Date	Rev	Revision Record	Approved
01/30/2003	1.0	Initial Release	KD

AIR/GROUND COMMUNICATIONS
ACB-560

Voice Channel Module (VCM)
Module Manual

This document has been prepared for:

Federal Aviation Administration
William J. Hughes Technical Center
Atlantic City International Airport, NJ 08405

Job	Approvals	Date
Originator:	J CHEN	01/30/2003
Approved:	K DEVITO	01/30/2003
Checked:	K DEVITO	01/30/2003
Checked:	K DEVITO	01/30/2003

CIE ENGINEERING INC.
6001 Woodlake Lane, Alexandria, VA 22310 (703) 922-7061

Title: Voice Channel Module (VCM) Module Manual

Form F01 Microsoft Word	**Sheet** 1 **of** 54	**CAGE No.** 07MD5	**No.** FA100-00002

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

1.0 INTRODUCTION

This Module Manual provides detailed information about the Voice Channel Module (VCM).

The VCM module is part of the NEXCOM Real Time Platform (RTP). The RTP handles the real-time functions of the Prototype Radio Interface Unit (PRIU) and/or the Prototype Ground Network Interface (PGNI). The VCM provides voice processing for the platform supporting local analog audio interfaces, PCM digital data transmission and ATC-10B voice encoding/decoding.

The VCM has been developed for the **NEXCOM Group (ACB-560)** of the Federal Aviation Administration's William J. Hughes Technical Center. The NEXCOM Group supports the following NEXCOM programs:

- Next Generation A/G Communications System (NEXCOM)
- Rapid Prototype Development Effort (RPDE)
- NEXCOM System Demonstrations

1.1 PURPOSE

The purpose of this document is to present Voice Channel Module (VCM) specifications and operating instructions.

1.2 DOCUMENT CONVENTIONS

N/A.

1.3 INTENDED AUDIENCE AND READING SUGGESTIONS

This document is intended for NEXCOM contractors.

1.4 REFERENCES

Reference documentation includes:

- Octal T1 Module (OTM) Module Manual, CIE Document FA100-00020, v1.0, 01/30/2002.
- OTM-VCM Interface Control Document, CIE Document FA100-00066, v1.0, 1/30/2002.
- Analog Devices, 218x DSP Hardware Reference, Analog Devices Part Number 82-002010-01.

1.5 REVISION HISTORY

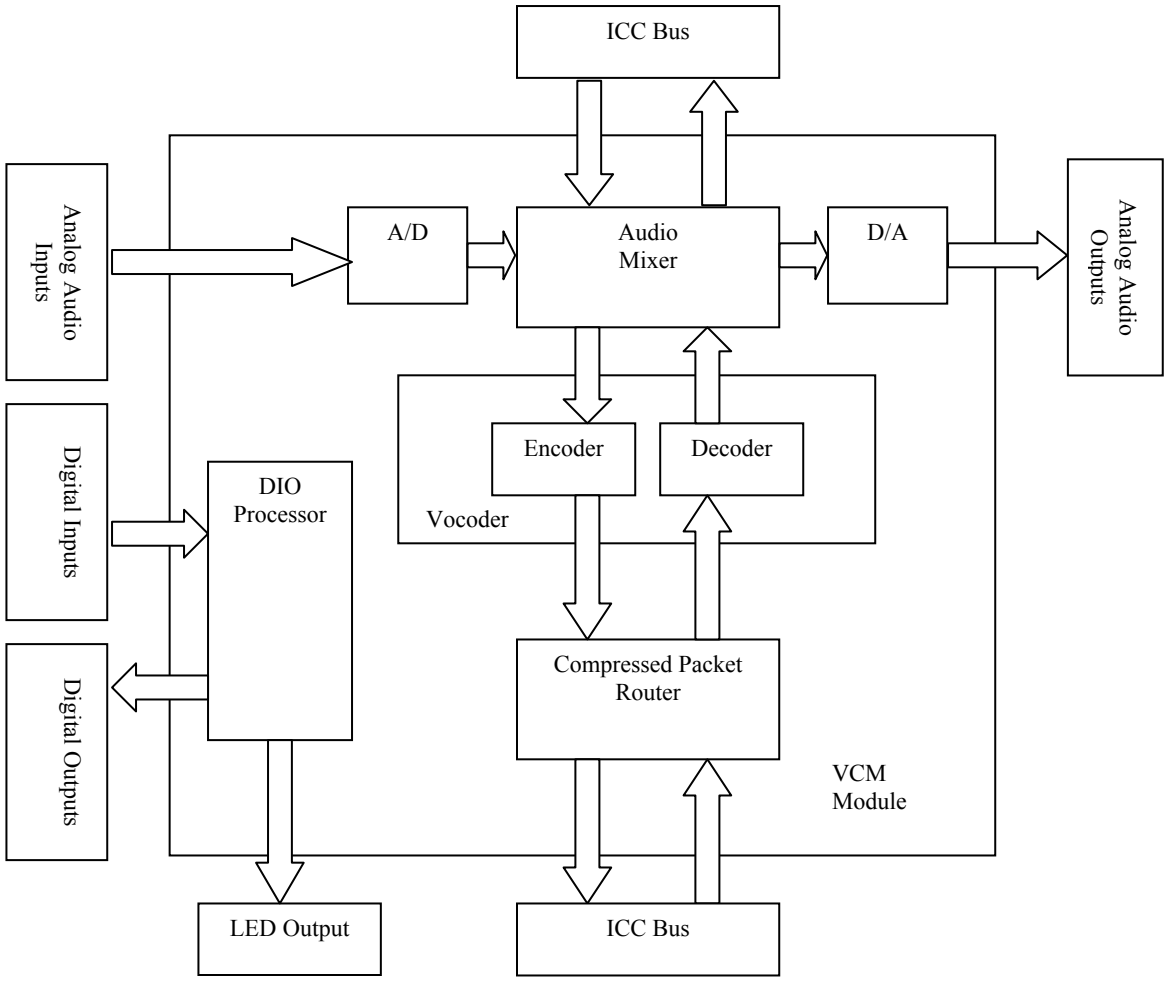
Date	Revision	Description of Changes
01/30/2003	1.0	Initial Release

2.0 GENERAL DESCRIPTION

2.1 OVERVIEW

Figure 1 provides an interface diagram for the Voice Channel Module (VCM).

Figure 1. Voice Channel Module (VCM) Block Diagram



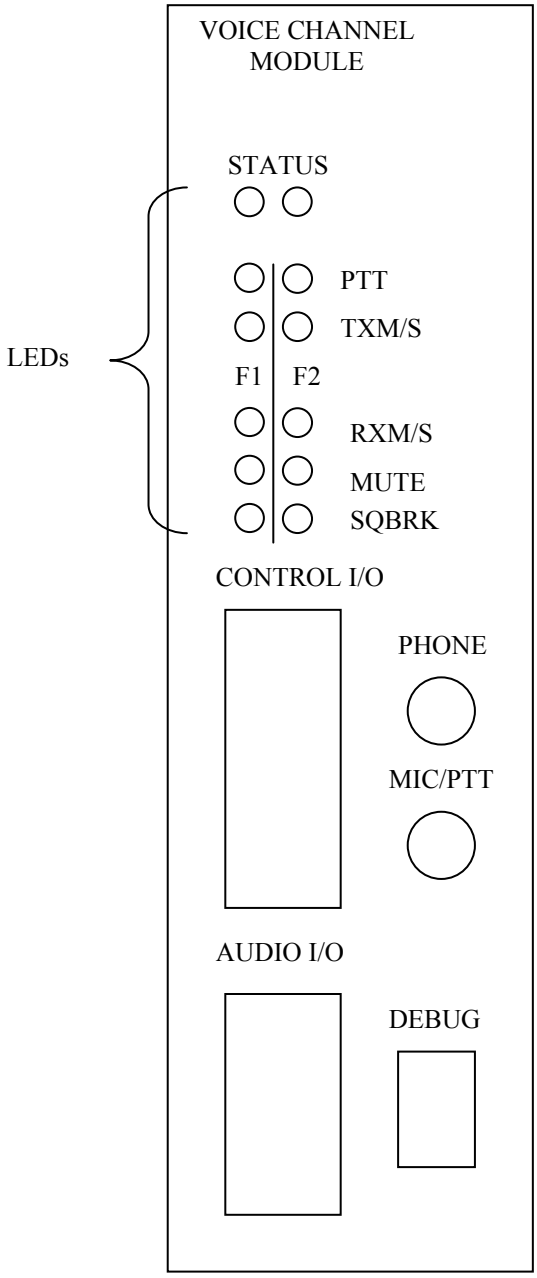
3.0 CONNECTORS & INDICATORS

The VCM includes the following connecters and LED displays

- Control I/O connecter (1 each) –
- Audio I/O connecter (1 each) – Analog audio input and output
- Headphone connecter (1 each) – Output for headset monitor audio
- Microphone/Push to Talk connecter (1 each) – Input for microphone and push to talk local override
- LED Indicators (12) – Unit mode and status information

Figure 2 contains illustrations of the VCM interface panel showing the location of connectors and LEDs. The following sections contain connector/pin out and LED display information. See "**Error! Reference source not found. Error! Reference source not found.**" for functional information.

Figure 2. VCM Interface Panel



3.1 AUDIO INPUT OUTPUT CONNECTOR

The 24 pin audio input output connector provides the interface for transferring analog audio to the VCM module. Table 1 provides connector pin and signal information.

Table 1. Analog Audio Input Output Signal Description			
Symbol	**Pin**	**Pin Type**	**Name/Function**
A1-	24	Input	FMTX1M – Frequency 1 main transmit
A1+	12		
A2-	23	Input	FMTX1S – Frequency 1 standby transmit
A2+	11		
A3-	21	Input	FMTX2M – Frequency 2 main transmit
A3+	10		
A4-	21	Input	FMTX2S – Frequency 2 standby transmit
A4+	9		
A5-	20	Input	UHFRXM – UHF main receive
A5+	8		
A6-	19	Input	UHFRXS – UHF standby receive
A6+	7		
B1-	6	Output	FMRX1M – Frequency 1 main receive
B1+	18		
B2-	5	Output	FMRX1S – Frequency 1 standby receive
B2+	17		
B3-	4	Output	FMRX2M – Frequency 2 main receive
B3+	16		
B4-	3	Output	FMRX2S – Frequency 2 standby receive
B4+	15		
B5-	2	Output	UHFTXM – UHF transmit main
B5+	14		
B6-	1	Output	UHFTXS – UHF transmit standby
B6+	13		

3.2 DIGITAL INPUT OUTPUT CONNECTER

The 50 pin digital input output connecter (DIO) can be used by the RCE to alter the state of the VCM. Such operations such as audio input channel, PTT assert can be initiated through the DIO interface.

Table 2. Digital Input Output Signal Description			
Symbol	**Pin**	**Pin Type**	**Name/Function**
MTX1C+	1	Output	Main transmitter select confirm frequency 1
MTX1C-	26		
STX1C+	2	Output	Standby transmitter select confirm frequency 1
STX1C-	27		
MTX2C+	3	Output	Main transmitter select confirm frequency 2
MTX2C-	28		
STX2C+	4	Output	Standby transmitter select confirm frequency 2
STX2C-	29		
MRX1C+	5	Output	Main receiver select confirm frequency 1
MRX2C-	30		
SRX1C+	6	Output	Standby receiver select confirm frequency 1
SRX1C-	31		
MRX2C+	7	Output	Main receiver select confirm frequency 2 Shorted to each other when asserted, else open
MRX2C-	32		

Table 2. Digital Input Output Signal Description			
Symbol	**Pin**	**Pin Type**	**Name/Function**
SRX2C+	8	Output	Standby receiver select confirm frequency 2
SRX2C-	33		Shorted to each other when asserted, else open
MTX1KC+	9	Output	Main transmitter keying confirm frequency 1
MTX1KC-	34		Shorted to each other when asserted, else open
STX1KC+	10	Output	Standby transmitter keying confirm frequency 1
STX1KC-	35		Shorted to each other when asserted, else open
MTX2KC+	11	Output	Main transmitter keying confirm frequency 2
MTX2KC-	36		Shorted to each other when asserted, else open
STX2KC+	12	Output	Standby transmitter keying confirm frequency 2
STX2KC-	37		Shorted to each other when asserted, else open
SQBRX1C+	13	Output	Receiver squelch break confirm frequency 1
SQBRX1C-	38		Shorted to each other when asserted, else open
SQBRX2C+	14	Output	Receiver squelch break confirm frequency 2
SQBRX2C-	39		Shorted to each other when asserted, else open
TRUNKL+	19	Output	Trunk lockout
TRUNKL-	44		Shorted to each other when asserted, else open
ATRTX+	15	Output	Connected to ground when asserted, else open
MTX2K+	16	Output	Connected to ground when asserted, else open
MUTERX2C+	17	Output	Connected to +12V when asserted, else open
MUTERX1C+	18	Output	Connected to +12V when asserted, else open
ATRRX+	40	Output	Connected to ground when asserted, else open
STX2K+	41	Output	Connected to ground when asserted, else open
+24F	42	Output	Always +24V
+12F	43	Output	Always +12V
MSST1	20	Input	Frequency 1 main/standby transmitter selector
MSSR1	21	Input	Frequency 1 main/standby receiver selector
MTR1	22	Input	Mute 1 assert
MPTT1	23	Input	Frequency 1 main PTT assert
SPTT1	24	Input	Frequency 1 standby PTT assert
MSST2	45	Input	Frequency 2 main/standby transmitter selector
MSSR2	46	Input	Frequency 1 main/standby receiver selector
MTR2	47	Input	Mute 2 assert
MPTT2	48	Input	Frequency 2 main PTT assert
SPTT2	49	Input	Frequency 2 standby PTT assert
GND	25, 50	Ground	

3.3 LED INDICATORS

The VCM module contains 12 external LEDs used to indicate system status. The LEDs have two modes of operation. During POST the LEDs indicate POST status. After POST has completed the LEDs indicate the transmitter and receiver selected for the different frequencies.

3.3.1 POST

During POST the external LEDs are used to indicate POST status. This was done such that there would be a means of retrieving POST status if there were a debug port failure. If a POST test should fail the LEDs will not transition to runtime behavior but will instead hold the last POST status state. The LEDs in the following discussion will be referred to according to Table 3

Table 3. External LED labels		
F1	**F2**	**Label**
11	10	Status
9	8	PTT
7	6	TXM/S
5	4	RXM/S
3	2	MUTE
1	0	SQBRK

Led Meaning:
LED's 2-5 will be used to indicate the numeric identification of the VCM module

	LED State			
VCM id	**LED 8**	**9**	**6**	**7**
0	Green	Off	Off	Off
1	Off	Green	Off	Off
2	Off	Off	Green	Off
3	Off	Off	Off	Green

LED's 6-7 will be used to indicate the clocking of the sport 0 bus

	LED State	
Sport 0 Clocking	**LED 4**	**5**
Local	Green	Off
Remote	Off	Green
Incorrect Rate	Red	Red

LED 0 will be used to indicate the status of the A/D

	Led State
A/D State	**LED 10**
Good	Green
Bad	Red

LED's 8-9 will be used to indicate the number of vocoders present on the VCM

	LED State	
Number Vocoder	**LED 2**	**3**
0	Red	Red
1	Green	Off
2	Off	Green

3.3.2 Runtime LED(s)

After the system has completed booting the LEDs have the following meaning

Table 4. Run Time LEDs			
LED #	**Label**	**State**	**Meaning**
11	Status	Blink Green	System operating normally
		Solid Red	System error detected
10	ICC Activity	Off	No ICC bus activity
		Blink Green	Messages are being transmitted or received on the ICC bus
9	F1 PTT	Off	No PTT asserted
		Green	Frequency 1 main PTT asserted
		Red	Frequency 1 standby PTT asserted
8	F2 PTT	Off	No PTT asserted
		Green	Frequency 2 main PTT asserted
		Red	Frequency 2 standby PTT asserted
7	F1 TXM/S	Green	Frequency 1 main transmitter selected
		Red	Frequency 1 standby transmitter selected
6	F2 TXM/S	Green	Frequency 2 main transmitter selected
		Red	Frequency 2 standby transmitter selected
5	F1 RXM/S	Green	Frequency 1 main receiver selected
		Red	Frequency 1 standby receiver selected
4	F2 RXM/S	Green	Frequency 2 main receiver selected
		Red	Frequency 2 standby receiver selected
3	F1 MUTE	Off	Unused
2	F2 MUTE	Off	Unused
1	F1 SQBRK	Off	Unused
0	F2 SQBRK	Off	Unused

4.0 FUNCTIONAL DESCRIPTION

4.1 VOICE FLOWS

4.1.1 Analog to Digital, Digital to Analog Converter

An Analog Devices AD1843 codec provides A/D and D/A functionality for the VCM module. The codec contains two analog to digital converters, four digital to analog converters, several audio selectors and gain and attenuation control for each D/A, A/D converter.

The codec is configured to sample the audio at an 8 KHz rate. The codec outputs 16-bit signed values.

The A/D channels are labeled ADCL, ADCR. The analog audio (A1 – A6) input passes through a selector before being passed through to the A/D this provides the ability to have more audio input channels than A/D converters.

The D/A channels are labeled DAC2R, DAC2L and DAC1R. The output for each D/A converter is driven on all of its outputs simultaneously. For example the same audio is heard on outputs B1 – B4.

The audio inputs and outputs are grouped by function for each channel. Inputs A1-A4 is collectively known as F1F2TX, A5-A6 UHFRX. The outputs B1-B4 is known as F1F2RX, B5-B6 UHFTX.

The microphone input has the ability to override the input audio on either of the A/D converters. The headset has its own dedicated D/A converter, by utilizing the mixer it is possible for the headset to monitor any combination of inputs and outputs to/from the codec.

4.1.2 Audio Selector Control

The audio selector is controlled by the TX1, TX2, RX1, RX2 commands. TX1 and TX2 are used to modify the selector for input ADCL and RX2 is used to modify the selector for input ADCR. Entering any of the selector commands will toggle between the main and standby channels for that input. For example, if ADCR is currently set to UHFRXM entering 'RX2' will switch the input to UHFRXS.

Because the commands only toggle between the main and standby source for a given channel an interesting issue arises with the ADCL selector which has four input sources. If you use the command 'SHOW FP' it will display the currently selected sources for each input channel

FPT	PT1	PT2	TX1	TX2	RX1	RX2
---	---	---	S	S	M	S

From reading the table it appears that TX1 and TX2 are set to the standby input channel FMTX1S and FMTX2S respectively. If Figure 3 is examined it can be seen that it is not possible for both FMTX1S and FMTX2S to be selected. The active input for ADCL is the last channel selected; the last channel selected is not displayed by the 'SHOW FP' command. In the previous example we do not know whether the selected input for ADCL is FMTX1S or FMTX2S. The following procedure is recommended for ADCL input modification to ensure that the proper input is selected

- Use the 'SHOW FP' command to display the current settings
- If the input source (main or standby) that you want selected is already selected for the channel type the channel twice
- If the input source that you want select is not selected for the channel type the channel once

Example 1:

FMTX1S is the desired input. 'SHOW FP' displays

```
+-----+-----+-----+-----+-----+-----+-----+
| FPT | PT1 | PT2 | TX1 | TX2 | RX1 | RX2 |
+-----+-----+-----+-----+-----+-----+-----+
| --- | --- | --- |  S  |  S  |  M  |  S  |
+-----+-----+-----+-----+-----+-----+-----+
```

'TX1' [ENTER]
'TX1' [ENTER]

Is typed at the command line

Example 2:

FMTX1S is the desired input. 'SHOW FP' displays

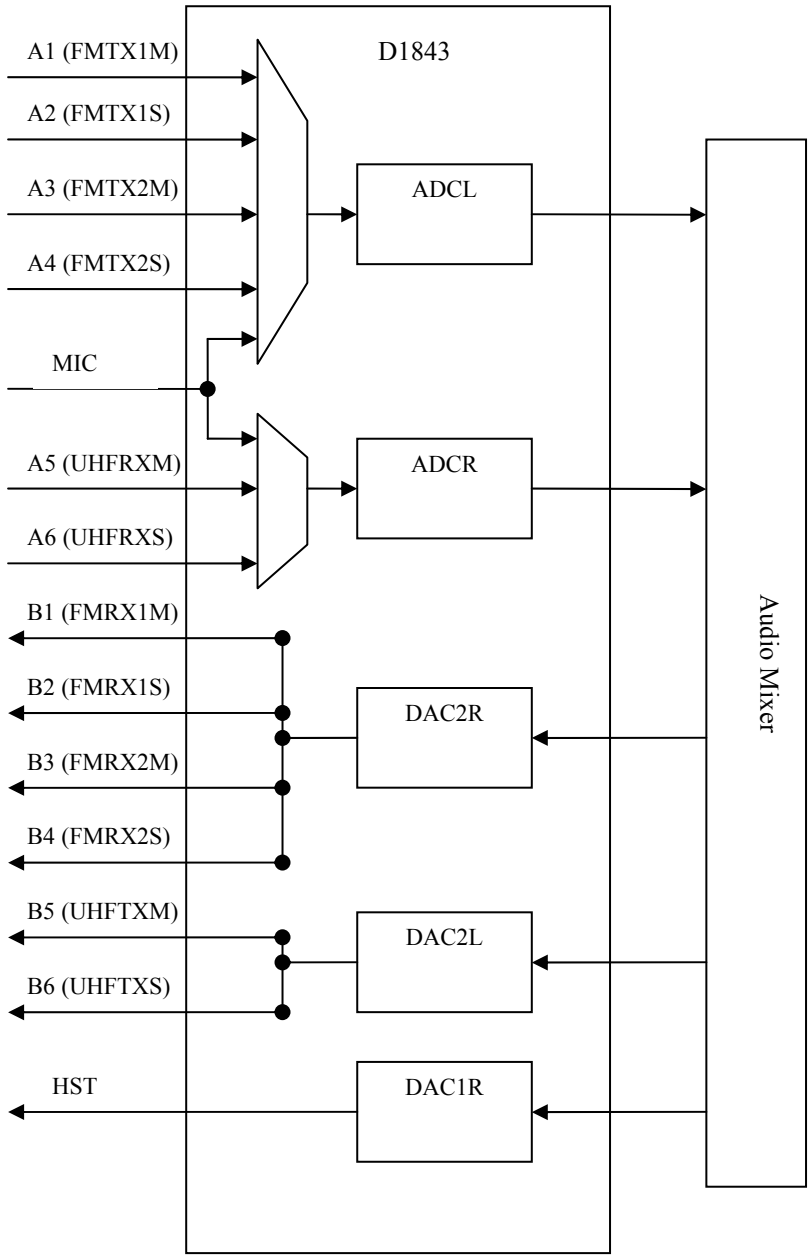
```
+-----+-----+-----+-----+-----+-----+-----+
| FPT | PT1 | PT2 | TX1 | TX2 | RX1 | RX2 |
+-----+-----+-----+-----+-----+-----+-----+
| --- | --- | --- |  M  |  S  |  M  |  S  |
+-----+-----+-----+-----+-----+-----+-----+
```

'TX1' [ENTER]

Is typed at the command line

Figure 3 is a logical diagram of the codec as configured by the VCM.

Figure 3. Analog Audio Connecters



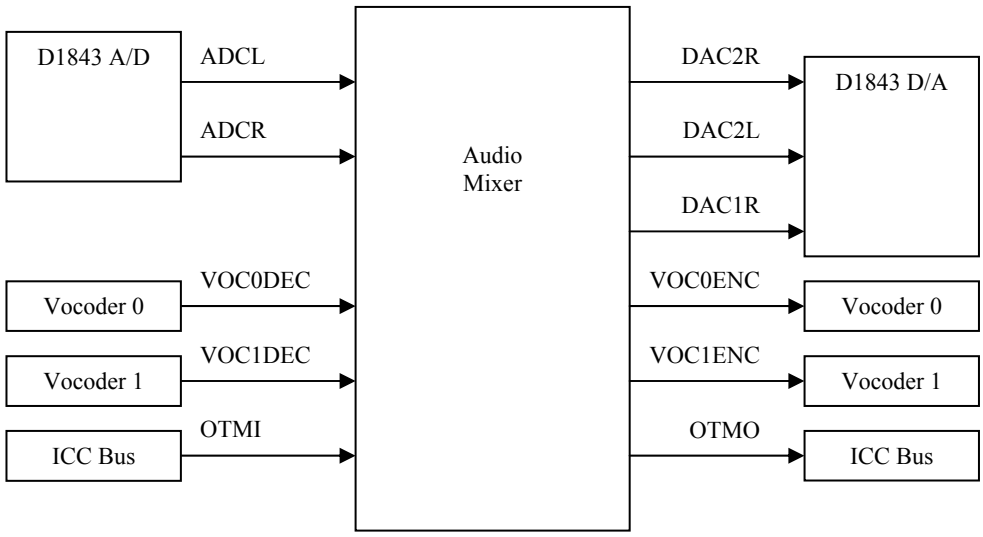
4.2 LINEAR AUDIO MIXER

The mixer provides basic audio mixing and routing capability for linear audio inputs and outputs. The mixer receives and outputs 16-bit signed audio. The majority of the connections to the mixer are to the AD1843 (4.1.1). The mixer also has an additional set of inputs (OTMI) and outputs (OTMO) directly connected to the TDM bus. This set of connecters can be used to transfer linear audio directly to/from the OTM module.

The mixer can combine any number of inputs on a single output. The combining of inputs is implemented by simply summing the input audio sources. If the summed inputs were to exceed a signed 16-bit value the result would saturate.

By default the mixer outputs 0's on an output channel.

Figure 4. Linear Audio Mixer

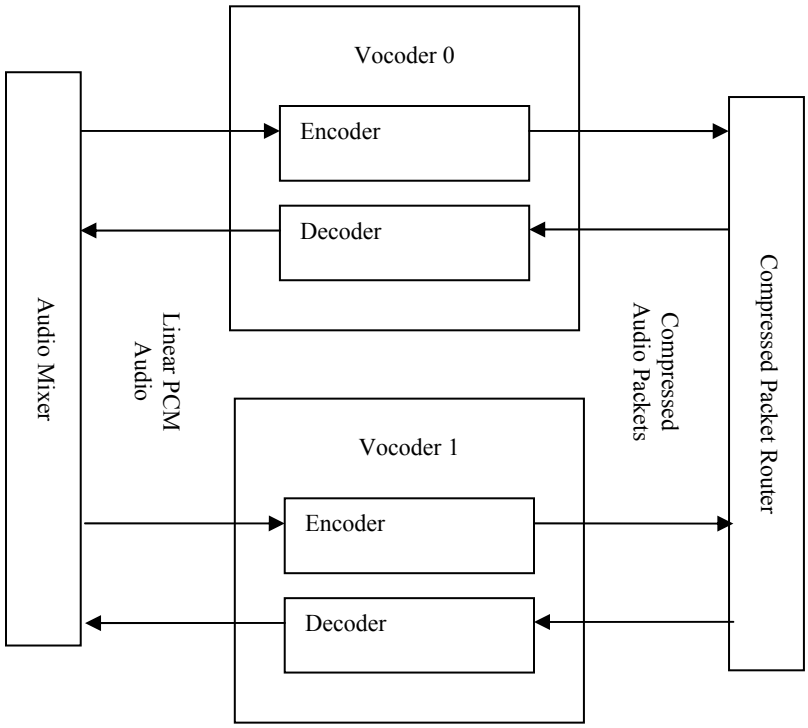


4.3 VOCODER OPERATION

The VCM uses a Digital Voice Systems Inc. (DVSI) VC-20-MTC voice codec (vocoder) to compress/decompress the linear audio provided by the AD1843 codec. The vocoder converts a 128kbps linear audio stream into a 4.8kbps compressed data stream and vice versa. The compressed data stream is provided in packet format and will be referred to as "compressed voice packets" through out this document. The vocoder utilizes the Improved Multi-Band Excitation (AMBE) algorithm to encode the audio.

The VCM can operate with one or two Vocoders. The number of vocoders present in the system is automatically detected during boot. In general the VCM will only operate with a single Vocoder.

Figure 5. Vocoder

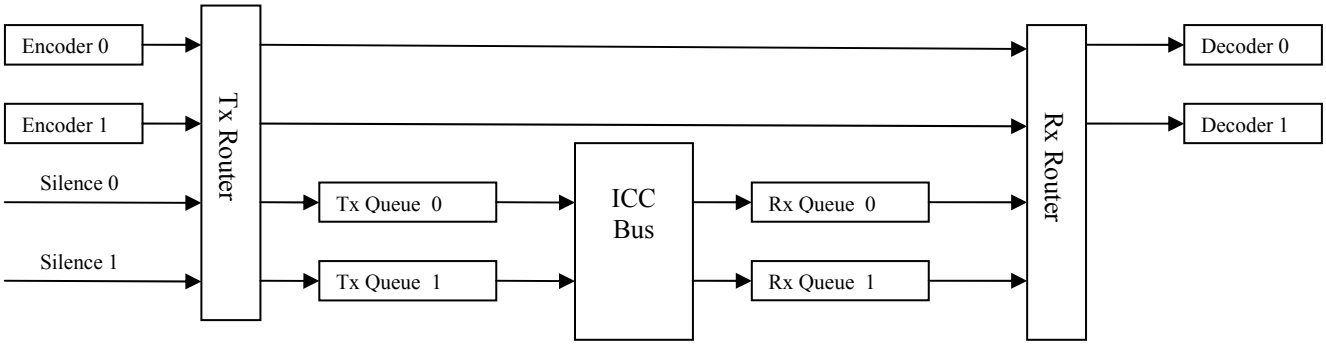


4.4 COMPRESSED PACKET ROUTER

The router is responsible for transferring the compressed voice packets to the correct location in the VCM. At its most basic the router receives compressed voice packets from the encoder section of the Vocoder and transmits compressed voice packets to the decoder section of the Vocoder. The compressed voice routers can make any 1 to 1 connection between the inputs and outputs in a router. The queues leading to and from the ICC bus have a depth of 9 packets. The queue depth of 9 packets was chosen such that the VCM would be able to receive an entire MAC burst (6) of compressed voice packets with room to spare. The router is capable of generating a special "silence" compressed voice packet. The silence packet ensures that the vocoder generates silence on the linear output.

Note: When the compressed packet router is operating in loop back mode (encoder output directly connected to decoder) the compressed packet delay will be greatly reduced

Figure 6. Compressed Packet Router



5.0 DEBUG COMMAND REFERENCE

The VCM provides a simple command line interface. The command line interface is case sensitive and commands must be entered as described in the command listing below

5.1 CONTROL I/O COMMANDS

5.1.1 SHOW FP

Command:	SHOW FP
Description:	Displays state of external switches
Syntax	SHOW FP
Parameters:	None
Notes:	The fields FPT, PT1, PT2 will display '---' if the signal is not asserted. They will display 'KEY' if they are asserted. The fields TX1, TX2, RX1, RX2 display 'S' if the secondary channel is selected. 'M' will be displayed if the main channel is selected.
Examples:	DBG: SHOW FP +-----+-----+-----+-----+-----+-----+-----+ | FPT | PT1 | PT2 | TX1 | TX2 | RX1 | RX2 | +-----+-----+-----+-----+-----+-----+-----+ | --- | --- | --- | S | S | M | S | +-----+-----+-----+-----+-----+-----+-----+
Related Commands:	TX1,TX2, RX1, RX2, PT1, PT2
First Release:	<3.0
Savable:	N/A

5.1.2 TX1

Command:	TX1
Description	Toggles ADCL input selector between TX1 main and TX1 standby
Syntax	TX1
Parameters:	none
Notes:	
Examples:	
Related Commands:	SHOW FP, TX2, RX1, RX2, PT1, PT2
First	<3.0

Release:	
Savable:	N/A

5.1.3 TX2

Command:	TX2
Description:	Toggles ADCL input selector between TX2 main and TX2 standby
Syntax	TX2
Parameters:	none
Notes:	
Examples:	
Related Commands:	SHOW FP, TX1, RX1, RX2, PT1, PT2
First Release:	<3.0
Savable:	N/A

5.1.4 RX1

Command:	RX1
Description:	Toggles ASCR input selector between RX1 main and RX1 standby
Syntax	RX1
Parameters:	none
Notes:	
Examples:	
Related Commands:	SHOW FP, TX1, TX2, RX2, PT1, PT2
First Release:	<3.0
Savable:	N/A

5.1.5 RX2

Command:	RX2
Description:	Toggles RX2 between main and standby
Syntax	RX2
Parameters:	none

Notes:	
Examples:	
Related Commands:	SHOW FP, TX1, TX2, RX1, PT1, PT2
First Release:	<3.0
Savable:	N/A

5.1.6 PT1

Command:	PT1
Description:	Toggles PT1 between asserted and non-asserted
Syntax	PT1
Parameters:	none
Notes:	
Examples:	
Related Commands:	SHOW FP, TX1, TX2, RX1, RX2, PT2
First Release:	<3.0
Savable:	N/A

5.1.7 PT2

Command:	PT2
Description:	Toggles PT2 between asserted and non-asserted
Syntax	PT2
Parameters:	none
Notes:	
Examples:	
Related Commands:	SHOW FP, TX1, TX2, RX1, RX2, PT1
First Release:	<3.0
Savable:	N/A

5.2 AUDIO COMMANDS

5.2.1 ROUTE

Command:	ROUTE
Description:	Sets the audio routing mode
Syntax	ROUTE MODE
Parameters:	MODE – New audio Routing mode 0 = AM 1 = VDL3 2 = test mode 3 = Custom routing mode
Notes:	
Examples:	DBG: ROUTE 1 *<sets system to VDL3 routing mode>* Successful
Related Commands:	SHOW RTMODE, TX, RX, SHOW RXTX
First Release:	<3.0
Savable:	N/A

5.2.2 SHOW RTMODE

Command:	SHOW RTMODE
Description:	Displays the current audio routing mode
Syntax	SHOW RTMODE
Parameters:	none
Notes:	See 5.2.2for a description of the output
Examples:	DBG: SHOW RTMODE Routing: 1 *<indicates VDL3 routing mode>*
Related Commands:	ROUTE, TX, RX, SHOW RXTX
First Release:	<3.0
Savable:	N/A

5.2.3 TX

Command:	TX
Description:	Enables or Disables audio TX transmission
Syntax	TX X

Parameters:	X – New TX state ENABLE = Enable DISABLE = Disable
Notes:	
Examples:	DBG: TX ENABLE *<audio TX transmission enabled>*
Related Commands:	ROUTE, RX, SHOW RXTX
First Release:	<3.0
Savable:	N/A

5.2.4 RX

Command:	RX
Description:	Enables or Disables audio RX reception
Syntax	RX X
Parameters:	X – New RX state ENABLE = Enable DISABLE = Disable
Notes:	
Examples:	DBG: RX ENABLE *<RX transmission enabled>*
Related Commands:	ROUTE, TX, SHOW RXTX
First Release:	<3.0
Savable:	N/A

5.2.5 SHOW RXTX

Command:	SHOW RXTX
Description:	Displays state of RX and TX audio control
Syntax	SHOW RXTX
Parameters:	none
Notes:	0 = Disabled 1 = Enabled
Examples:	DBG: SHOW RXTX RX 1 *<RX transmission is enabled>* TX 0 *<TX transmission is disabled>*
Related Commands:	ROUTE, TX, SHOW RXTX

First Release:	<3.0
Savable:	N/A

5.2.6 SETHST

Command:	SETHST
Description:	Controls the audio channel(s) that the front panel headset should monitor
Syntax	SETHST X
Parameters:	X – Channel the headset should monitor 0 = NONE 1 = F1F2TX 2 = F1F2RX 3 = F1F2 (both TX and RX) 4 = UHFTX 5 = UHFRX 6 = UHF (both TX and RX)
Notes:	The number displayed indicates if the command was successful or not. 1 indicates that the command was successful. 0 indicates that the command was not.
Examples:	DBG: SETHST 1 *<headset is monitoring F1F2TX>* 1 *<command was successful>*
Related Commands:	SHOW HST, SETMIC, SHOW MIC
First Release:	<3.0
Savable:	Yes

5.2.7 SHOW HST

Command:	SHOW HST
Description:	Displays the audio channel(s) that the front panel headset is monitoring
Syntax	SHOW HST
Parameters:	none
Notes:	See 5.2.6for a description of the output
Examples:	DBG: SHOW HST 1 *<Headset is monitoring F1F2 TX>*
Related Commands:	SETHST, SETMIC, SHOW MIC
First Release:	<3.0
Savable:	N/A

5.2.8 SETMIC

Command:	SETMIC
Description:	Controls the audio channel that the front panel microphone should override
Syntax	SETMIC X
Parameters:	X – Audio channel that should be overridden 0 = NONE 1 = F1F2TX 2 = UHFRX
Notes:	The number displayed indicates if the command was successful or not. 1 indicates that the command was successful. 0 indicates that the command was not.
Examples:	DBG: SETMIC 2 *<microphone is overriding UHFRX>* 1 *<command was successful>*
Related Commands:	SETHST, SHOW HST, SHOW MIC
First Release:	<3.0
Savable:	Yes

5.2.9 SHOW MIC

Command:	SHOW MIC
Description:	Displays the audio channel that the front panel microphone is overriding
Syntax	SHOW MIC
Parameters:	none
Notes:	
Examples:	DBG: SHOW MIC 2 *<microphone is monitoring UHFRX>*
Related Commands:	SETHST, SHOW HST, SETMIC
First Release:	<3.0
Savable:	N/A

5.2.10 VRCON

Command:	VRCON
Description:	Voice route connect will create a connection between the ports passed in
Syntax	VRCON PORT1 PORT2 [MQUE]

Parameters:	PORT1, Start of route to be added PORT2, End of route to be added MQUE, Optional, causes voice traffic to be to be routed off board via message queues Valid port choices are: V0 – vocoder 0 V1 – vocoder 1 S0 – silence 0 S1 – silence 1
Notes:	The voice route connect command is used to configure the compressed packet router. Routes start at the receiver side and end at the transmitter side
Examples:	DBG: VRCON V0 V0 MQUE *<encoder V0 is connected to decoder V0 with packets being routed off board>* Created Created
Related Commands:	VRDIS, VRVIEW
First Release:	<3.0
Savable:	N/A

5.2.11 VRDIS

Command:	VRDIS
Description:	Voice route disconnect will disconnect the voice route passed in
Syntax	VRDIS PORT1 [MQUE]
Parameters:	PORT1, Start of route to be disconnected MQUE, Optional, removes route leading to message queues Valid port choices are: V0 – vocoder 0 V1 – vocoder 1 S0 – silence 0 S1 – silence 1
Notes:	The command VRDIS is used to configure the compressed packet router. Routes start at the receiver side and end at the transmitter side.
Examples:	DBG: VRDIS V0 MQUE *<destroy route starting at encoder V0 with off board routing>* Destroyed Destroyed
Related Commands:	VRCON, VRVIEW
First Release:	<3.0
Savable:	N/A

5.2.12 VRVIEW

Command:	VRVIEW

Description:	Voice route view will display the voice routes present
Syntax	VRVIEW
Parameters:	none
Notes:	The command VRVIEW display the current configuration of the compressed packet router
Examples:	DBG: VRVIEW RX TX ----- S1 V1 *<silence 1 is being routed to decoder V1>*
Related Commands:	VRCON, VRDIS
First Release:	<3.0
Savable:	N/A

5.2.13 SETVOL

Command:	SETVOL
Description:	Set volume
Syntax	SETVOL CHANNEL VOLUME
Parameters:	CHANNEL – The channel whose audio needs to be modified 0 - F1MTX, valid range 0 - 15 1 - F1STX, valid range 0 - 15 2 - F2MTX, valid range 0 - 15 3 - F2STX, valid range 0 - 15 4 - UHFMRX, valid range 0 - 15 5 - UHFSRX, valid range 0 - 15 6 - MIC, valid range 0 - 15 7 - F1F2RX, valid range 0 - 63 8 - UHFTX, valid range 0 - 63 9 - Hst, valid range 0 - 63 VOLUME – The new volume for the audio channel
Notes:	
Examples:	DBG: SETVOL 0 8 *<The volume for F1F2TX is set to 8>*
Related Commands:	SHOW VOL
First Release:	4.0
Savable:	Yes

5.2.14 SHOW VOL

Command:	SHOW VOL

Description:	Displays input and output volume for the different audio channels
Syntax	SHOW VOL CHANNEL
Parameters:	CHANNEL – The channel whose volume will be displayed 0 - F1MTX, valid range 0 - 15 1 - F1STX, valid range 0 - 15 2 - F2MTX, valid range 0 - 15 3 - F2STX, valid range 0 - 15 4 - UHFMRX, valid range 0 - 15 5 - UHFSRX, valid range 0 - 15 6 - MIC, valid range 0 - 15 7 - F1F2RX, valid range 0 - 63 8 - UHFTX, valid range 0 - 63 9 - Hst, valid range 0 - 63
Notes:	
Examples:	DBG: SHOW VOL 2 8 *<F1F2RX volume is set to 8>*
Related Commands:	SETVOL
First Release:	4.0
Savable:	N/A

5.2.15 STVOL

Command:	STVOL [VOL]
Description:	Display or modifies the volume of the side tone heard on the headset when using the microphone override
Syntax	STVOL [VOL]
Parameters:	VOL – 0 – 0x7fff where 0 is no side tone and 0x7fff is maximum side tone level
Notes:	STVOL with out any parameters will display the current side tone volume. STVOL [VOL] will change the volume to VOL.
Examples:	*DBG: STVOL 800 <set side tone to 800)* *DBG: STVOL <request display of side tone volume>* *800*
Related Commands:	SETVOL
First Release:	6.0
Savable:	Yes

5.3 UTILITY COMMANDS

5.3.1 SAVE CONFIG

Command:	SAVECONFIG
Description:	Saves current configuration to flash
Syntax	SAVECONFIG
Parameters:	none
Notes:	'Pass' will be displayed if we are able to save the current configuration to flash. 'Fail' will be displayed if we are not able to save the current configuration to flash.
Examples:	`DBG: SAVECONFIG` `Pass`
Related Commands:	DEFAULTCONFIG,
First Release:	`4.0`
Savable:	`N/A`

5.3.2 DEFAULT CONFIG

Command:	DEFAULTCONFIG
Description:	Changes saved configuration to default values. Note the system will need to be power cycled before the new values take effect
Syntax	DEFAULTCONFIG
Parameters:	none
Notes:	'Pass' will be displayed if we are able to save the current configuration to flash. 'Fail' will be displayed if we are not able to save the current configuration to flash.
Examples:	`DBG: DEFAULTCONFIG` `Pass`
Related Commands:	SAVECONFIG
First Release:	`4.0`
Savable:	`N/A`

5.4 DEBUGGING COMMANDS

5.4.1 SHOW AUTOOTM

Command:	SHOW AUTOOTM
Description:	Displays if VCM has been enabled to send autonomous messages to the OTM module

Syntax	SHOW AUTOOTM
Parameters:	none
Notes:	Displays "Enabled" if AUTOOTM has been enabled, "Disabled" if AUTOOTM has not been enabled.
Examples:	`DBG: SHOW AUTOOTM` `Enabled <AUTOOTM has been enabled>`
Related Commands:	
First Release:	`<3.0`
Savable:	`N/A`

5.4.2 SHOW CVMCTR

Command:	SHOW CVMCTR
Description:	Displays a count of the number of compressed voice packets sent/received to/from the OTM. The numbers are displayed in hexadecimal format with the number of packets transmitted displayed first.
Syntax	SHOW CVMCTR
Parameters:	none
Notes:	The count of compressed packets is displayed in hexadecimal format. The first number displayed is for transmitted packets The second number displayed is for received packets
Examples:	`DBG: SHOW CVMCTR` `0x000AC02 0x0000ABF0`
Related Commands:	ZERO CVMCTR
First Release:	`<3.0`
Savable:	`N/A`

5.4.3 ZEROCVMCTR

Command:	ZEROCVMCTR
Description:	Clears the transmit and received compressed voice counters
Syntax	ZEROCVMCTR
Parameters:	none
Notes:	
Examples:	
Related Commands:	SHOW CVMCTR

First Release:	<3.0
Savable:	N/A

5.4.4 R0

Command:	R0
Description:	Displays the sport 0 receive buffer
Syntax	R0
Parameters:	none

Notes:

Slot(s)	RX Data	TX Data
0-5	AD1843 output data	AD1843 input data
6	Vocoder 0 transmit data	Vocoder 0 receive data
7	Vocoder 1 transmit data	Vocoder 1 receive data
8	System frame count (from OTM)	Unused
9	Unused	VCM 0 PCM audio
10	VCM 0 ICCPHY receive data	VCM 0 ICCPHY transmit data
11	Unused	VCM 1 PCM audio
12	VCM 1 ICCPHY receive data	VCM 1 ICCPHY transmit data
13	Unused	VCM 2 PCM audio
14	VCM 2 ICCPHY receive data	VCM 2 ICCPHY transmit data
15	Unused	VCM 3 PCM audio

Examples:

```
DBG: R0
SPORT_0 RX Data (Snapshot)
382A: 0303 0000 0076 ... FFFF FFFF FFFF FFFF
383B: 0303 0001 0073 ... FFFF FFFF FFFF FFFF
384C: 0303 0000 0077 ... FFFF FFFF FFFF FFFF

                         .
                         .
                         .

39E4: 0303 0000 0075 ... FFFF FFFF FFFF FFFF
39F5: 0303 0555 0074 ... FFFF FFFF FFFF FFFF
3A06: 0303 93D3 0073 ... FFFF FFFF FFFF FFFF
3A17: 0303 5C18 0072 ... FFFF FFFF FFFF FFFF
3A28: 0303 0000 0071 ... FFFF FFFF FFFF FFFF
3A39: 0303 0000 0072 ... FFFF FFFF FFFF FFFF
```

Sport 0 super frame 32 frames 0 to 31

Sport 0 Frame
17 Slots numbered 0 – 16
from left to right

Related Commands:	T0
First Release:	<3.0

Savable:	N/A

5.4.5 T0

Command:	T0
Description:	Displays the sport 0 transmit buffer
Syntax	T0
Parameters:	none
Notes:	See 5.4.5for a full description of the output
Examples:	
Related Commands:	R0
First Release:	<3.0
Savable:	N/A

5.4.6 R1

Command:	R1
Description:	Displays the sport 1 receive buffer
Syntax	R1
Parameters:	none

Notes:

Slot(s)	RX Data	TX Data
0	Vocoder 0, 1 nibble and control data receive (from vocoder)	Unused
1-18	Unused	Unused
19	Unused	Vocoder 0, 1 nibble and control

Examples:

```
DBG: R1
SPORT_1 RX Data (Snapshot)
382A: 3000 0000 0000 ... 0000 0000 0000 0000
383E: 3000 0000 0000 ... 0000 0000 0000 0000
3852: 3000 0000 0000 ... 0000 0000 0000 0000
                         .
                         .
                         .
3A32: 3000 0000 0000 ... 0000 0000 0000 0000
3A46: 3000 0000 0000 ... 0000 0000 0000 0000
3A5A: 3000 0000 0000 ... 0000 0000 0000 0000
3A6E: 3000 0000 0000 ... 0000 0000 0000 0000
3A82: 3000 0000 0000 ... 0000 0000 0000 0000
3A96: 3000 0000 0000 ... 0000 0000 0000 0000
```

Sport 1 super frame 32 frames 0 to 31

Sport 1 Frame
20 Slots numbered 0 – 19
from left to right

Related Commands:	T1
First Release:	<3.0
Savable:	N/A

5.4.7 T1

Command:	T1
Description:	Displays the sport 1 transmit buffer
Syntax	T1
Parameters:	none
Notes:	See 5.4.6for a full description of the output
Examples:	
Related Commands:	T0
First Release:	<3.0
Savable:	N/A

5.4.8 V0

Command:	V0
Description:	Displays vocoder 0 transmit and receive compressed nibbles
Syntax	V0
Parameters:	none
Notes:	For both the receive and transmit packet the bytes are ordered as follows 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 Nibble 0 in the TX packet is the vocoder control nibble. Nibble 0 on the receive side is the vocoder status nibble. The "Handle Data" field is for software debug purposes and is uninteresting in most cases.
Examples:	DBG: V0 V0 Nibble Data (Snapshot) RX Data 3798: 0000 000a 0006 0007 000d 379d: 0003 000e 0006 000c 000c 37a2: 000e 0007 000c 0001 0007 37a7: 000d 0009 0008 0002 0006 37ac: 0008 000b 000e 0007 0001 TX Data 37b3: 0000 0009 0008 0009 000e 37b8: 0006 0004 0002 000a 000f 37bd: 0001 000c 000d 0003 0006 37c2: 000c 0008 000a 000f 0007 37c7: 0001 0004 0005 0002 0006 Handle Data 2633: 0000 0000 0006 0000 0020 2638: 0014 0280 0013 0000 0000 263d: 0000 0000 0000 0000 0001 Compressed audio packet received from encoder Compressed audio packet sent to decoder
Related Commands:	V1
First Release:	<3.0
Savable:	N/A

5.4.9 V1

Command:	V1
Description:	Displays vocoder 0 transmit and receive compressed nibbles
Syntax	V1
Parameters:	none
Notes:	See 5.4.8for a full description of the output format
Examples:	
Related Commands:	V0
First Release:	<3.0
Savable:	N/A

5.4.10 VRVIEWFLUSHCOUNT

Command:	VRVIEWFLUSHCOUNT
Description:	Will display count of number of times the message queues associated with voice routing overflowed and needed to be flushed. The number displayed is in hexadecimal format.
Syntax	VRVIEWFLUSHCOUNT
Parameters:	none
Notes:	The count of message queue flushes is displayed in hexadecimal format
Examples:	DBG: VRVIEWFLUSHCOUNT Message Queues Flushed 1f71: 0000 *<message queue flushed zero times>*
Related Commands:	
First Release:	<3.0
Savable:	N/A

5.4.11 DIR

Command:	DIR
Description:	Reads a I/O space memory location
Syntax	DIR ADDR
Parameters:	ADDR – I/O space memory address to be read in hexadecimal format
Notes:	The value at the I/O space memory location is displayed in hexadecimal format
Examples:	`DBG: DIR 2` `0x0002 0000`
Related Commands:	DIW
First Release:	5.0
Savable:	N/A

5.4.12 DIW

Command:	DIW
Description:	Writes a I/O space memory location with a given value
Syntax	DIW ADDR VALUE
Parameters:	ADDR – I/O space memory address to be written in hexadecimal format VALUE – Value to be written to location in hexadecimal format
Notes:	
Examples:	`DBG: DIW 2 3` `0x0002 0003`
Related Commands:	DIR
First Release:	5.0
Savable:	N/A

5.4.13 DMR

Command:	DMR
Description:	Read a data memory location
Syntax	DMR ADDR
Parameters:	ADDR – I/O space memory address to be read in hexadecimal format
Notes:	
Examples:	`DBG: DMR A` `0x000a 0000`
Related Commands:	DMW
First Release:	5.0
Savable:	N/A

5.4.14 DMW

Command:	DMW
Description:	Writes a data memory location with a given value
Syntax	DMW ADDR VALUE
Parameters:	ADDR – data memory address to be written in hexadecimal format VALUE – Value to be written to location in hexadecimal format
Notes:	
Examples:	`DBG: DMW 2 3` `0x0002 0003`
Related Commands:	DMR
First Release:	5.0
Savable:	N/A

5.4.15 DIOR

Command:	DIOR
Description:	Reads the requested DIO control register
Syntax	DIOR REGNUM
Parameters:	REGNUM – Number of DIO control register to read
Notes:	DIO Register Offsets. See TBA for a full description of the registers

Number	Register Name
0	CDINHW
1	CDINSW
2	CDINSEL
3	CDIN
4	CTCFG
5	CLOUTSEL
6	CLOUTSW0
7	CLOUTSW1
8	CLOUTSW2
9	CLOUTSW3
A	CLOUTSW4
B	CLOUTSW5
C	CLOUTSW6
D	CLOUTSW7
E	CLOUTSW8
F	CLOUTSW9
10	CLOUTSW10
11	CLOUTSW11
12	CLOUT0
13	CLOUT1
14	CLOUT2
15	CDOUTSEL
16	CDOUTSW
17	CDOUTTBL
18	CDOUT
19	CDOUTHW0
1A	CDOUTHW1
1B	CDOUTHW2

Examples:	`DBG: DIOR 9` `0x0009 0x00000000`
Related Commands:	DIOW
First Release:	5.0
Savable:	N/A

5.4.16 DIOW

Command:	DIOW
Description:	Writes the requested DIO control register
Syntax	DIOW REGNUM MSW LSW
Parameters:	REGNUM – Number of DIO control register to written MSW – Most significant 16-bits to be written LSW – Least significant 16-bits to be written
Notes:	DIO Register Offsets. See TBA for a full description of the registers See 5.4.15for a list of register offsets.
Examples:	`DBG: DIOW 8 1234 5678` `0x0008 0x12345678`
Related Commands:	DIOR
First Release:	`5.0`
Savable:	`N/A`

5.4.17 UARTCVMTX

Command:	UARTCVMTX
Description:	Causes compressed voice packets to be sent out VIA the debug port
Syntax	UARTCVMTX
Parameters:	none
Notes:	Press the 'escape' key to stop transmission of compressed voice packets over the UART

Examples:	`DBG: UARTCVMTX`
	`DBG: 00000173acaeb54a7e753d206272fb07db`
	`00000219ec796d46712f881be397590741`
	`00000358fd79a59b679a0c47db452a07c2`
	`0000045d30207893f430a1fff0b38e0780`**<ESC>**

Related Commands:	UARTCVMRX, HWFLOWON, HWFLOWOFF
First Release:	`5.0`
Savable:	`N/A`

5.4.18 UARTCVMRX

Command:	UARTCVMRX
Description:	Causes compressed voice packets to be received via the debug port
Syntax	UARTCVMRX
Parameters:	none
Notes:	Press the 'escape' key to stop reception of compressed voice packets over the debug port Hardware flow control of the debug port MUST be enabled via the HWFLOWON command before using the UARTCVMRX command.
Examples:	
Related Commands:	UARTCVMTX, HWFLOWON, HWFLOWOFF
First Release:	5.0
Savable:	N/A

5.4.19 HWFLOWON

Command:	HWFLOWON
Description:	Causes hardware flow control to be enabled on the debug port
Syntax	HWFLOWON
Parameters:	none
Notes:	It is important that hardware flow control be enabled in the terminal program after the VCM module
Examples:	
Related Commands:	UARTCVMRX, UARTCVMTX, HWFLOWOFF
First Release:	5.0
Savable:	N/A

5.4.20 HWFLOWOFF

Command:	HWFLOWOFF
Description:	Causes hardware flow control to be disabled on the debug port
Syntax	HWFLOWOFF
Parameters:	none
Notes:	
Examples:	
Related Commands:	UARTCVMRX, UARTCVMTX, HWFLOWON
First Release:	5.0
Savable:	N/A

5.4.21 CLRSTATLED

Command:	CLRSTATLED
Description:	Sets the status LED to normal (blinking green) status
Syntax	CLRSTATLED
Parameters:	none
Notes:	If the error that caused the LED to turn RED has not been corrected the status LED will immediately revert to the RED state
Examples:	
Related Commands:	
First Release:	5.0
Savable:	N/A

5.4.22 LEDTEST

Command:	LEDTEST
Description:	Causes the front panel LEDs to be toggled in a distinctive pattern
Syntax	LEDTEST
Parameters:	none
Notes:	
Examples:	

Related Commands:	
First Release:	
Savable:	N/A

5.4.23 ADW

Command:	ADW
Description:	Writes a D1843 codec control register
Syntax	ADW OFFSET VALUE
Parameters:	ADDR – Offset of codec register to be written in hexadecimal format VALUE – Value to be written to offset in hexadecimal format
Notes:	
Examples:	DBG: ADW A FA10 0x000a FA10
Related Commands:	DMW
First Release:	6.0
Savable:	N/A

5.4.24 SDSP

Command:	SDSP
Description:	Displays current state of D1843 codec control registers
Syntax	SDSP
Parameters:	None
Notes:	
Examples:	DBG: SDSP To DSP 031f: 4001 0000 2a46 8888 0323: 8888 8888 8888 8888 0327: 88e8 880a 0a0a 0000 032b: 0000 8080 8080 0505 032f: 80ff 1f40 0000 8055 0333: 1f40 0000 00ff bb80 0337: 0000 0000 0555 93d3 033b: 5c18 0000 0000 0000 From DSP 033f: 0001 0000 2a46 8888 0343: 8888 8888 8888 8888 0347: 88e8 880a 0a0a 0000 034b: 0000 8080 8080 0505 034f: 80ff 1f40 0000 8055 0353: 1f40 0000 00ff bb80 0357: 0000 0000 0555 93d3 035b: 5c18 0000 0000 0000
Related Commands:	DMR
First Release:	6.0
Savable:	N/A

5.5 COMMANDS WITH SAVABLE PARAMETERS

Command	*Parameter*	*Default Value*
SETVOL	0 (F1MTX)	6
	1 (F1STX)	6
	2 (F2MTX)	6
	3 (F2STX)	6
	4 (UHFMRX)	6
	5 (UHFSRX)	6
	6 (MIC)	6
	7 (F1F2RX)	10
	8 (UHFTX)	10
	9 (HST)	10
SETHST	X	2 (F1F2RX)

SETMIC	X	1 (F1F2TX)
STVOL	X	0x1800

5.6 DIGITAL INPUT OUTPUT

The Digital Input Output (DIO) section of the VCM is responsible for monitoring the 11 digital input signals and controlling the 24 digital output signals and the 12 bi-color LEDs on the front panel. In normal DIO operation the digital inputs will be de-bounced, combined and compared to one of three internal tables. The output(s) from the table will be used to drive the digital outputs and the LEDs. To facilitate testing it is possible to manually (via software) to control any of the digital inputs, digital outputs or LEDs.

The DIO registers described in this section are accessible from the DIOR and DIOW commands.

5.6.1 DIO Selector

There are three selector registers that determine which source should drive the output. The selector operates on a bit by bit basis. For example if C_LOUTSEL were set to 0x0011 LEDs 0 and 4 would be driven by the table with all other LEDs being driven by the corresponding software control register.

5.6.2 Software DIO Control

Software control is the simplest method of controlling the DIO. In the software control method the bit in the control register is directly used to drive the output.

5.6.3 Table DIO Control

The Hardware Output Table Processor (HOTP) sets the table driven output states for based on the input signal states. The HOTP reads through rows in the hardware output table (HOT) to determine the appropriate output states. The table format is shown in Figure TBD. It also includes sample entries.

The C_TRESULT register contains the logical device number associated with the digital and LED outputs.

Table 5 Hardware Output Table Processor Format

TODEVID	TIMASK	TIDATA	TRESULT
0x00	0x00000001	0x00000000	0x00
0x00	0x00000001	0x00000001	0x01
0x80	0x00000001	0x00000000	0x00
0x80	0x00000001	0x00000001	0x03
0x01	0x00000002	0x00000000	0x00
0x01	0x00000002	0x00000002	0x01

For each row in the table, the HOTP performs the following operations:

if ((C_DIN & TIMASK) == TIDATA), then TODEVID = TRESULT

After each row is processed, the HOTP takes the new output values and creates a **xOUTBL** value (all the bits set/reset as applicable).

The following table details how TODEVID values are mapped to the outputs from the DIO

Table 6 Hardware Output Table Output ID's

TODEVID	Type	Name

0x00	Digital Output	RX2MC
0x01	Digital Output	RX2SC
0x02	Digital Output	RX1MC
0x03	Digital Output	RX1SC
0x04	Digital Output	TX2MC
0x05	Digital Output	TX2SC
0x06	Digital Output	TX1MC
0x07	Digital Output	TX1SC
0x08	Digital Output	SQBR2C
0x09	Digital Output	RX2MS
0x0a	Digital Output	SQBR1C
0x0b	Digital Output	TX2MS
0x0c	Digital Output	PTT2C
0x0d	Digital Output	SPARE2
0x0e	Digital Output	PTT1C
0x0f	Digital Output	SPARE3
0x10	Digital Output	SPARE1
0x11	Digital Output	N/A
0x12	Digital Output	LOCOUT
0x13	Digital Output	N/A
0x14	Digital Output	PTT2S
0x15	Digital Output	RX1MUTEC
0x16	Digital Output	PTT2M
0x17	Digital Output	RX2MUTEC
0x18	LED	0
0x19	LED	1
0x1a	LED	2
0x1b	LED	3
0x1c	LED	4
0x1d	LED	5
0x1e	LED	6
0x1f	LED	7
0x20	LED	8

0x21	LED	9
0x22	LED	10
0x23	LED	11

TRESULT can have two different ranges of values depending on the TODEVIDTYPE (Digital Output, LED). The following table lists the possible values for each output type and its meaning

Table 7 TRESULT values

Type	TIRESULT Value	Meaning
Digital Output	0	Clear output
Digital Output	1	Set output
LED	0	Off
LED	1	Green, Solid
LED	2	Red, Solid
LED	3	Green, Flashing
LED	4	Red, Flashing

5.6.3.1 DIO Control Tables

Table 8 DIO Table 0

TODEVID	TIMASK	TIDATA	TRESULT
0x21	0x00000003	0x00000000	0x00
0x21	0x00000003	0x00000001	0x01
0x21	0x00000003	0x00000002	0x02
0x1f	0x00000004	0x00000000	0x01
0x1f	0x00000004	0x00000004	0x02
0x1d	0x00000008	0x00000000	0x01
0x1d	0x00000008	0x00000008	0x02
0x19	0x00200000	0x00000000	0x00
0x19	0x00200000	0x00200000	0x01
0x16	0x00000020	0x00000000	0x00
0x16	0x00000020	0x00000020	0x01
0x20	0x00000060	0x00000000	0x00
0x20	0x00000060	0x00000020	0x01
0x20	0x00000060	0x00000040	0x02
0x14	0x00000040	0x00000000	0x00
0x14	0x00000040	0x00000040	0x01
0x0b	0x00000080	0x00000000	0x00
0x0b	0x00000080	0x00000080	0x01
0x09	0x00000100	0x00000000	0x00
0x09	0x00000100	0x00000100	0x01
0x1e	0x00000080	0x00000000	0x01
0x1e	0x00000080	0x00000080	0x02
0x1c	0x00000100	0x00000000	0x01
0x1c	0x00000100	0x00000100	0x02
0x18	0x00000200	0x00000000	0x00
0x18	0x00000200	0x00000200	0x01
0x22	0x01800000	0x00000000	0x00
0x22	0x01800000	0x00800000	0x01
0x22	0x01800000	0x01000000	0x02

Table 9 DIO Table 1			
TODEVID	**TIMASK**	**TIDATA**	**TRESULT**
0x0E	0x00000001	0x00000000	0x00
0x0E	0x00000001	0x00000001	0x01
0x21	0x00000005	0x00000000	0x00
0x21	0x00000005	0x00000001	0x01
0x21	0x00000005	0x00000004	0x00
0x21	0x00000005	0x00000005	0x02
0x06	0x00000004	0x00000000	0x01
0x06	0x00000004	0x00000004	0x00
0x07	0x00000004	0x00000000	0x00
0x07	0x00000004	0x00000004	0x01
0x1F	0x00000004	0x00000000	0x01
0x1F	0x00000004	0x00000004	0x02
0x02	0x00000008	0x00000000	0x01
0x02	0x00000008	0x00000008	0x00
0x03	0x00000008	0x00000000	0x00
0x03	0x00000008	0x00000008	0x01
0x1D	0x00000008	0x00000000	0x01
0x1D	0x00000008	0x00000008	0x02
0x15	0x00000010	0x00000000	0x00
0x15	0x00000010	0x00000010	0x01
0x1B	0x00000010	0x00000000	0x00
0x1B	0x00000010	0x00000010	0x02
0x0A	0x00200000	0x00000000	0x00
0x0A	0x00200000	0x00200000	0x01
0x19	0x00200000	0x00000000	0x00
0x19	0x00200000	0x00200000	0x01
0x16	0x000000A0	0x00000000	0x00
0x16	0x000000A0	0x00000020	0x01
0x16	0x000000A0	0x00000080	0x00
0x16	0x000000A0	0x000000A0	0x00
0x20	0x000000A0	0x00000000	0x00
0x20	0x000000A0	0x00000020	0x01
0x20	0x000000A0	0x00000080	0x00
0x20	0x000000A0	0x000000A0	0x02
0x14	0x000000A0	0x00000000	0x00
0x14	0x000000A0	0x00000020	0x00
0x14	0x000000A0	0x00000080	0x00
0x14	0x000000A0	0x000000A0	0x01
0x0B	0x00000080	0x00000000	0x00
0x0B	0x00000080	0x00000080	0x01
0x09	0x00000100	0x00000000	0x00
0x09	0x00000100	0x00000100	0x01
0x0C	0x00000020	0x00000000	0x00
0x0C	0x00000020	0x00000020	0x01
0x04	0x00000080	0x00000000	0x01
0x04	0x00000080	0x00000080	0x00
0x05	0x00000080	0x00000000	0x00
0x05	0x00000080	0x00000080	0x01
0x1E	0x00000080	0x00000000	0x01
0x1E	0x00000080	0x00000080	0x02
0x00	0x00000100	0x00000000	0x01
0x00	0x00000100	0x00000100	0x00
0x01	0x00000100	0x00000000	0x00
0x01	0x00000100	0x00000100	0x01
0x1C	0x00000100	0x00000000	0x01
0x1C	0x00000100	0x00000100	0x02
0x17	0x00000200	0x00000000	0x00
0x17	0x00000200	0x00000200	0x01
0x1A	0x00000200	0x00000000	0x00
0x1A	0x00000200	0x00000200	0x02

0x08	0x00000040	0x00000000	0x00
0x08	0x00000040	0x00000040	0x01
0x18	0x00000040	0x00000000	0x00
0x18	0x00000040	0x00000040	0x01
0x12	0x00400000	0x00000000	0x00
0x12	0x00400000	0x00400000	0x01
0x23	0x00400000	0x00000000	0x00
0x23	0x00400000	0x00400000	0x02
0x22	0x01800000	0x00000000	0x00
0x22	0x01800000	0x00800000	0x01

Table 10 DIO Table 2

TODEVID	TIMASK	TIDATA	TRESULT
0x21	0x00000003	0x00000000	0x00
0x21	0x00000003	0x00000001	0x01
0x21	0x00000003	0x00000002	0x02
0x1f	0x00000004	0x00000000	0x01
0x1f	0x00000004	0x00000004	0x02
0x1d	0x00000008	0x00000000	0x01
0x1d	0x00000008	0x00000008	0x02
0x19	0x00200000	0x00000000	0x00
0x19	0x00200000	0x00200000	0x01
0x16	0x00000020	0x00000000	0x00
0x16	0x00000020	0x00000020	0x01
0x20	0x00000060	0x00000000	0x00
0x20	0x00000060	0x00000020	0x01
0x20	0x00000060	0x00000040	0x02
0x14	0x00000040	0x00000000	0x00
0x14	0x00000040	0x00000040	0x01
0x0b	0x00000080	0x00000000	0x00
0x0b	0x00000080	0x00000080	0x01
0x09	0x00000100	0x00000000	0x00
0x09	0x00000100	0x00000100	0x01
0x1e	0x00000080	0x00000000	0x01
0x1e	0x00000080	0x00000080	0x02
0x1c	0x00000100	0x00000000	0x01
0x1c	0x00000100	0x00000100	0x02
0x18	0x00000200	0x00000000	0x00
0x18	0x00000200	0x00000200	0x01
0x22	0x01800000	0x00000000	0x00
0x22	0x01800000	0x00800000	0x01
0x22	0x01800000	0x01000000	0x02

5.6.4 User Interface

By utilizing the DIOW, DIOR commands it is possible to control the behavior of the DIO section of the VCM

Table 11 List of DIO control registers

No	Width	Register	Details	Type	Default
00	16	C_DINHW	Hardware data input	R	N/A
01	32	C_DINSW	Software data input	R/W	0x00000000
02	32	C_DINSEL	Input signals selected by HW or by SW	R/W	0x000007FF
03	32	C_DIN	Composite data input	R	N/A
04	16	C_TCFG	Select a table (configuration)	R/W	0

05	16	C_LOUTSEL	LED output signals selected by HW or SW	R/W	0x000003FF
06	16	C_LOUTSW0	Software LED data output	R/W	0x00000000
07	16	C_LOUTSW1	Software LED data output	R/W	0x00000000
08	16	C_LOUTSW2	Software LED data output	R/W	0x00000000
09	16	C_LOUTSW3	Software LED data output	R/W	0x00000000
0a	16	C_LOUTSW4	Software LED data output	R/W	0x00000000
0b	16	C_LOUTSW5	Software LED data output	R/W	0x00000000
0c	16	C_LOUTSW6	Software LED data output	R/W	0x00000000
0d	16	C_LOUTSW7	Software LED data output	R/W	0x00000000
0e	16	C_LOUTSW8	Software LED data output	R/W	0x00000000
0f	16	C_LOUTSW9	Software LED data output	R/W	0x00000000
10	16	C_LOUTSW10	Software LED data output	R/W	0x00000000
11	16	C_LOUTSW11	Software LED data output	R/W	0x00000000
12	16	C_LOUT0	Composite LED data output as written to HW location 3	R	N/A
13	16	C_LOUT1	Composite LED data output as written to HW location 4	R	N/A
14	16	C_LOUT2	Composite LED data output as written to HW location 5	R	N/A
15	32	C_DOUTSEL	Digital output signals selected by HW or SW	R/W	0xFFFFFFFFF
16	32	C_DOUTSW	Software digital data output	R/W	0x00000000
17	32	C_DOUTBL	Table driven digital output	R	N/A
18	32	C_DOUT	Composite digital data output	R	N/A
19	16	C_DOUTHW0	Composite digital data output written to HW location 0	R	N/A
1a	16	C_DOUTHW0	Composite digital data output written to HW location 1	R	N/A
1b	16	C_DOUTHW0	Composite digital data output written to HW location 2	R	N/A

5.6.4.1 C_DINHW Register

The C_DINHW register contains hardware data inputs. This register contains the state of the hardware digital input signals after they have been de-bounced.

C_DINSW Register Bit Assignments															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					FPPTT	2MUTE	RX2MS	TX2MS	PTT2S	PTT2M	1MUTE	RX1MS	TXIMS	PTT1S	PTT1M
0	**0**	**0**	**0**	**0**	X	X	X	X	X	X	X	X	X	X	X

DINHW Hardware Data Input Register. Logic '1' asserts the input; logic '0' releases the input. Register is read-only.

5.6.4.2 C_DINSW Register

The DINSW register contains the software data inputs.

C_DINSW Register Bit Assignments															
31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DINSW[31:16]															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

C_DINSW Register Bit Assignments															
DINSW[15:0]															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DINSW Software Data Input Register. With proper configuration of C_DINSEL writes to this register can be used to simulate changes of hardware input states. Logic '1' asserts the input; logic '0' releases the input.

5.6.4.3 C_DINSEL Register

The C_DINSEL register configures whether input signals will be driven by the hardware or by the software data input register.

C_DINSEL Register Bit Assignments															
31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DINSEL[31:16]															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DINSEL[15:0]															
0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

DINSEL Input Source Select. A set state (1) indicates that the associated hardware input will be used to drive the event table. A reset state (0) indicates a software input will be used to drive the event table. Since the card only contains 11 hardware inputs, the most significant 6 bits represent virtual inputs. Virtual inputs should only be configured for software initiated state changes.

5.6.4.4 C_DIN Register

The C_DIN register contains the composite data input result after selecting the either the hardware or software data inputs on a bit-by-bit basis.

C_DINSW Register Bit Assignments															
31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							DIN[24:16]								
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DIN[15:0]															
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

DIN Composite Hardware Data Input Register. The value here is used to drive the control table output. It is read-only.

5.6.4.5 C_TCFG Register

The TABLE register selects one of three tables that can be used to drive output events.

C_TABLE Register Bit Assignments															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Table															
0															

TABLE Table Select. The register is used to select one of three tables that drive digital outputs, LED outputs, and autonomous event messages. The default table is 0.

5.6.4.6 C_LOUTSEL Register

The C_LOUTSEL register configures the input source (table or software) that will be driving the LEDs.

C_LOUTSEL Register Bit Assignments															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				LED11	LED10	LED9	LED8	LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
0	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

LOUTSEL LED Source Select. A set state (1) indicates that the associated table driven result will be used to control the LED output signal. A reset state (0) indicates that the software data output register will be used to control the LED output signal.

5.6.4.7 C_LOUTSW(x) Register

The C_LOUTSW(x) register contains the software LED output states.

C_LOUTSEL Register Bit Assignments															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LED State															
0															

LEDx LED Software State will be used to control the LED output signal if the corresponding LED's state of C_LOUTSEL is enabled for software. The following states are valid.

0 Off
1 Green – Solid
2 Red – Solid
3 Green – Flash
4 Red - Flash

5.6.4.8 C_LOUT(x) Register

The C_LOUT(x) registers contains the actual values written to the internal LED control registers.

C_LOUT(x) Register Bit Assignments															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
C_LOUT(x)[15..0]															
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

C_LOUT(x) This register is read only. This register is only useful for software debug purposes. C_LOUT0 is written to internal location 3, C_LOUT1 2, C_LOUT2 3.

5.6.4.9 C_DOUTSEL Register

The C_DOUTSEL register configures the source (table or software) that will be used to set the C_DOUT register.

C_DOUTSEL Register Bit Assignments															
31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved								RX2MUTEC	PTT2M	RX1MUTEC	PTT2S	RSVD	LOCKOUT	RSVD	SPARE1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPARE3	PTT1C	SPARE2	PTT2C	TX2MS	SQBR1C	RX2MS	SQBR2C	TX1SC	TX1MC	TX2SC	TX2MC	RX1SC	RX1MC	RX2SC	RX2MC
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

DOUTSEL Output Source Select. A set state (1) indicates that the associated table driven result will be used to control the hardware output signal. A reset state (0) indicates that the software data output register will be used to control the hardware output signal.

5.6.4.10 C_DOUTSW Register

The C_DOUTSW register contains software data output bits. When a DOUTSEL register bit is configured to use the software data output bit, the contents of the associated bit below will be used to drive the output signal.

C_DOUTSW Register Bit Assignments															
31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved								RX2MUTEC	PTT2M	RX1MUTEC	PTT2S	RSVD	LOCKOUT	RSVD	SPARE1
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPARE3	PTT1C	SPARE2	PTT2C	TX2MS	SQBR1C	RX2MS	SQBR2C	TX1SC	TX1MC	TX2SC	TX2MC	RX1SC	RX1MC	RX2SC	RX2MC
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DOUTSW Software Data Output Register. The host can write bit values into this register to stimulate output signals regardless of the table output value. Logic '1' asserts the input; logic '0' releases the input.

5.6.4.11 C_ DOUTBL Register

The C_DOUTBL register contains the table driven output result that can be used to drive digital output signals.

C_DOUTBL Register Bit Assignments															
31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved								RX2MUTEC	PTT2M	RX1MUTEC	PTT2S	RSVD	LOCKOUT	RSVD	SPARE1
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPARE3	PTT1C	SPARE2	PTT2C	TX2MS	SQBR1C	RX2MS	SQBR2C	TX1SC	TX1MC	TX2SC	TX2MC	RX1SC	RX1MC	RX2SC	RX2MC
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

DOUTBL Table Data Output Register. This value is read-only. The values are driven by the table algorithms. Logic '1' asserts the output; logic '0' releases the output. It is read-only.

5.6.4.12 C_DOUT Register

The C_DOUT register contains composite data output value that results from the combination of table and software data outputs.

C_DOUT Register Bit Assignments															
31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved								RX2MUTEC	PTT2M	RX1MUTEC	PTT2S	RSVD	LOCKOUT	RSVD	SPARE1
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPARE3	PTT1C	SPARE2	PTT2C	TX2MS	SQBR1C	RX2MS	SQBR2C	TX1SC	TX1MC	TX2SC	TX2MC	RX1SC	RX1MC	RX2SC	RX2MC
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

DOUT Software Data Output Register. This value is read-only. This is a composite output used to drive the hardware. Logic '1' asserts the output; logic '0' releases the output. It is read-only.