eMDee Technology, Inc.

User's Manual

USB
Simulated FFCS
Communications Interface Box

Document No: **MAN10906**
Revision: **C**
Date: **22JUL14**

eMDee Technology, Inc.
www.emdee.com

Table of Contents

1. Overview

The Simulated FFCS Interface Box is a USB peripheral that provides audio and digital I/O interfaces for simulation of a tactical VIC-3 communications system.

The box provides interface through a standard USB 2.0 interface for I/O (switches, buttons, etc.) and audio input and output. By itself, the device is simply an interface box and will require special software to allow the box to function properly. A description of the software is provided in accompanying user's manuals.

The device takes its +5VDC power from the USB bus and can be operated with only the USB cable connected. Some tactical headsets require a higher voltage to power the microphone and require an external power supply to be connected to the interface box (see Section 2.1). In these cases, the interface box is still operating from the USB power supply and the external supply is only used for the microphone.

2. External Connectors

Figure 2.1 shows the locations of the external connections to the interface box:

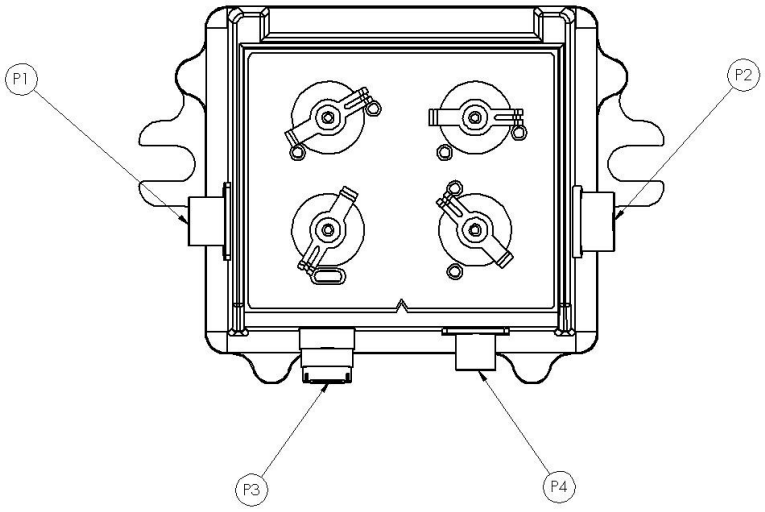

Figure 2.1: FFCS External Connectors

2.1 P1 - Microphone Power

When using a headset that requires power for the microphone, it will be necessary to supply external power in the range of +6VDC to +30VDC. The power supply should provide a minimum of 500mA.

The microphone power is applied through P1 and is internally routed to the correct pin on the headset connector(s). The pinout of the external power connector is:

PIN	***SIGNAL***
A	6 – 30VDC
B	n/c
C	n/c
D	Common
E	n/c
F	n/c

The mating connector to P1 is part number PT06A10-6P manufactured by Amphenol. We also provide a power cable assembly (eMDee Technology part number 10910).

2.2 P2 – USB

The USB connector is a mini USB B-Type connector with a locking shroud. A standard mini Type-B USB cable can be used without a locking shroud, but it is recommended to use a cable with a locking shroud. The following cables are compatible:

Length	***eMDee Part Number***	***Bulgin Part Number***
2 meters	10957	PX0441/2M00
3 meters	10765	PX0441/3M00
4.5 meters	10534	PX0441/4M50

2.3 P3 – Headset

The headset connector P3 is a standard U-283/U connector with pin-out that matches that from tactical equipment. It is compatible with a number of different types of tactical headsets/handsets (see Section 4.). The pin-out for this connector is:

PIN	*SIGNAL*
A	Common
B	Headphone Audio
C	PTT
D	Mic Signal
E	Mic Common
F	Power

2.4 P4 – External PTT

This connector is inactive and is not internally connected.

3. DIP Switch Settings

The DIP switch on the back of the interface box is used to configure the USB interface board and internal microphone bias. The table below details the function of each of the DIP switches:

Switch #	*Function*
1 2 3 4	The combination of these four switches creates a binary equivalent for the identifier of the interface box. Switch 1 is the LSB and switch 4 is the MSB (binary number is read backwards)
5	Not Used
6	Not Used
7	Activate Internal Microphone Bias for left channel (ON)
8	Activate Internal Microphone Bias for right channel (not used for this device)

3.1 USB Device Identifier

More than one interface box can be used in system simultaneously. When using more than one of the same type of USB interface board, the software will need to know the difference between the boards. This is accomplished using the first 4 switches on the DIP switch that is accessible on the back of the interface box. The software driver that is loaded will reflect the identifier that is configured with these switches. Here are two examples:

Example 1: The first 4 switches on the DIP switch are set to OFF. This corresponds to an identifier of zero. Therefore, the driver that is loaded for this device will be labeled “eMDee USB Composite Device 0”.

Example 2: The first 4 switches on the DIP switch are set to ON, ON, OFF, OFF (as seen from left to right). This corresponds to an identifier of three. Therefore, the driver that is loaded for this device will be labeled “eMDee USB Composite Device 3”.

3.2 Internal Microphone Bias Voltage for PC-Style Headsets

Switch 7 and 8 on the DIP switch are used to activate an internal microphone bias voltage that is applied to pin D of the headset connector(s). Switch 7 activates the bias voltage on the left side and switch 8 activates it on the right side. The internal microphone bias voltage must only be activated (ON) when using a standard PC-style headset and compatible extension cable. Do not use the internal bias voltage for a CVC or other tactical headset or handset.

4. Compatible Headsets

This interface box is compatible with many different kinds of tactical audio equipment plus a standard PC-style headset. The box is designed to connect directly to tactical equipment but does require an adapter cable to be used with a PC-style headset (also refer to section 3.2 on how to configure the DIP switches for using a PC headset). These adapter cables are available:

- eMDee PN 10909: Female Adapter Cable with 1/8” stereo phone jacks
- eMDee PN 10914: Male Adapter Cable with 1/8” stereo phone plugs

The following table lists several pieces of tactical audio equipment that is compatible with this interface box (this list is not comprehensive but lists the headsets and handsets that have been tested with this interface box).

Equipment	***Type***
CVC helmet with integrated headset	Powered microphone
Bose Tri-Port noise canceling headset	Powered microphone
H-250 handset	Non-powered microphone

5. Software Interface

5.1 USB Interface API

This product uses a USB CODEC that is common among many of our USB-based products. We have developed an API interface that allows a programmer to gain access to the functionality of the audio CODEC itself and the extended functionality provided by this specific product.

The documentation for this API is provided on-line from our website. To access the documentation, go to our website www.emdee.com and click on Support. On the support page you will see a link labeled "USBI2C API Reference". From this link you will be able to download the latest version of the API and view the documentation for each of the functions that are available.

The following sections describe how to use this API to interface to the FFCS Interface Box.

5.2 Basic Initialization

To properly initialize the USB board for this configuration and prepare it for use, follow these steps using the USBI2C API (refer to Section 5.1 for details on the API documentation).

1) Call **UsbI2cOpen** with *devAddress* equal to the setting of the DIP switch on the back of the interface box (refer to Section 3 for details on the DIP switch). DIP switches 1 through 4 set up a binary number from 0 to 15 that is used to identify this interface box.
2) Set the CODEC input select modes to "Line-In" by setting XDFP register 1 bits 5 & 6 to zero. To prevent corrupting the default settings of the register, first, call **UsbGetXDFP** with *nXDFPAddr* = 1. Take the return value of that call and AND it with FFFFFF9F to zero out bits 5 and 6. Use the result of the AND operation to set the new value of the XDFP register with **UsbSetXDFP**.
3) Set the Microphone Bias by setting XDFP register 1 bit 9 to one. To prevent corrupting the default settings of the register, first, call **UsbGetXDFP** with nXDFPAddr = 1. Take the return value of that call and OR it with 200h to set bit 9. Use the result of the OR operation to set the new value of the XDFP register with **UsbSetXDFP**.
4) Set the Sref by setting XDFP register 45147 bit 0 to one. To prevent corrupting the default settings of the register, first, call **UsbGetXDFP** with nXDFPAddr = 41345 (This is the read address for the register). Take the return value of that call and OR it with 1 to set bit 0. Use the result of the OR operation to set the new value of the XDFP register with nXDFPAddr=45147 by calling **UsbSetXDFP**.
5) Call **UsbI2cWrite** with *address* = 20h and *data* = [C9h, C0h]. This sets the input gain to the lowest level, deactivates the mixer control, and sets bit 6 of byte 1 to input mode.
6) Call **UsbI2cWrite** with *address* = 22h and *data* = [FFh, FFh]. This puts byte 0 and byte 1 in input mode so that subsequent calls to **UsbI2Read** at address 22h will retrieve the settings of those inputs.
7) Call **UsbI2cWrite** with *address* = 23h and *data* = [FFh, FFh]. This puts byte 0 and byte 1 in input mode so that subsequent calls to **UsbI2Read** at address 23h will retrieve the settings of those inputs.
8) Call **UsbI2cWrite** with *address* = 48h and *data* = [05h]. This enables and configures the on-board A/D converter.

Once initialized, the FFCS Interface Box is ready for use. Refer to the tables in Section 5.3 for interfacing to the box during run-time. You can record and playback to the CODEC device just as you would any standard Windows audio device.

Before exiting the application, call **UsbI2cClose** with the handle to the open device to release the driver.

5.3 Hardware I/O Map

The following tables detail the digital I/O mapping of the FFCS interface box. Use **UsbI2cRead** and **UsbI2cWrite** to retrieve and set the I/O during run-time.

5.3.1 I2C Address 20h

In this configuration, bit 6 of byte 1 is connected to a discrete input so it will be necessary to set that bit to input mode. Do this by writing a 1 to that bit at initialization and any other time you write to this address. The rest of the bits are discrete outputs used for microphone volume control and routing control of the internal mixer.

I2C Address 20h		
Byte	**Bit**	**Function**
0	0 – 2	001 - Input Volume 0dB 010 - Input Volume +12dB 100 - Input Volume +24dB 000 - Input Volume +36dB
	3	Always set to 1
	4	Always set to 0
	5	Always set to 0
	6	Input Volume +40dB Boost (active low)
	7	Always set to 1
1	0	Mixer Control – Left Mic to Left Headphone (active high)
	1	Always set to 0
	2	Always set to 0
	3	Always set to 0
	4	Always set to 0
	5	Always set to 0
	6	Pin F Detection [1]
	7	Always set to 1

NOTES:

[1] Pin F Detection is active low when current is being drawn from pin F of the headset connector. When using a headset that requires voltage on pin F (such as the Bose Tri-port) this bit can be used to determine if the headset is plugged in or not. This bit is for detection only and can be ignored if auto-sensing headsets is not desired.

5.3.2 I2C Address 22h

This address is composed of all discrete inputs for the Monitor and Work Switches. During initialization, you should use **UsbI2cWrite** to write FFh to this address to put it in input mode. Then use **UsbI2cRead** to read the state of the switches. All inputs at this address are active low.

I2C Address 22h		
Byte	**Bit**	**Function**
0	0	Monitor Switch - WK
	1	Monitor Switch - A
	2	Monitor Switch - B
	3	Monitor Switch - C
	4	Monitor Switch - D
	5	Monitor Switch - E
	6	Monitor Switch - F
	7	Monitor Switch - ALL
1	0	Work Switch - INT
	1	Work Switch - A
	2	Work Switch - B
	3	Work Switch - C
	4	Work Switch - D
	5	Work Switch - E
	6	Work Switch - F
	7	N/C

5.3.3 I2C Address 23h

This address is composed of all discrete inputs for the Intercom Switch. During initialization, you should use **UsbI2cWrite** to write FFh to this address to put it in input mode. Then use **UsbI2cRead** to read the state of the switches. All inputs at this address are active low.

I2C Address 23h		
Byte	**Bit**	**Function**
0	0	N/C
	1	N/C
	2	N/C
	3	N/C
	4	N/C
	5	N/C
	6	N/C
	7	N/C
1	0	N/C
	1	N/C
	2	N/C
	3	N/C
	4	Intercom – PTT
	5	Intercom – LIVE
	6	Intercom – VOX
	7	Intercom – O/R

5.3.4 I2C Address 48h

This address is composed of the analog inputs that gives the current status of the PTT and to detect PC headset connection. During initialization, you should use **UsbI2cWrite** to write 05h to initialize it. Then use **UsbI2cRead** to read the four bytes of analog input. The range of each input is from 0 to 255.

I2C Address 48h	
Byte	**Function**
0	PTT State [1]
1	Not Used
2	Microphone Bias Detection [2]
3	Not Used

NOTES:

[1] PTT State is the state of the PTT button that is connected to the headset connector. An analog input is used for the PTT since only one pin on the headset connector is provided for PTT but multiple PTT switch settings are desired. In some headsets there will be a radio PTT button and an intercom PTT button. The radio PTT switch usually sets the pin to ground and PTT State will read 0 (or close to 0). The intercom switch usually applies a load to the pin that results in PTT State reading between 0 and 128. PTT State will read approximately 128 when no switch is pressed.

[2] DIP switches on the back of the FFCS box set an internal bias voltage to the microphone input when using a PC-style headset (see Section 3.2). Microphone Bias Detection will read approximately 60% full scale (~150) when the DIP switch is ON and the microphone is not plugged in. When a microphone is plugged in, Microphone Bias Detection will read a lower level depending on the load of the microphone being used. This is for detection only and can be ignored if auto-sensing headsets is not desired.