `usbkern/build/ft32/mm900beta/none-gcc-checked/` folder onto the FT90x EVM. Connect the EVM to a PC via USB and monitor the UART0 output on a serial port monitor. The "free" binary can be used if UART messages are not desired.

2.5.4.3 Execution

1. A welcome message should appear like so (followed by other Data Pump related messages):

```
(C) Copyright Bridgetek Pte Ltd
--------------------------------------------------------------------
Welcome to MCCI USBD HID Example...

Device will enumerate as an USB HID Class
--------------------------------------------------------------------
```

2. A new USB Input Device shall be visible in Windows device manager.

3. Alternatively software such as [Microsoft USBView]() can be used to view the descriptors of the attached device. This is shown in Figure 8 and Figure 9.

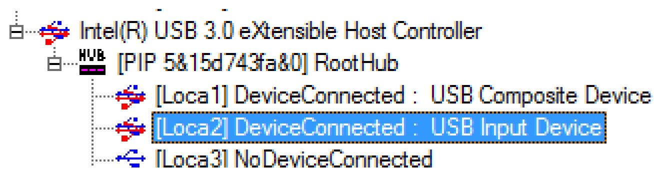

Figure 8 HID Demo – USB View - Left Pane

```
Device Descriptor:
bcdUSB:             0x0200
bDeviceClass:         0x00
bDeviceSubClass:      0x00
bDeviceProtocol:      0x00
bMaxPacketSize0:      0x40 (64)
idVendor:           0x040E (MCCI)
idProduct:          0xF113
bcdDevice:          0x0100
iManufacturer:        0x01
0x0409: "MCCI Corporation"
iProduct:             0x02
0x0409: "MCCI(r) Generic HID demo"
iSerialNumber:        0x03
0x0409: "DUMMY"
bNumConfigurations:   0x01

ConnectionStatus: DeviceConnected
Current Config Value: 0x01
Device Bus Speed:     High
Device Address:       0x1A
Open Pipes:              1

Endpoint Descriptor:
bEndpointAddress:     0x81  IN
Transfer Type:   Interrupt
wMaxPacketSize:     0x0040 (64)
bInterval:            0x08

Configuration Descriptor:
wTotalLength:       0x0022
bNumInterfaces:       0x01
bConfigurationValue:  0x01
iConfiguration:       0x00
bmAttributes:         0xE0 (Bus Powered Self Powered Remote Wakeup)
MaxPower:             0x01 (2 Ma)

Interface Descriptor:
bInterfaceNumber:     0x00
bAlternateSetting:    0x00
bNumEndpoints:        0x01
bInterfaceClass:      0x03 (HID)
bInterfaceSubClass:   0x00
bInterfaceProtocol:   0x00
iInterface:           0x00

HID Descriptor:
bcdHID:             0x0101
bCountryCode:         0x00
bNumDescriptors:      0x01
bDescriptorType:      0x22
wDescriptorLength:  0x002F

Endpoint Descriptor:
bEndpointAddress:     0x81  IN
Transfer Type:   Interrupt
wMaxPacketSize:     0x0040 (64)
bInterval:            0x08
```

Figure 9 HID Demo – USB View - Right Pane

2.5.5 MCCI USBD RNDIS Example

This example creates a Remote Network Driver Interface Specification (RNDIS) compliant device when connected to a USB host. The FT900 device is the network device that provides network connectivity to the host PC over USB.

A USB RNDIS device is implemented as a USB Communication Device Class (CDC) device with two interfaces. A Communication Class interface, of type Abstract Control, and a Data Class interface combined to form a single functional unit representing the USB Remote NDIS device. The Communication Class interface includes a single endpoint for event notification and uses the shared bidirectional Control endpoint for control messages. The Data Class interface includes two bulk endpoints for data traffic.

2.5.5.1 Purpose

The example illustrates the RNDIS function by enabling the FT900 to appear as a USB-to-Ethernet bridge.

2.5.5.2 Setup

Compile the example source code and program the "checked" binary (with UART logs enabled) from the `usbkern/build/ft32/mm900beta/none-gcc-checked/` folder onto the FT90x EVM. Connect the EVM to a PC via USB and monitor the UART0 output on a serial port monitor. The "free" binary can be used if UART logs are not desired.

An Ethernet cable should be connected to the FT90x EVM. The cable should be connected to a LAN or Internet via a router (for testing internet access).

2.5.5.3 Execution

1. A welcome message should appear like so (followed by other Data Pump related messages):

```
(C) Copyright Bridgetek Pte Ltd
--------------------------------------------------------------------
Welcome to MCCI USBD RNDIS Example...

Device will behave as a USB-to-Ethernet bridge.
--------------------------------------------------------------------
```

2. After enumeration on the Host PC, the RNDIS device should appear as a new Network Adapter in Windows device manager.

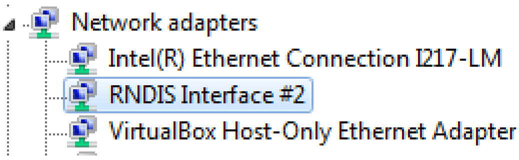

Figure 10 RNDIS Demo - Device Manager

3. The host PC should be able to browse the internet if the Ethernet cable on the FT900 EVM is connected to the internet. To test, open a browser and navigate to http://www.brtchip.com/.

2.5.6 MCCI USBD Video Example

This example creates a simple USB Video Class (UVC) device.

2.5.6.1 Purpose

The FT900 will enumerate as a UVC device and a dummy image/video can be viewed from the Host PC. The image is sent in MJPEG format.

2.5.6.2 Setup

Compile the example source code and program the "checked" binary (with UART logs enabled) from the `usbkern/build/ft32/mm900beta/none-gcc-checked/` folder onto the FT90x EVM. Connect the EVM to a PC via USB and monitor the UART0 output on a serial port monitor. The "free" binary can be used if UART logs are not desired.

2.5.6.3 Execution

1. A welcome message should appear like so (followed by other Data Pump related messages):

```
(C) Copyright Bridgetek Pte Ltd
--------------------------------------------------------------------
Welcome to MCCI USBD Video Example...

FT90x enumerates as an USB Video Class (UVC) Device.
--------------------------------------------------------------------
```

2. After enumeration on the Host PC, the UVC device will appear as a USB Composite device on the Host PC.

3. Any PC application that accepts video input like a MyCam application, VLC player or Skype can be used to view the default video data. The video capture shows the Bridgetek CleO board logo (Figure 11) and a still image capture will show the FT900 logo (Figure 12). Both images are 160x120 resolution JPG files stored in the file `usbkern\app\videodemo\videodemo_mjpegimage.c`

Figure 11 Default video image - *CleO* Logo

Figure 12 Default still image - FT900 log

3 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

Document References

FT900 Product Page: http://brtchip.com/mcu/

MCCI Data Pump

AN_324 FT900 User Manual

USB Virtual COM Port Tutorial

USB Class Specifications

AN_400 MCCI USB Resource Compiler UserGuide

AN_402 MCCI USB DataPump UserGuide

AN_403 MCCI USB DataPump Mass Storage Protocol UserGuide

AN_406 MCCI USB DataPump Virtual Ethernet Protocol UserGuide

FT90x Development Modules: MM900EVxA and UMFTPD2A

FT90x Toolchain

AN_325 FT900 Toolchain Installation Guide

Microsoft USBView

MCCI Example Projects: http://brtchip.com/ft90x/

Acronyms and Abbreviations

Terms	Description
ACM	Abstract Control Model
BOMS	Bulk Only Mass Storage
CDC	Communications Device Class
EVM	Evaluation Module
HID	Human Interface Device
RNDIS	Remote Network Driver Interface Specification
USB	Universal Serial Bus
USB-IF	USB Implementers Forum
UVC	USB Video Class
VCP	Virtual COM Port

Appendix B – List of Tables & Figures

List of Figures

List of Tables

Appendix C – Revision History

Document Title: AN_426 FT90x USB Examples using MCCI Data Pump USB Stack

Document Reference No.: BRT_000125

Clearance No.: BRT#097

Product Page: http://brtchip.com/product/

Document Feedback: Send Feedback

Revision	Changes	Date
1.0	Initial Release	2017-09-13