

CFSound-IV

Compact FLASH

Digital Audio

System

Revision 1.2.12

Monday, May 07, 2018



Copyright(c)1992-2018 by Arbitrary Precision, Sarasota, FL ALL RIGHTS RESERVED

Table of Contents

Overview	1
Features.....	1
Connections	2
Main Connector	2
<i>Power Supply Connection</i>	<i>2</i>
<i>Built-in Contacts Connection</i>	<i>2</i>
<i>Speaker Connection.....</i>	<i>3</i>
Serial Connector	4
<i>DB-9P Signals</i>	<i>4</i>
JB1 Serial Configuration Jumpers.....	4
<i>RS-232 Connection to PC.....</i>	<i>5</i>
RS-232 CFSound (JB1 jumpered as DCE) to PC (DTE)	5
RS-232 CFSound (JB1 jumpered as DTE) to PC (DTE)	6
<i>RS-485 Connection.....</i>	<i>6</i>
Ethernet Connector.....	7
Micro USB Connector.....	7
LINE OUT Connector	7
LINE IN Connector	7
PHONES Connector	8
Remote Volume Connector	8
SD Card Connector.....	8
Quick Start	9
Troubleshooting.....	12
No sound, no Green or Red LED(s) flashing	12
No sound, Red LED flashing	12
Low sound volume	12
Programming the Secure Digital FLASH Card	13
SD/SD-HC Flash Card Requirements.....	13
Installing and Removing the Secure Digital FLASH Card	13
Audio Rates Supported	13
Formatting the SD Card.....	14
Configuration Settings	15
Configuration Settings (BETA software)	17
How the Communication Works.....	18
Serial Flow Control	19
RS-485 Driver Enable.....	19
Network Support	20
MAC Address	20
IP Address.....	21
IP Mask.....	22

Router IP Address	22
FTP Server Port	23
VNC Server Port.....	24
HTTP Server Port	26
TCP/IP Raw Port	26
NTP Client	27
NTP Server	27
NTP Client Port	27
Local Timezone	27
Daylight Savings.....	27
Art-Net™	28
Art-Net IP Port	28
Modes of Operation	29
CFSound Mode	30
Filename Formats	30
<i>Sample Filenames.....</i>	<i>32</i>
<i>File Naming Format for CFSound style operation.....</i>	<i>32</i>
<i>Secure Digital FLASH Card Loading and Scanning.....</i>	<i>32</i>
<i>Background Audio Tracks</i>	<i>33</i>
<i>CFSOUND.INI File.....</i>	<i>34</i>
Sample CFSOUND.INI files	37
<i>Quiz / Kiosk Mode</i>	<i>38</i>
<i>Sequential Sound Play-out</i>	<i>39</i>
<i>RS-232 Serial Port Protocol.....</i>	<i>41</i>
Play / Queue Sound Command	42
Volume Command	43
Amplifier Command	44
Line Input Command	44
DMX Operation.....	45
<i>DMX Controller (Master).....</i>	<i>45</i>
<i>DMX Device (Slave).....</i>	<i>45</i>
Basic Mode	48
Diagnostic LED Codes (Red & Green LEDs).....	49
Green LED	49
Red LED Error Flashes	49
Power Requirements	50
Digital Audio Recording	50
Recommended Recording Procedure	50
Self-Recording Procedure.....	50
Optional Plug-in Modules.....	52
CFSound-IV Contact Sense 8 Module.....	52
CFSound-IV Contact I/O 8 Module.....	52
CFSound-IV Contact Sense 24 Module.....	54
Contact Wiring.....	55

Connecting a Motion Sensor	56
Normal Sized Motion Sensor.....	56
Small Sized Motion Sensor.....	57
Connecting Multiple Speakers	58
Controlling Eight LEDs with a Sound Sequence	59
Light LED while Sound is Playing	60
Frequently Asked Questions	61
Sample Messages	65
Sample Power Up Message, no SD card.....	65
Sample Power Up Message, CFSound Mode	65
Sample Power Up Message, Basic Mode	66
Mechanical.....	67
CFSound-IV Beta Software Changes	68
Fixes and Corrections	68
Improvements	68
New Features	68
Firmware Revisions	69
Upgrading the Firmware.....	70

Overview

Thank you for purchasing the **Compact FLASH IV Digital Audio System**; be assured that it will provide you with thousands of hours of trouble free, solid-state operation. You will also find that it is simple to change audio themes or tracks and is easy to make your own recordings or “program” the device.

End users can change the audio themes or tracks simply by inserting another, different Secure Digital FLASH (SD/SDHC) Card, as easily as the changing of a cassette tape. Multiple cards can be programmed with different audio themes or tracks allowing daily, hourly, weekly or monthly changes. It truly is as simple as removing the old card and plugging in the new one. Depending on the manufacturer, SD Flash Cards may be re-used up to 100,000 times.

Sound files are easily copied onto the card via your PC or laptop, using a SD Flash to USB adapter. As the files are copied to the SD Flash Card, they are renamed. This process instructs the **Compact FLASH IV Digital Audio System** to associate sounds with contacts and control how and when they are played.

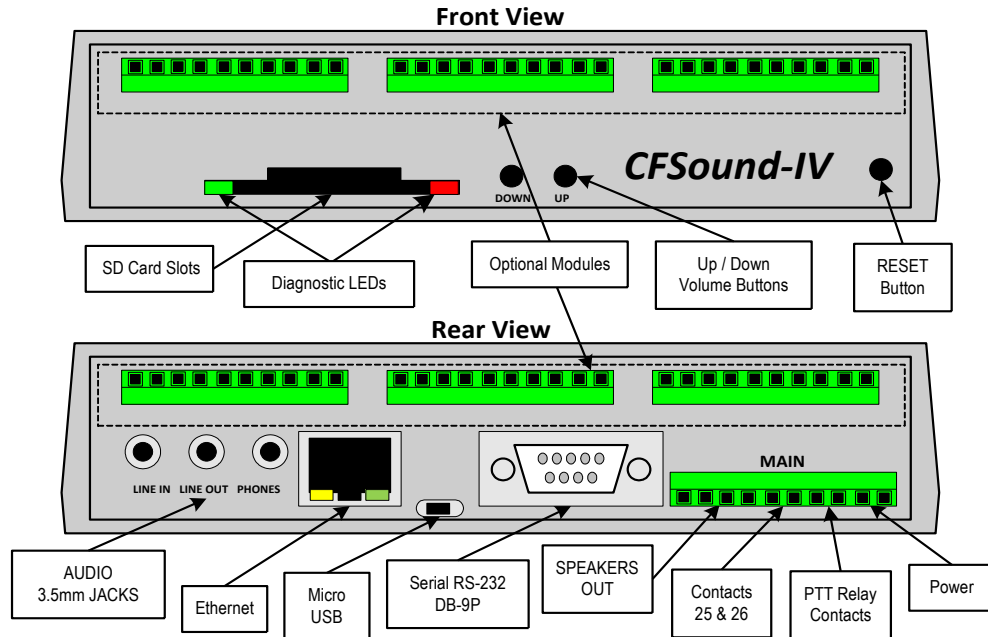
See the **Quick Start** section below to try out your CFSound-IV immediately.

Features

The Compact FLASH IV Digital Audio System incorporates the following features:

- Uses inexpensive, industry standard Secure Digital FLASH (SD/SDHC) Cards.
- Runs on 12 – 15VDC with supplied 120 – 240VAC 50/60Hz wall transformer
- RS-232 Serial Port for controlling audio play out via an attached computer or PLC.
- USB port for connection to PC as a Flash Drive or Serial device.
- Diagnostic LED's to indicate operating status.
- Two built-in contact inputs to activate sounds.
- Push-To-Talk (PTT) dry relay contact output that can optionally close whenever a sound is played.
- Digital Up/Down volume control push buttons with remote connector.
- Built-in 20 Watt Class D Stereo (2 x 10W) Amplifier.
- Built-in 35mW @ 32 ohms Headphone Amplifier.
- Scriptable via built-in [Basic](#)
- Ethernet connection with programmable configuration and multiple protocol support: DHCP client, FTP server, VNC server, HTTP, TCP/IP Raw, NTP client, SMTP client (via Basic) and Art-Net™.
- Optional boards for additional contact inputs to activate sounds.
- Optional boards for contact outputs activated with sounds for other control.
- Optional Power Over Ethernet (PoE) operation.
- Optional RS-485 operation.

Connections



Main Connector

Most of the connections to the CFSound-IV are supplied to the unit with a ten pin, 2-piece pluggable terminal block style connector at **Main**.

Power Supply Connection

The CFSound-IV requires a regulated or unregulated voltage of 10V to 18V DC.

Voltages above 18VDC will damage the unit.

The unit has resettable fusing internally to protect the Power input. Should the fuse trip, remove power, correct the fault, wait a minute for the thermal fuse to cool and reset, then, re-apply power. The non-OEM version of the CFSound-IV is supplied with a 120-240VAC 50/60Hz wall mounted transformer rated at 15VDC @ 1.6A.

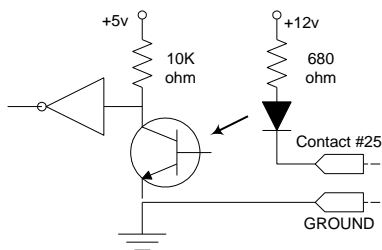
Arbitrary Precision supplies different versions of the wall-mounted transformer power supply for the CFSound-IV. The designated positive lead which should be connected to pin 9 of the **Main** connector with the other negative or ground lead connected to pin 10.

Built-in Contacts Connection

The Push-To-Talk (PTT) relay dry contact output on the **Main** connector may be activated whenever a sound is playing. The contact output is implemented with a relay whose contacts are **rated at 1A @ 30VDC maximum**. The Common and Normally Open relay contacts are available on the **Main** connector.

The contact inputs on the **Main** connector appear logically to the CFSound-IV as Contact #25 & #26, and may be activated by momentarily applying ground to the Input Contact #xx pin on the **Main** connector. The contact inputs are implemented with an optocoupler. The Cathodes of the LEDs in the optocoupler are connected to the Contact #25 and #26 Input pins. The Anodes of the LEDs in the optocoupler are connected to an internal 12VDC power supply, with a 680 ohm current limiting resistor in series. The output transistor of the optocouplers have 10K pull-up resistors on their collectors, with the emitters connected to ground and

are buffered by an inverting gate. The following diagram is representative of the Contact #25 input and is similar for Contact #26:

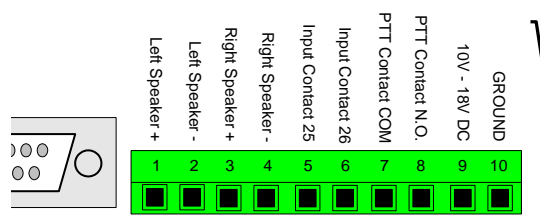


The input is activated by sinking current from the Contact input pin to ground. A Ground connection is supplied on pin 10 of the **Main** connector for this purpose. The input current sink requirement is approximately 10mA. The contacts are associated with sound files named 19x.WAV and 1Ax.WAV. The x denotes additional file control attribute characters. See the **Filename Formats** section below.

Speaker Connection

The speaker terminals on the **Main** connector provide connections for the two external left and right speakers. These may be either 4 or 8 ohm speakers, with 4 ohms providing higher output power. The speakers should be capable of handling up to 25 watts.

NOTE: THE SPEAKER OUTPUTS ARE NOT REFERENCED TO GROUND – EACH SPEAKER REQUIRES ITS OWN PAIR OF WIRES, AND NEITHER WIRE MAY BE CONNECTED TO GROUND.



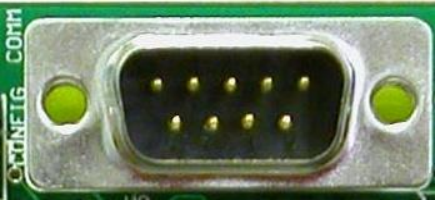
Pin #	Signal	Filename
1	Left Speaker +	
2	Left Speaker -	
3	Right Speaker +	
4	Right Speaker -	
5	Input Contact 25	19C.WAV / 19O.WAV
6	Input Contact 26	1AC.WAV / 1AO.WAV
7	PTT Contact COM	
8	PTT Contact N.O.	
9	10VDC – 18VDC	
10	Ground	

Serial Connector

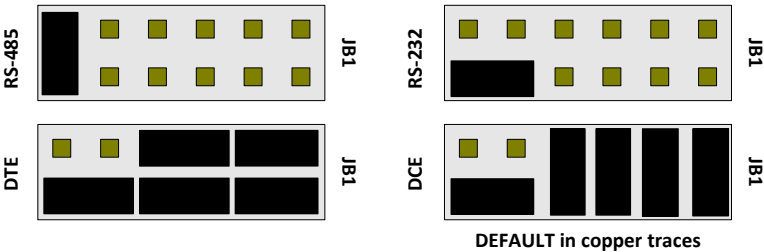
The CFSound-IV can function as either a RS-232 DTE (Data Terminal Equipment) or RS-232 DCE (Data Communications Equipment) device. There is a jumper block site on the board labeled JB1 that can be used to reverse the pins of the signals pairs TxD, RxD and RTS, CTS on the SERIAL connector to accomplish this change.

DB-9P Signals

Connection to the SERIAL connector requires a female DB-9S mating connector. The following signals appear on the male SERIAL DB-9P connector:

 <i>SERIAL DB-9P Connector</i>	PIN	DCE Signal	JB1 as DCE Direction	DTE Signal	JB1 as DTE Direction
	1	RS-485 B-	I/O	RS-485 B-	I/O
	2	RS-232 TxD	OUT	RS-232 RxD	IN
	3	RS-232 RxD	IN	RS-232 TxD	OUT
	4				
	5	GND	PWR	GND	PWR
	6	RS-485 A+	I/O	RS-485 A+	I/O
	7	RS-232 CTS	IN	RS-232 RTS	OUT
	8	RS-232 RTS	OUT	RS-232 CTS	IN
	9	+12-15VDC	PWR	+12-15VDC	PWR

JB1 Serial Configuration Jumpers



The board is configured as RS-232 DCE to allow use of a 1 to 1 cable between the CFSound and a PC. This configuration is established by copper jumpers on the bottom of the board between the JB1 pins.

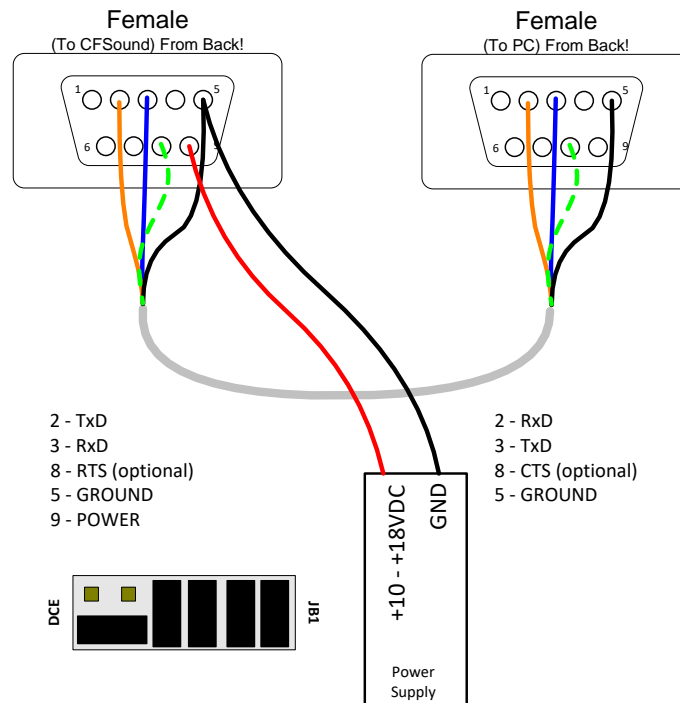
RS-232 Connection to PC

The PC is configured as a DTE device. A one-to-one, female DB-9S to DB-9S cable can be used to connect the PC communication port to the CFSound-IV SERIAL connector when the CFSound is configured as DCE. Only four wired connections are required.

The RTS connection is for optional flow control of the transmit data to the CFSound to prevent overflowing the input buffer at higher baud rates. It requires disabling the RS-485 enable in the CFSound configuration and implementation of flow control on the host device supplying the data to the CFSound.

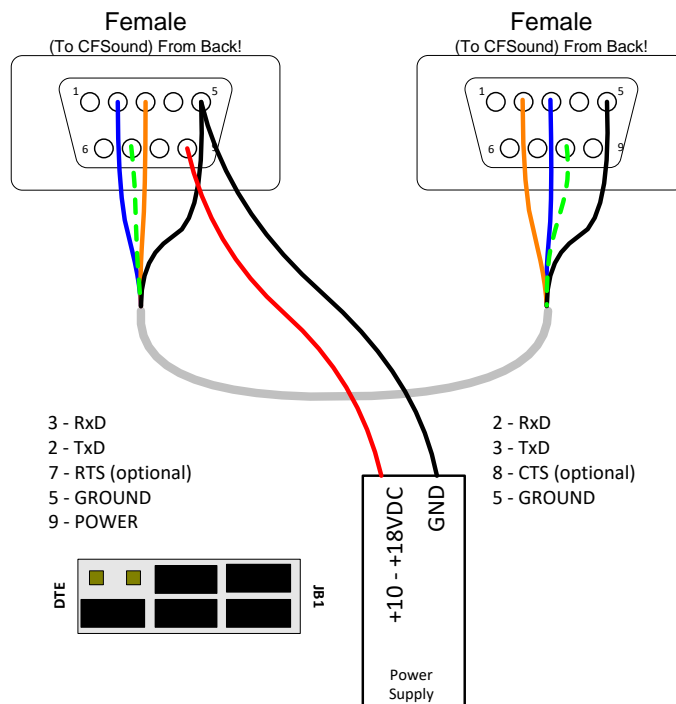
RS-232 CFSound (JB1 jumpered as DCE) to PC (DTE)

The CFSound-IV can be connected with a one-to-one cable between the CFSound and host PC when JB1 is configured for DCE (default configuration in copper traces on bottom of the board beneath JB1):



RS-232 CFSound (JB1 jumpered as DTE) to PC (DTE)

The CFSound-IV can be connected with a null modem cable (or one-to-one cable with null modem adaptor) between the CFSound and host PC when JB1 is configured for DTE:



RS-485 Connection

The CFSound-IV can be used with half-duplex RS-485 serial connections when the factory installed RS-485 option is present. The RS-485 Enable configuration item must be turned on to control when the driver is enabled on the bus:

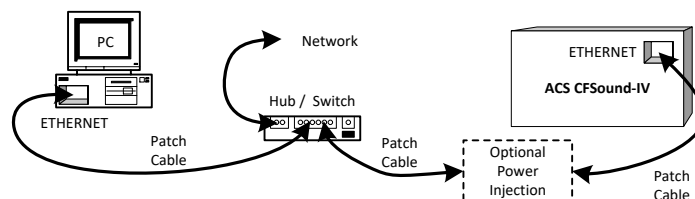
SERIAL Pin #	Power Supply	CFSound SIGNAL
6		A+
1		B-
5	GND	GND
9	+10 → +18VDC	+10 → +18VDC

NOTE: Be sure to connect both the Ground of the PC or Host computer and the Ground of your +10 → +18VDC Power supply together!

The RS-485 A+ and B- signals are terminated with a 100 ohm resistor between them. In addition, there is a 10K pull-up resistor on the A+ signal to +3.3v and a 10K pull-down resistor on the B- signal to Ground, to put the received data line in an idle state when there is no connection.

Ethernet Connector

The CFSound-IV can be connected as an Ethernet device. A standard RJ-45 connector is provided and it can be connected to a network with a standard Ethernet cable – either straight or crossover, detection and correction is automatic via HP Auto MDI/MDI-X configuration. The network speed can be either 10 or 100 mbps with auto link negotiation. A link activity indicator is provided on the ETHERNET jack.



The CFSound-IV supports a configurable MAC address and configurable static or dynamic IP address and IP mask. A variety of protocols are supported with configurable port numbers.

Micro USB Connector

The CFSound-IV can be connected as a USB peripheral device. A standard micro USB B receptacle connector is provided and it can be connected to a PC using a standard USB-A to micro USB B cable. The CFSound-IV can be configured to function as either a USB flash drive or USB serial device. When configured as a flash drive (the default) the CFSound-IV SD card appears as a USB drive on the host PC and supports file transfers. When configured as a serial device the CFSound-IV sends and receives serial data as if connected to the RS-232 port.

Pin #	Signal
1	VBUS
2	D-
3	D+
4	ID
5	GND

LINE OUT Connector

The volume controlled sound is also available at an audio line level (1V RMS @ 0db volume, 47K ohms) at the 3.5mm Stereo LINE OUT jack on the rear of the CFSound-IV.

Pin #	Signal
TIP	Left
RING	Right
SLEEVE	Ground

LINE IN Connector

Line level audio appearing at the 3.5mm Stereo LINE IN jack (1V RMS @ 0db volume, 47K ohms) on the rear of the CFSound-IV may be enabled to be amplified at the current volume setting and presented to the Speakers/Line Outputs whenever another sound is not playing. See the **CFSOUND.INI** section below.

PHONES Connector

The volume controlled sound is also available for driving stereo headphones (35mW @ 0db volume, 32 ohms) at the 3.5mm Stereo PHONES jack on the rear of the CFSound-IV.

NOTE: The common headphone output is not referenced to, and should not be connected to Ground.

Pin #	Signal
TIP	Left
RING	Right
SLEEVE	Common

Remote Volume Connector

The sound volume level is controlled by two pushbuttons on the front of the CFSound-IV: one for Up and one for Down. A momentary button press of the Up button raises the volume level, a button press of the Down button lowers the volume level. Pressing and holding the button for ¼ second starts an automatic volume increase/decrease until the button is released or the maximum/minimum volume is achieved. The volume is essentially db linear in sixty-four, 1db increments from 0db → -66db. The current volume level is ‘remembered’ in non-volatile memory on the CFSound-IV and is restored to its last setting upon power-up.

The Up and Down button inputs are also available on the VOL connector along the front edge of the printed circuit board inside the enclosure. Connecting a momentary button between the input and ground parallels the operation of the on-board volume controls:

Pin #	Signal
1	DOWN-
2	GND
3	UP-

SD Card Connector

There are two slots for a SD and microSD memory card front of the CFSound-IV.

NOTE: Only one SD card slot may be in use at a time – simultaneous use of both a SD and microSD memory card is NOT SUPPORTED.

All card accesses are performed in SPI mode using the following signals:

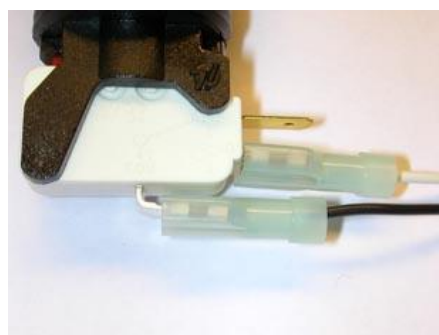
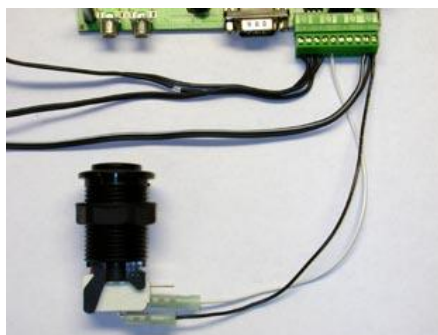
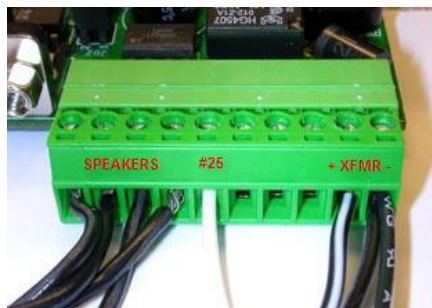
uSD Pin #	Signal Name	Description
1	DAT2 / NC	No connection
2	CD / DAT3 / CS-	Chip Select (active low)
3	CMD / DI	Data Input
4	VDD	Power (+3.3v)
5	CLK	Clock
6	VSS	Ground
7	DAT0 / DO	Data Output
8	DAT1 / RSV	No connection

SD or microSD cards of Standard Capacity (SDSC or SD, 1MB to 1GB) and High Capacity (SDHC, 2GB to 32GB) are supported.

Quick Start

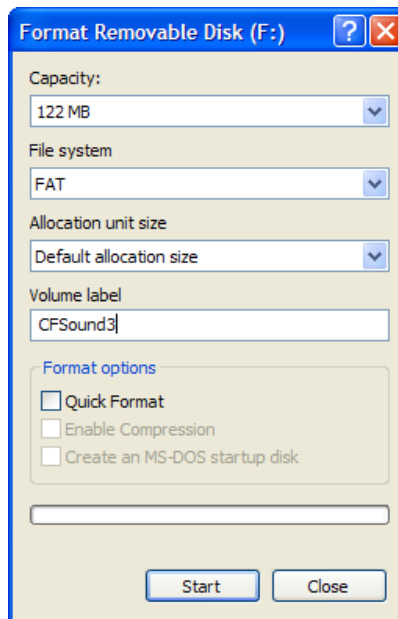
Also refer to the videos on the [SUPPORT](#) page of the [CFSOUND.COM](#) website.

1. Connect the transformer, speakers and a normally open momentary pushbutton switch to the Main terminal block. See **Main Connector** section above for more information.



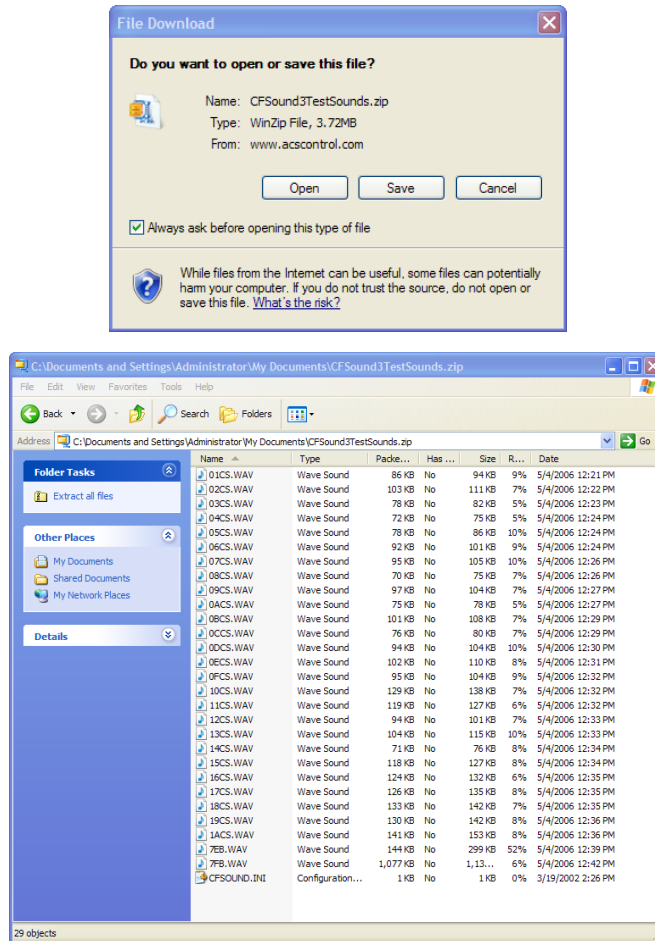
2. Apply power to the CFSound-IV. With no SD card inserted, the Green LED should flash fast then slow. See **Diagnostic LED Codes** section below for more information.

3. Format a SD Flash card using either FAT16 (FAT) or FAT32:

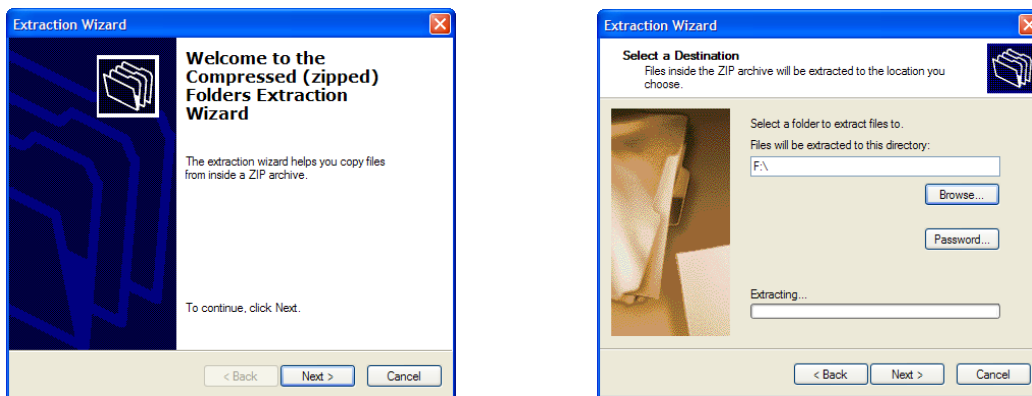


4. Download the sample sounds from the **CFSound** website:

<http://www.acscontrol.com/Downloads/Products/CF/CFSoundIII/CFSound3TestSounds.zip>



5. Extract the test sounds onto the formatted SD Flash card. Browse to the SD drive:



6. Carefully insert the SD card into the CFSound-IV. The Green LED should flash fast then go on steady and the background sounds should start playing then repeating.
7. Press the button. The CFSound-IV should interrupt the background sound, play sound #25, then resume playing the background sound.

Troubleshooting

No sound, no Green or Red LED(s) flashing

The CFSound-IV doesn't have correct power. Verify the power supplied to the MAIN connector for the correct polarity and level using a voltmeter if available. If this is correct, verify that none of the wiring to the unit is shorted causing the internal resettable fuse to be tripped. Try removing power, waiting for 15 seconds and reapplying power to allow the internal fuse to reset.

No sound, Red LED flashing

The Red LED is flashing a pattern of one or more numbers to help identify the problem. Please see the **Diagnostic LED Codes** and **Frequently Asked Questions** sections below. Also refer to the videos on the [SUPPORT](#) page of the CFSOUND.COM website.

Low sound volume

Sound volume may be adjusted using the up / down pushbuttons on the front of the unit. Push and hold the up button to increase. If this doesn't adjust the volume, verify your speaker wiring and impedance.

Programming the Secure Digital FLASH Card

SD/SD-HC Flash Card Requirements

The **Compact FLASH IV Digital Audio System** was designed to use only Secure Digital FLASH Cards (SD or SD/Hc Cards) that support the Serial Peripheral Interface mode at 3.3 volt operation. SD-HC cards up to 32GB may be used. Tested cards include SanDisk, Samsung, Dane-Elec, Memorex and Hitachi. The SanDisk card was tested, since they manufacture 90% of the retail cards on the market. Secure Digital FLASH Cards are available in many local electronics or camera stores or can be ordered directly from Arbitrary Precision.

The Secure Digital FLASH Card is programmed by inserting it into a SD Card FLASH adapter connected to your computer's USB port. Sound files are copied to the SD FLASH Card just as if it were a disk drive. The files used are the standard "*.WAV" files typically employed by Windows and DOS audio applications for years.

When a new SD FLASH Card is first installed into your computer, Windows (95 or later) should recognize the device and install the software necessary to access the card. After the software installation is complete, the SD FLASH Card appears as a disk drive icon under My Computer. You may then use Windows Explorer to drag and drop the sound files from your computer's hard disk directly to the SD FLASH Card drive icon. It really is just that easy.

Please take care not to use audio tracks that are copyrighted. If you wish to avoid this issue, you will find that many computer software outlets have CD's with non-copyrighted music and audio sound effects that you may use without limitation. They are commonly found in the multimedia section.

Please be advised that you as the user are solely responsible for audio that you record and play from the Compact FLASH IV Digital Audio System.

Installing and Removing the Secure Digital FLASH Card

Care should be taken when installing or removing the SD FLASH Card. Please be certain to insert the card with a straight and level motion. Please do not force the card to insert. Otherwise the SD card socket may become damaged and the card and/or **Compact FLASH IV Digital Audio System** may become unusable.

Before REMOVING the SD FLASH Card from your PC under Windows 95 or later, Microsoft recommends that you double click on the Removable Drive icon at the bottom of your computer screen. This will bring up the Removable Drive properties window. Click on the card you wish to remove by clicking on it. Finally, click on the stop button. Windows will then tell you that the SD card can be safely removed. Under Windows XP and later OSs you may also right-click on the folder and select Eject.

Failure to stop the card reader and/or eject the SD card after writing to it may result in incorrect or incomplete data on the card.

On the **Compact FLASH IV Digital Audio System** this will all happen automatically simply by removing the card.

Audio Rates Supported

The Compact FLASH IV Digital Audio System supports 8 or 16-bit mono or stereo Windows PCM format (.WAV) sound files at sampling rates of 8KHz, 11.025KHz, 16KHz, 22.050KHz, 32KHz and 44.1KHz only. The file must be recorded in the PCM WAV format – simply renaming a MP3 file will not work.

The **Compact FLASH IV Digital Audio System** reads the sampling rates and formats encoded in the .WAV sound files whenever a card is first inserted or a sound play-out is triggered. The unit will not play any

file that does not meet these requirements. Further, this error condition should be identified by the red led flashing twice every six seconds. See the [Diagnostic LED Codes](#) section below for more information.

Formatting the SD Card

Secure Digital FLASH cards **must only be formatted** using the FAT16 or FAT32 file systems. Most SD cards come correctly pre-formatted for their use in digital cameras and other devices.

SD cards must be formatted as FAT16 or FAT32 file systems.

SD cards formatted as NTFS, Linux EXT3 or other formats are not supported.

Some digital cameras and cellphones incorrectly format their SD cards.

The maximum supported partition size is 32G. The CFSound-IV will attempt to locate a valid, active partition on the formatted card in order to be able to access the files. SD cards that have been previously used in some digital cameras may not be correctly formatted, or may not have valid partitions defined.

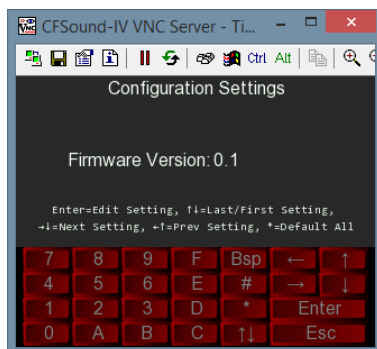
By default, Windows XP will format any Secure Digital card of 64MB or more with FAT32 format. The Arbitrary Precision Compact FLASH IV Digital Audio System uses the FAT (FAT16 or FAT32) format and cannot operate with a NTFS formatted card. You must select FAT or FAT32 file system to format your Secure Digital card in a Windows XP PC.

Configuration Settings

#	Setting	Description
0	Firmware Version	= version # of firmware
1	Serial Baud Rate	110, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600 (default), 14400, 19200, 28800, 38400, 57600, 115200
2	Serial Data Bits	7, 8 (default)
3	Serial Stop Bits	1 (default), 2
4	Serial Parity	NONE (default), EVEN, ODD
5	RS485 Mode	None (default), Primary, Secondary
6	USB Mode	SD Card (default), Serial port
7	TPad Show Up Left-X	= Touch Keypad Show upper left X coordinate (0 – 319) (default=270)
8	TPad Show Up Left-Y	= Touch Keypad Show upper left Y coordinate (0 – 239) (default=0)
9	TPad Show Lo Right-X	= Touch Keypad Show upper left x coordinate (0 – 319) (default=319)
10	TPad Show Lo Right-Y	= Touch Keypad Show upper left x coordinate (0 – 239) (default=48)
11	TPad Style	= NONE, QWERTY (default), NUMERIC, QWERTY TOP, NUMERIC TOP
12	TPad Scheme	= RED (default), GREEN, BLUE, GRAY 25%, GRAY 50%
13	MAC Address	= 02:01:23:45:67:89 (default)
14	Use Static IP	= 0-DHCP client, 1-static IP (default)
15	IP Address	= 192.168.1.200 (default)
16	IP Mask	= 255.255.255.0 (default)
17	Router IP Address	= 192.168.1.1 (default)
18	FTP Server Port	= 21 (default)
19	VNC Server Port	= 5900 (default)
20	HTTP Server Port	= 80 (default)
21	TCP/IP Raw Port	= 23 (default)
22	NTP Server	= 192.168.1.1 (default)
23	NTP Client Port	= 123 (default)
24	Local Time Zone	= -5 hours from UTC (default)
25	Daylight Savings	0 = no (default) adds one hour to received NTP time
26	Serial Number	00000000 (default)
27	Volume	Last volume setting
28	Art-Net IP Port	6454 (default)
29	Art-Net Net	0 (default) – 127
30	Art-Net Sub-Net	0 (default) – 15
31	Art-Net Universe	0 (default) – 15
32	Art-Net Target IP	255.255.255.255 (default)
33	End of Config	= end of configuration items placeholder
34	NV Status Byte	= NV status byte (default = 227)

Changes to some configuration items require the CFSound-IV to be reset in order for the change to take effect.

The configuration settings may be changed in three ways; a VNC connection to the CFSound-IV's IP address with the DEFAULT jumper installed, via an Basic program or via the built-in HTTP server.



The Basic configuration program is shown as an example in the CFSound-IV Basic Programming Manual.

The built-in HTTP configuration server is accessed using a web browser to access the CFSound-IV's URL: <http://192.168.1.200/config.cgi>. The following built-in configuration page is displayed:

#	Name	Value	
0	Firmware Version	0.1	<input type="button" value="Update"/>
1	Serial Baud Rate	9600	<input type="button" value="Update"/>
2	Serial Data Bits	<input checked="" type="radio"/> 7 <input checked="" type="radio"/> 8	<input type="button" value="Update"/>
3	Serial Stop Bits	<input checked="" type="radio"/> 1 <input type="radio"/> 2	<input type="button" value="Update"/>
4	Serial Parity	NONE	<input type="button" value="Update"/>
5	RS-485 Mode	NONE	<input type="button" value="Update"/>
6	USB Mode	SD CARD	<input type="button" value="Update"/>
7	TPad Pop Up Left X	270	<input type="button" value="Update"/>
8	TPad Pop Up Left Y	0	<input type="button" value="Update"/>
9	TPad Pop Lo Right X	320	<input type="button" value="Update"/>
10	TPad Pop Lo Right Y	48	<input type="button" value="Update"/>
11	TPad Style	QWERTY	<input type="button" value="Update"/>
12	TPad Scheme	RED	<input type="button" value="Update"/>
13	MAC Address	02 . 01 . 23 . 45 . 67 . 8A	<input type="button" value="Update"/>
14	Use Static IP	<input type="radio"/> No <input checked="" type="radio"/> Yes	<input type="button" value="Update"/>
15	IP Address	192 . 168 . 001 . 200	<input type="button" value="Update"/>
16	IP Mask	255 . 255 . 255 . 000	<input type="button" value="Update"/>
17	Router IP Address	192 . 168 . 001 . 001	<input type="button" value="Update"/>
18	FTP Server Port	21	<input type="button" value="Update"/>
19	VNC Server Port	5900	<input type="button" value="Update"/>
20	HTTP Server Port	80	<input type="button" value="Update"/>
21	TCP/IP Raw Port	23	<input type="button" value="Update"/>
22	NTP Server	192 . 168 . 001 . 100	<input type="button" value="Update"/>
23	NTP Client Port	123	<input type="button" value="Update"/>
24	Local Time Zone	-5	<input type="button" value="Update"/>
25	Daylight Savings	<input type="radio"/> No <input checked="" type="radio"/> Yes	<input type="button" value="Update"/>
26	Serial Number	0 0 0 0 0 0 0 0	<input type="button" value="Update"/>
27	Volume	32	<input type="button" value="Update"/>
28	Art-Net IP Port	6454	<input type="button" value="Update"/>
29	Art-Net Net	0	<input type="button" value="Update"/>
30	Art-Net Sub-Net	0	<input type="button" value="Update"/>
31	Art-Net Universe	0	<input type="button" value="Update"/>
32	Art-Net Target IP	255 . 255 . 255 . 255	<input type="button" value="Update"/>

Configuration Settings (BETA software)

#	Setting	Description
0	Firmware Version	= version # of firmware
1	Serial Baud Rate	110, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600 (default), 14400, 19200, 28800, 38400, 57600, 115200
2	Serial Data Bits	7, 8 (default)
3	Serial Stop Bits	1 (default), 2
4	Serial Parity	NONE (default), EVEN, ODD
5	RS485 Mode	None (default), Primary, Secondary
6	USB Mode	SD Card (default), Serial port
7	TPad Show Up Left-X	= Touch Keypad Show upper left X coordinate (0 – 319) (default=270)
8	TPad Show Up Left-Y	= Touch Keypad Show upper left Y coordinate (0 – 239) (default=0)
9	TPad Show Lo Right-X	= Touch Keypad Show upper left x coordinate (0 – 319) (default=319)
10	TPad Show Lo Right-Y	= Touch Keypad Show upper left x coordinate (0 – 239) (default=48)
11	TPad Style	= NONE, QWERTY (default), NUMERIC, QWERTY TOP, NUMERIC TOP
12	TPad Scheme	= RED (default), GREEN, BLUE, GRAY 25%, GRAY 50%
13	MAC Address	= 02:01:23:45:67:89 (default)
14	Ethernet Logging	0 = no (default)
15	Use Static IP	= 0-DHCP client, 1-static IP (default)
16	IP Address	= 192.168.1.200 (default)
17	DHCP Logging	0 = no (default)
18	IP Mask	= 255.255.255.0 (default)
19	Router IP Address	= 192.168.1.1 (default)
20	FTP Server Port	= 21 (default)
21	FTP Logging	0 = no (default)
22	VNC Server Port	= 5900 (default)
23	VNC Logging	0 = no (default)
24	HTTP Server Port	= 80 (default)
25	HTTP Logging	0 = no (default)
26	TELNET Port	= 23 (default)
27	TELNET Logging	0 = no (default)
28	NTP Server	= 192.168.1.1 (default)
29	NTP Client Port	= 123 (default)
30	NTP Logging	0 = no (default)
31	Local Time Zone	= -5 hours from UTC (default)
32	Daylight Savings	0 = no (default) adds one hour to received NTP time
33	Serial Number	00000000 (default)
34	Volume	Last volume setting
35	Art-Net IP Port	6454 (default)
36	Art-Net Logging	0 = no (default)
37	Art-Net Net	0 (default) – 127
38	Art-Net Sub-Net	0 (default) – 15
39	Art-Net Universe	0 (default) – 15
40	Art-Net Target IP	255.255.255.255 (default)
41	Basic Editor Lines	= 1 – 255 (default = 25)
42	Basic Editor Cols	= 40 – 255 (default = 80)
43	End of Config	= end of configuration items placeholder
44	NV Status Byte	= NV status byte (default = 227)

Changes to some configuration items require the CFSound-IV to be reset in order for the change to take effect.

How the Communication Works

The RS-232 SERIAL connection is essentially always connected – even if no physical connection is present.

The USB connection is ‘connected’ when the cable is plugged into a host PC where the driver has been installed and the CFSound-IV is configured for USB Mode: SERIAL.

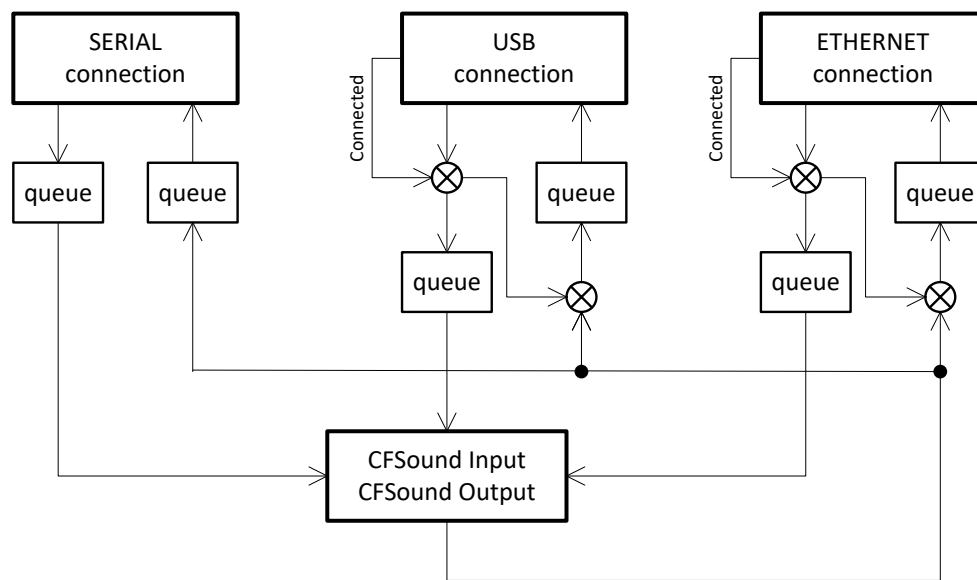
The ETHERNET connection is ‘connected’ when the CFSound-IV has been connected to the network and a TCP/IP raw sockets link has been established.

Incoming data from each of the three sources are combined asynchronously in parallel into the CFSound input buffer.

Any output that the CFSound generates is copied to all three outputs synchronously. If a connected output queue fills up, the other outputs are delayed until the full queue has space available - so the slowest connected output limits the output data rate to all of the outputs.

Note that some USB hosts will stop responding to the USB connection if it is connected and there is no application running to receive the incoming data – stopping the output data to all of the outputs.

Here’s a diagram:



Serial Flow Control

CFSound firmware provides optional support for flow control of transmit data to the player to prevent overflowing the serial input buffer at higher baud rates.

The CFSound will assert its RTS signal on power-up and whenever there is room in its input buffer.

The CFSound will de-assert its RTS signal whenever the input buffer is at 90% of its 2048 byte capacity or higher.

It will re-assert its RTS signal whenever the input buffer is below 80% capacity.

The host device that the CFSound is connected to should monitor the RTS signal, typically by connecting it to the host's CTS input, and only send transmit data to the player when the signal is asserted.

Without flow control it is possible to overrun the CFSound input buffer at baud rates above 9600 baud. Symptoms of input buffer overflow include missing or incorrectly displayed data or commands.

The following configuration settings determine the CFSound-IV's serial port operation:

#	Setting	Description
1	Serial Baud Rate	110, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600 (default), 14400, 19200, 28800, 38400, 57600, 115200
2	Serial Data Bits	7, 8 (default)
3	Serial Stop Bits	1 (default), 2
4	Serial Parity	NONE (default), EVEN, ODD

RS-485 Driver Enable

CFSound firmware provides configurable support for enabling the optional factory installed RS-485 transceiver to operate in a half-duplex mode. This allows multiple devices to exist on a multi-drop bus with devices receiving from the bus when they are not transmitting.

The CFSound will enable its transceiver driver whenever it needs to transmit data, then it will disable the driver after the data is fully transmitted.

Enabling this option in the configuration menu overrides the Serial Flow Control via RTS above as it uses the RTS signal to control the RS-485 driver enable.

The following configuration settings determine the CFSound-IV's RS-485 driver operation:

#	Setting	Description
5	RS485 Mode	None (default), Primary, Secondary

Network Support

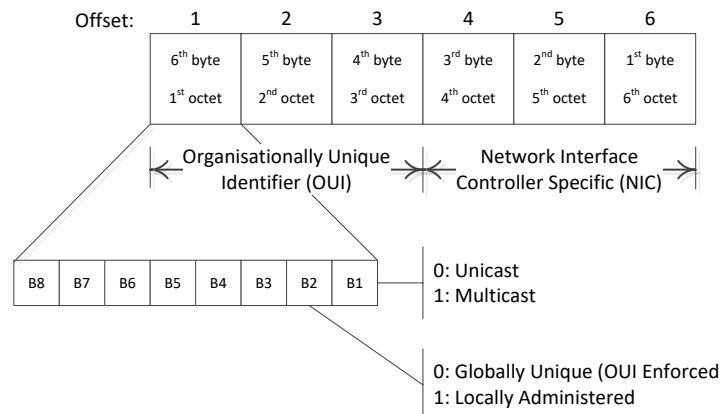
The Arbitrary Precision Compact Flash Digital Audio System supports multiple modes of access via the Ethernet connection. The networking support is controlled by several configuration parameters accessible through the Configuration Settings screen. Here are the relevant settings shown with their default values:

#	Setting	Description
13	MAC Address	= 02:01:23:45:67:89 (default)
14	Use Static IP	= 0-DHCP client, 1-static IP (default)
15	IP Address	= 192.168.1.200 (default)
16	IP Mask	= 255.255.255.0 (default)
17	Router IP Address	= 192.168.1.1 (default)
18	FTP Server Port	= 21 (default)
19	VNC Server Port	= 5900 (default)
20	HTTP Server Port	= 80 (default)
21	TCP/IP Raw Port	= 23 (default)
22	NTP Server	= 192.168.1.1 (default)
23	NTP Client Port	= 123 (default)
24	Local Time Zone	= -5 hours from UTC (default)
25	Daylight Savings	0 = no (default) adds one hour to received NTP time
...		
28	Art-Net IP Port	6454 (default)
29	Art-Net Net	0 (default) – 127
30	Art-Net Sub-Net	0 (default) – 15
31	Art-Net Universe	0 (default) – 15
32	Art-Net Target IP	255.255.255.255 (default)

MAC Address

The Media Access Control address (MAC Address) is a unique identifier assigned to network interfaces for communications on the physical network segment. Each device on a physical network segment must be unique amongst all devices on the network segment to allow network traffic to be routed to and from the correct hardware device. MAC addresses may be assigned universally across all manufactured devices or locally by the device users. The CFSound-IV supports locally administered MAC addresses – the address is assigned to the display by the user or network administrator, replacing the configured default address.

The MAC address is 48 bits long and is typically expressed as six groups of hexadecimal digits separated by colons in transmission order.



Universal and locally administered MAC addresses are distinguished by setting the second-least significant bit of the most significant byte of the address – known as the U/L bit. If the U/L bit is zero the address is universally administered, if it is one the address is locally administered.

If the least significant bit of the most significant octet is zero the network frame is meant to reach only one receiving network device (NIC). This type of transmission is called Unicast addressing – the frame is transmitted to all nodes within the collision domain, which usually ends at the nearest network switch or router. Only the network device with the matching MAC address will accept the frame.

If the least significant bit of the most significant octet is one each network device will choose to accept the frame based upon a different criteria other than a matching MAC address; maybe a configurable list of accepted MAC addresses. This type of transmission is called Multicast addressing.

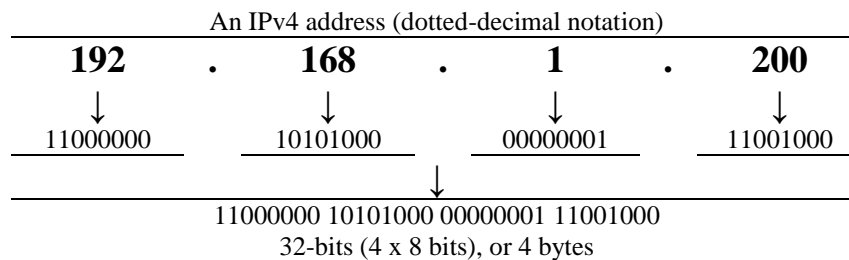
The CFSound-IV currently only supports Unicast addressing. The default MAC address is 02:01:23:45:67:89 – which configures a Unicast, Locally Administrated MAC address.

The MAC address must be configured to be unique amongst all devices on the network segment that the CFSound is attached to. As it is locally administrated it is the user's responsibility to ensure that it is a unique value.

IP Address

An Internet Protocol address (IP Address) is a numerical label assigned to each device participating in the network that uses Internet Protocol for communication. The IP Address serves two functions; host or network identification and location addressing. The CFSound-IV currently supports IPv4 protocol which requires a 32-bit IP Address number.

The IP Address is a 32-bit binary number, but it is usually expressed as 4 decimal numbers, each ranging from 0 to 255, separated by dots. Each decimal number represents a group of 8 bits (octet) of the address:



Devices that are not directly connected to the Internet but only use the network to communicate do not need globally unique IP Addresses. The Internet Assigned Numbers Authority (IANA) has defined three ranges of reserved, private IP Addresses for these devices:

IANA-reserved private IPv4 network ranges			
	Start	End	# of addresses
8-bit prefix 24-bit block	10.0.0.0	10.255.255.255	16,777,216
12-bit prefix 20-bit block	172.16.0.0	172.31.255.255	1,048,576
16-bit prefix 16-bit block	192.168.0.0	192.168.255.255	65,536

Private networks typically connect to the Internet through a Network Address Translation (NAT) device such as a router. The NAT hides a large number of IP Addresses in the private network by mapping TCP and UDP port numbers on the public network to individual addresses on the private network; the port numbers are site specific extensions to an IP Address. This allows a large number of devices on the private network to appear to have a single IP Address on the public Internet.

The CFSound-IV can utilize a static, user-configured IP address/mask or a dynamic NAT assigned IP address/mask (DHCP client).

IP Mask

A sub-network or subnet is a logically visible subdivision of an IP network. All computers that belong to the subnet are addressed with a common, identical, most-significant bit-group in their IP Address. This results in the logical division of an IP Address into two fields; a network or routing prefix and the host identifier:

Creating a subnet by dividing the host identifier		
Network Prefix	Host Identifier	
↓	↓	↓
Network Prefix	Subnet Number	Host Identifier

The process of sub-netting involves the separation of network prefix and subnet number portion of the IP Address from the host identifier. This is performed by a bitwise AND operation between the IP Address and the IP Mask. The result yields the network address or prefix and the remainder is the host identifier. Here is an example showing the separation of the network prefix and the host identifier from an IP Address (192.168.1.200) and IP Mask (255.255.255.0):

	Binary form	Dotted-decimal notation
IP Address	11000000.10101000.00000001.11001000	192.168.1.200
IP Mask	11111111.11111111.11111111.00000000	255.255.255.0
Network prefix	11000000.10101000.00000001.00000000	192.168.1.0
Host identifier	00000000.00000000.00000000.11001000	0.0.0.200

The purpose of sub-netting is to divide a network into smaller subnets. This is achieved by designating some high-order bits from the host part and grouping them with the network mask to form the IP Mask. Here the previous example is modified by moving two bits from the host part to the IP mask to form four smaller subnets one quarter the previous size:

	Binary form	Dotted-decimal notation
IP Address	11000000.10101000.00000001.11001000	192.168.1.200
IP Mask	11111111.11111111.11111111.11000000	255.255.255.192
Network prefix	11000000.10101000.00000001.11000000	192.168.1.192
Host identifier	00000000.00000000.00000000.00001000	0.0.0.8

The host identifier of all zeroes is reserved for the network ID, and the host identifier of all ones is reserved for the broadcast address, so in general the number of hosts on a subnet is $2^N - 2$ where N is the number of bits reserved for the host portion of the address. For a 24-bit prefixed network:

Network prefix size	IP Mask	Available subnets	Usable Hosts per subnet	Total usable Hosts
24-bits	255.255.255.0	1	254	254
25-bits	255.255.255.128	2	126	252
26-bits	255.255.255.192	4	62	248
27-bits	255.255.255.224	8	30	240
28-bits	255.255.255.240	16	14	224
29-bits	255.255.255.248	32	6	192
30-bits	255.255.255.252	64	2	128
31-bits	255.255.255.254	128	2 (point-to-point)	256

Router IP Address

In computer networking a router is a node that connects two networks – such as a private network and the public Internet. The host devices on the private network send IP packets addressed to IP Addresses on the private network by resolving the destination MAC address into a destination IP Address through an Address Resolution Protocol (ARP) sequence if the MAC address is not already in the host's cache. The IP packet is then encapsulated into a physical MAC frame addressed to the destination host.

IP packets addressed outside of the private network range cannot travel directly to the destination. Instead they must be sent to the router as a gateway to the other network. In order to resolve the MAC

address of the outside device the router's MAC address is used as the go-between – the outgoing packet is physically addressed to the router, and the router performs the outside network ARP to handle the physical message routing.

By definition any packet who's IP Address network prefix doesn't match the subnet network prefix of the private network needs to resolve to the MAC address of the router in order to be directed to the public network. This configuration item allows that to happen.

FTP Server Port

The CFSound-IV provides a limited File Transfer Protocol (FTP) server capability. FTP is a standard internet protocol used to transfer files between network hosts. It is built on a client/server architecture and uses separate control and data connections between the client and server.

If the configured FTP Server Port is non-zero, FTP protocol control request packets will be accepted and processed. The CFSound FTP Server responds to these requests over the control port with three-digit status codes and text in ASCII.

The CFSound-IV FTP Server only supports Passive mode and the data port that is used is always one less than the control port.

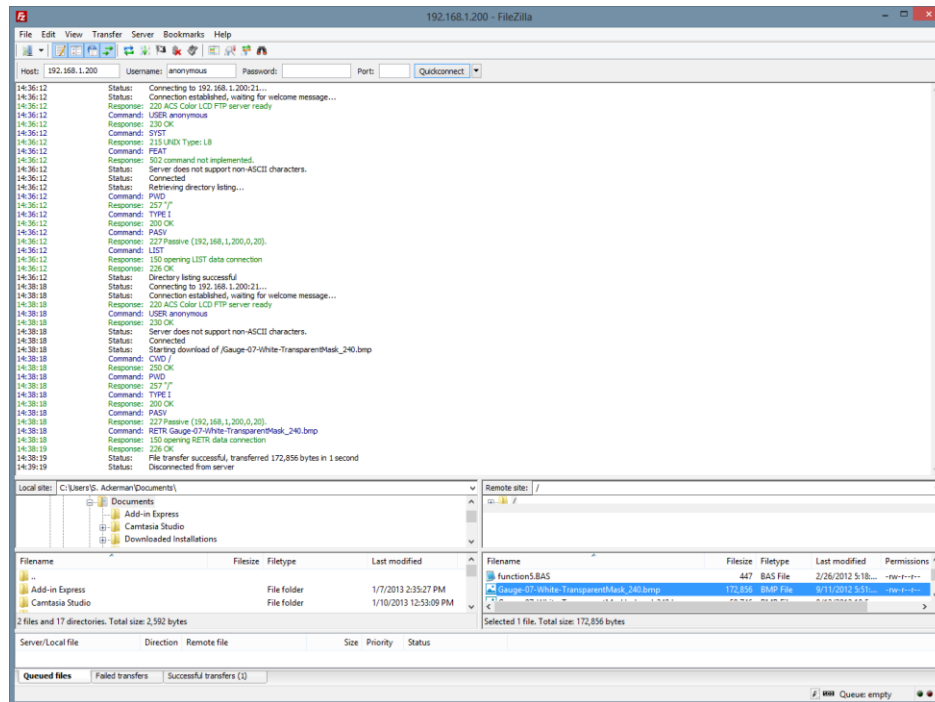
The server only supports anonymous logins with no password support.

Only one data transfer command at a time is supported.

The following FTP Commands are supported:

Command	Description	Response	Data
USER	Authentication username	230 OK	
SYST	Return system type	215 Unix Type: L8	
PWD XPWD	Print working directory	257 working directory	
TYPE	Sets the transfer mode ASCII/Binary	200 OK	
QUIT	Disconnect	221 OK	
PASV	Enter passive mode	227 Passive (IP.IP.IP.IP.port).	
SIZE	Return size of a file	502 command not implemented	
LIST NLST	Return list of files in current directory	150 opening LIST data connection 550 CFSound-IV file system error	*
RETR	Retrieve a copy of a file	150 opening RETR data connection 550 CFSound-IV file system error	*
STOR	Store a copy of a file	150 opening STOR data connection 550 CFSound-IV file system error	*
CWD	Change working directory	250 OK 550 CFSound-IV file system error	
DELE	Delete file	250 OK 550 CFSound-IV file system error	
MKD XMKD	Make directory	257 newDirectoryName 550 CFSound-IV file system error	
RMD	Remove directory	250 OK 550 CFSound-IV file system error	
RNFR	Rename File from	350 requested file action pending 550 CFSound-IV file system error	
RNFR	Rename File to	250 OK 503 command out of sequence 550 CFSound-IV file system error	
NOOP	No operation	250 OK	
*	Any other command not listed here	502 command not implemented	

Here's a sample session using the freeware FTP client FileZilla:



VNC Server Port

The CFSound-IV provides limited Virtual Network Computing (VNC) server capability. VNC is a graphical desktop sharing system that uses the Remote Frame Buffer (RFB) protocol to remotely control another computer. It transmits the keyboard and mouse events from the client to the server, relaying the graphical screen updates from the server back to the client over a network connection.

The CFSound-IV implements a 'virtual display' – screens can be displayed and interacted with by VNC connection only. In CFSound Mode the configuration screen is displayed. In Basic mode graphic commands can be used to display on and interact with this virtual screen which can only be observed via a VNC connection.

If the configured VNC Server Port is non-zero, RFB protocol messages will be exchanged over that port to establish and update a VNC session between VNC client software on a PC and the CFSound acting as a VNC server.

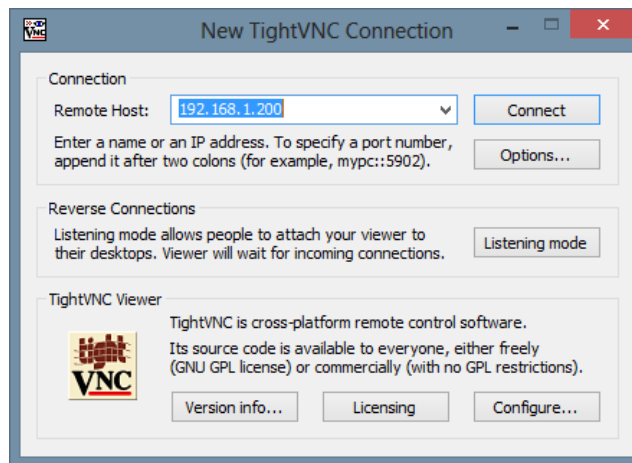
No passwords or authentication are supported. The server reported protocol is RFB 003.003.

Supported graphical update formats are 16-bit R5G6B5 or 32-bit R8G8B8A8. This is negotiated during the connection phase.

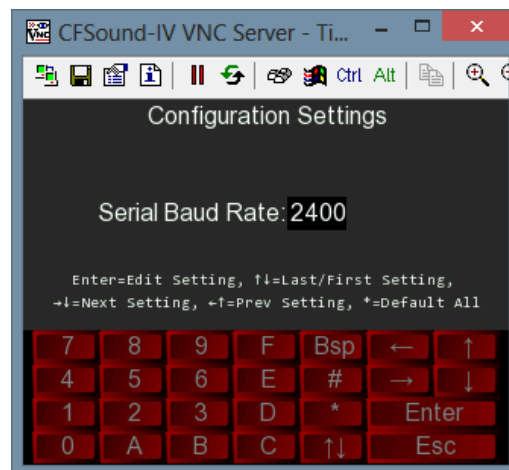
Received mouse events are translated to virtual touchscreen events; Press, Move, Release.

Received keyboard events for keys that have an ASCII code equivalent including Backspace, Tab, Escape and Enter are processed upon key down only – key up and all other keys are ignored.

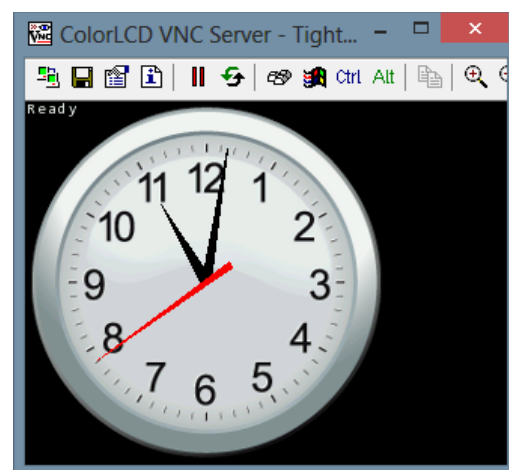
Here's an example using the free program Tight VNC to connect to the CFSound:



Setup connection



CFSound Mode Configuration Screen



Basic Animated Clock Demo Program

HTTP Server Port

The Arbitrary Precision CFSound-IV has the ability to serve HTML pages from the SD card. If the configured HTTP Server Port is non-zero then HTTP GET requests for SD card resident files will retrieve the file contents.

The server only responds to HTTP GET requests. The file path should immediately follow the IP Address:

GET /pathToFile

If the GET is followed by a slash/space “/ “ with no filename, an attempt is made to return the file “index.html” off of the card.

If the requested file is found and opens OK on the SD card a 200 response is returned:

HTTP/1.0 200 OK

Server: <http://www.cfsound.com>

Connection: close

This is immediately followed by a content type header determined by the requested file’s extension:

File extension	Returned Header
no extension	Content-type: application/octet-stream
.htm, .html	Content-type: text/html
.css	Content-type: text/css
.png	Content-type: image/png
.gif	Content-type: image/gif
.jpg	Content-type: image/jpeg
.bmp	Content-type: image/x-ms-bmp
any other extension	Content-type: text/plain

The content type header is followed by an empty line, then the requested file’s contents.

If the requested file doesn’t exist or can’t be opened on the micro SD card a 404 response is returned:

HTTP/1.0 404 Not found

Server: <http://www.cfsound.com>

Connection: close

This is immediately followed by the contents of file 404.html if it exists on the card.

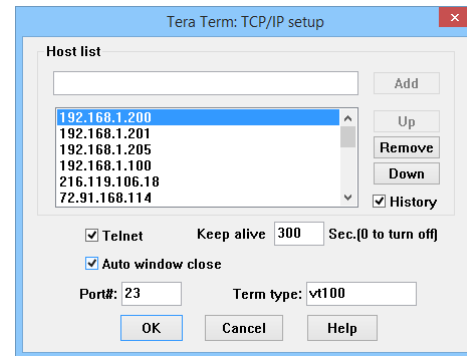
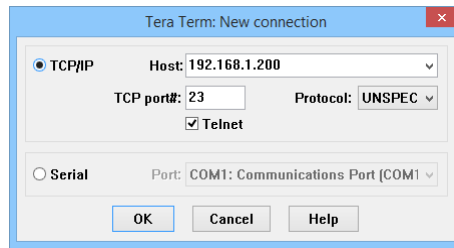
TCP/IP Raw Port

The Arbitrary Precision CFSound-IV has the ability to send and receive its serial data stream using TCP/IP raw sockets. Normally network traffic between two ports is encapsulated using an agreed upon protocol which may include message headers and other information required by the application. With raw sockets the data is sent / received un-encapsulated with no accompanying headers or other information.

If the configured TCP/IP Raw Port is non-zero a remote connection to that port allows the remote end to send data and received data to and from the CFSound as if over the conventional serial port.

In the [BETA](#) software the TCP/IP Raw configuration items have been renamed to TELNET and the code has been modified to work full-duplex. To support full-duplex operation the TELNET port now sends the <IAC> <DO> <suppress go ahead> command to the caller upon connection.

To connect to the CFSound-IV using TeraTerm and TELNET specify the TCP/IP connection using the CFSound's configured IP address and TELNET port:



NTP Client

The Arbitrary Precision CFSound-IV has the ability to synchronize its Real Time Clock (RTC) from a remote time server using the Network Time Protocol (NTP). NTP is a networking protocol for clock synchronization between computer systems over packet-switched, variable latency data networks.

There are published lists of NTP time servers, and the selected server should be geographically close. If at all possible the use of a local NTP server on the private network is strongly recommended to avoid overloading the public time servers. Windows XP for example can be configured to act as a NTP server. Some NAT routers can also be configured to provide NTP support.

The use of a remote time server external to the private network requires a correctly configured Router IP Address in order to perform the ARP for the requisite outgoing UDP packets.

NTP Server

This configuration item should be set to the IP address of the remote NTP time server.

NTP Client Port

If the configured NTP Client Port is non-zero and the NTP Server address is correctly addressing an IP Address hosting a NTP Server the CFSound-IV will attempt to sync its Real Time Clock to the time server after reset or power-up. No direct indication of successful synchronization is provided, however the RTC defaults to Tuesday, January 1, 2013, 12:00AM after reset or power-up.

Local Timezone

By definition, NTP servers provide Coordinated Universal Time (UTC). In order to compute the correct time the local time zone (offset hours from UTC) must be configured. For example; Eastern Standard Time is -5 hours from UTC.

Daylight Savings

In addition to the time zone offset from UTC correct local time requires knowledge of whether the location of the CFSound-IV is currently operating under Daylight Savings time. Setting this configuration item to Yes adds one hour to the computed local time.

Art-Net™

“Art-Net™ Designed by and Copyright Artistic Licence Holdings Ltd”

The CFSound-IV can operate as an Art-Net peer device to send and receive DMX512 data using an external Ethernet to DMX adaptor – such as the ENTTEC ODE:

http://www.enttec.com/?main_menu=Products&pn=70305

The CFSound-IV Artistic Licence OEM code is 1050₁₆.

The Arbitrary Precision Entertainment Services and Technology Association (ESTA) Manufacture ID is 08F9₁₆.

Art-Net IP Port

If the configured Art-Net IP Port is non-zero the CFSound-IV will send an ArtPollReply packet upon power-up or reset or upon receipt of an ArtPoll packet and then send ArtDmx packets every 4 seconds or whenever the DMX data generated by the CFSound mode or Basic mode changes at up to a 50mSEC rate.

The Art-Net Universe, Sub-Net and Net are configurable via the configuration settings. The target IP address defaults to 255.255.255.255 and is also configurable to support a directed broadcast or full IP address. Note that some Ethernet routers block directed broadcast packets.

In the [BETA](#) software, the changed DMX data is sent at up to a 25mSEC rate.

Modes of Operation

The CFSound-IV can operate in one of two different modes:

1. Conventional CFSound operation where the sound play-out is controlled by the association of contact numbers and sound filenames. This is referred to as the **CFSound Mode**.
2. Scriptable CFSound operation where the sound play-out is controlled by the execution of Arbitrary Precision Basic programs. This is referred to as the **Basic Mode**.

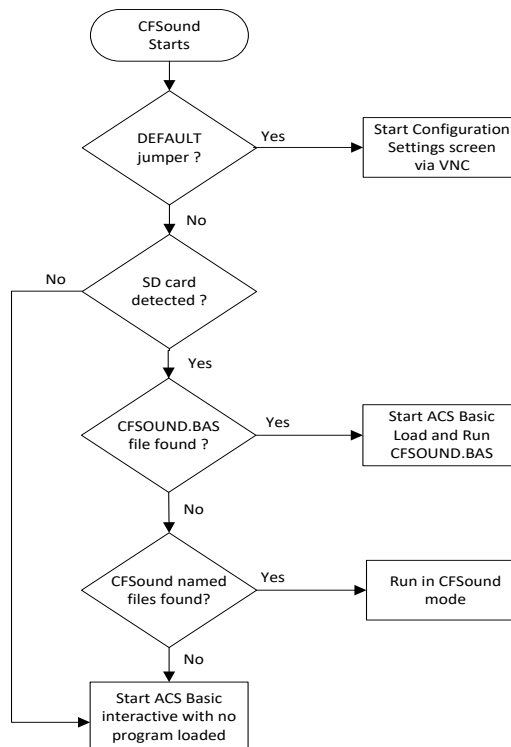
Upon power-up, reset or whenever a Secure Digital Flash card is inserted or removed, the unit determines its operating mode as follows; As the CFSound-IV scans for a SD card, and then files on the card, the Green LED flashes 3 times per second:

If a **SD card is not present**, the unit runs in Basic Mode by starting the Arbitrary Precision Basic interpreter and accepting/executing a subset of Basic commands that don't require file I/O via the serial port. The Green LED flashes once per second.

If a **SD card is present**, the unit looks for a file named **CFSOUND.BAS** and, if **found**, starts the Arbitrary Precision Basic interpreter, loading the file and executing the Basic program within in Basic Mode. The Green LED flashes once per second.

If a **SD card is present**, and the file **CFSOUND.BAS is not found**, the unit scans the card looking for sound files that follow the conventional CFSound Mode Filename Format. If at least one correctly named, valid sound file is found, the unit runs in CFSound Mode. If no valid sound files named in the conventional CFSound Mode Filename Format are found, the unit starts the Arbitrary Precision Basic interpreter and accepts/executes the complete set of Basic commands via the serial port.

The following flow diagram shows this operation determination sequence:



These two modes of operation are detailed in the following sections. Any errors encountered while scanning the SD card produce error code flashes on the Red LED. See the Diagnostic LED Codes section below for more information about LED error codes.

CFSound Mode

In CFSound Mode, the operation of the CFSound-IV is controlled by the names of the sound files on the SD card. Sounds are associated with contacts by the leading numeric portion of the filename, and the controlling state of the contact is specified by additional trailing non-numeric character attributes. Sounds may be played when a contact closes, opens, or only while a contact is closed or open. Sounds may be flagged as non-interruptible, or to only play when no other contact controlled sounds are playing. In addition, a small configuration text file may be placed onto the SD card along with the sound files to further control how the unit plays sounds (see the CFSOUND.INI section below).

Filename Formats

The filenames of the individual Sound files on the Secure Digital FLASH Card control how and when the sounds are played out by the **Compact FLASH IV Digital Audio System**. The filename must follow the DOS 8.3 naming convention – up to 8 characters of name with up to 3 characters of extension.

The filenames must start with an number which may be the contact input they are associated with expressed as a two digit HEX number.

All letters in the filename must be upper-case.

A HEX numeric digit is 0 through 9 and A through F, with 0 being equal to decimal 0, A being equal to 10 and F being equal to 15. The first digit (MSB) of the two digit hex number is multiplied by 16. The second digit (LSB) is added to the adjusted value derived from the first digit. The number 10 in HEX is thus equal to 16 decimal. The number 20 in HEX is 32 decimal, 21 in HEX is 33 decimal, and 7F in HEX is 127 decimal. For your convenience, a decimal to hex conversion table is provided below:

LSB → MSB↓	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1_	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2_	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3_	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4_	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5_	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6_	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7_	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8_	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9_	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A_	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B_	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C_	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D_	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E_	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F_	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

For example, a sound file associated with optional contact input #1 would be named “01.WAV,” a sound file for input #10 would be named “0A.WAV,” a file for input #16 would be “10.WAV” and a file for input #36 would be “24.WAV”. **There is no input #0 and so “00.WAV” is reserved and should not be used. There is a maximum of 254 sounds so “FF.WAV” is reserved and should not be used.**

The two digit hex input number must be the first two characters in the filename.

Additional letter tokens are appended to the input # in the filename to determine *when* and *how* the audio sound is played. The **Compact FLASH IV Digital Audio System** provides two inputs in its standard configuration and, by adding optional boards, can be expanded to 10, 18, 26 or 50 total inputs. The inputs can be wired to switches or other such devices, either normally opened or normally closed.

You can also connect the RS-232 port of the **Compact FLASH IV Digital Audio System** to the RS-232 port of your computer or PLC to control up to 254 sounds using serial commands.

The filename letter tokens are described as follows:

If a “**C**” appears after the hex number in the filename, the sound file will be played on a *Closure* of the corresponding optional contact input module input or serial command. The sound will start playing when the associated contact input goes from an open to closed state.

If an “**O**” appears after the hex number in the filename, the file is played on an *Opening* of the corresponding optional contact input module input or serial command. The sound will start when the associated contact input goes from a closed to an open state.

If a “**P**” after the hex number in the filename, the file will play only while the “**C**” or “**O**” state is true. If “**C**”, it will play while the associated contact input is closed and it will stop playing the sound when the input is opened. If “**O**”, it will play while the associated contact input is open and it will stop playing the sound when the input is closed.

If an “**R**” appears in the filename after the hex number, the sound is *repeated*, that is to say that the same sound file plays over and over. If used in conjunction with the “**P**” command, the sound file will repeat the entire time the “**C**” or “**O**” input condition is true. If “**C**”, the entire time the input is closed, if “**O**”, the entire time the input is open. If triggered without a “**P**” attribute, the file will play indefinitely or until another sound is played.

If an “**S**” is used in the filename after the hex number, the on-board Push-To-Talk (PTT) relay and contact closure corresponding to the input number on an optional plug-in board will be activated when the file is played.

If a “**B**” is used in the filename after the hex number, the sound is considered to be a *background* sound. Background sounds play when no other sound is playing, and multiple background sounds are played in the numerical sequence of the two digit input number. Sounds that are marked as Background may not be marked for contact closure or opening, play while or repeat. Such markings will be ignored.

If an “**N**” is used in the filename after the hex number, the sound is considered to be a *non-interruptible* sound. Sounds that are marked as non-interruptible will play to completion before any other sounds are played. Contact events that occur while non-interruptible sounds are playing are discarded unless the SaveNIContacts entry in the optional CFSOUND.INI file [Contacts] section is set to TRUE. Sounds that are marked as non-interruptible may not be marked for Repeat or as a Background sound. Such markings will be ignored.

If an “**F**” is used in the filename after the hex number, the sound is considered to be associated with a DMX channel Fade. Sounds that are marked for fade will ramp up the associated DMX channel value between the **BeginFadeValue** and **EndFadeValue** entries at the **FadeDuration** rate in the CFSOUND.INI file [DMX] section when the sound plays and ramp down when the sound ends or another sound is started.

If a “**D**” is used in the filename after the hex number, the sound is considered to be associated with one or more DMX channels defined by an associated scene ##.DMX file.

After the filename letter tokens a space, dash or underscore separator may be followed by a short text description.

Sample Filenames

- “01C.WAV” This file would play once, when triggered by optional contact input #1 closing.
- “02O.WAV” This file would play once, when triggered by optional contact input #2 opening.
- “04CP.WAV” This file would play once, when triggered by optional contact input #4 closing and play only while input #4 was closed without repeating.
- “07CPR.WAV” This file would play and repeat, the entire time optional contact input #7 was closed.
- “01OPR.WAV” This file would play and repeat, the entire time optional contact input #1 was open.
- “08CNS.WAV” This file would play once, without being interrupted when triggered by optional contact input #8 closing. The on-board PTT relay would be activated for the duration of the sound as well as optional contact output #8.
- “7EB.WAV” This file would play as a background sound whenever any other sound was not playing.
- “7FB.WAV” This file would also play as a background sound after the preceding file 7EB.WAV had played.
- “19C.WAV” This file would play once, when triggered by contact input #25 on the MAIN connector on the back of the CFSound-IV closing.
- “01CD.WAV” This file would play once, when triggered by optional contact input #1 closing. When the sound starts, the first 16 DMX channels would be set to values in the 01.DMX file’s [SoundStart] section, when the sound stops, the first 16 channels would be set to values in the [SoundStop] section of the same file.
- “19CF.WAV” This file would play once, when triggered by contact input #25 on the MAIN connector on the back of the CFSound-IV closing. When the sound starts, DMX channel #25 would fade-up to values at a rate determined by entries in the CFSOUND.INI file. When the sound stops DMX channel #25 would fade-down.

File Naming Format for CFSound style operation

Filename format: XX[COPRSBNFD].WAV

Where:

XX	Two digit ASCII Hex identifier 01 - FE, may be associated contact number
C	File plays on Closure of contact XX, may not be used with O
O	File plays on Opening of contact XX, may not be used with C
P	File plays while contact XX is closed or open, may not be used with B
R	File repeats, may not be used with B
S	On board PTT relay and contact XX will activate while sound is playing
B	File plays as background when no other sound is playing, may not be used with C, O, P, R or N
N	File playing is non-interruptible, may not be used with R or B
F	Matching DMX channel number Fades up/down with sound start/stop
D	First 32 channels set to entries in associated DMX scene file with sound start/stop
.WAV	File extension identifies Windows PCM sound file format

Secure Digital FLASH Card Loading and Scanning

Once programmed with the desired sounds and correct filenames, the card is inserted into the slot on the front of the **Compact FLASH IV Digital Audio System**. The green LED indicator should begin to flash indicating that the unit is scanning and loading the files from the card. When the card is correctly loaded and scanned, the green LED should glow steadily. Any problems encountered while reading the SD card are indicated by a repeating sequence of flashes on the red LED indicator (see the **Diagnostic LED Codes** section below for more information).

The card may be removed at any time. In that case, both the green and red LED indicators should turn off, then the Basic mode is started.

The optional contact inputs are ‘debounced’ by sampling them using a periodic interrupt. The sample rate is 50 Hertz (20 milliseconds). Valid input closures are detected by reading a 0 - 0 - 1 (open, open, closed) sample sequence. Valid input opens are detected by reading a 1 - 1 - 0 (closed, closed, open) sample sequence.

The status of the unit while loading and scanning is also displayed via messages sent out the serial port. Using a PC with “Tera Term”, “HyperTerminal” or other terminal emulator software connected to the **Compact FLASH IV Digital Audio System** serial port the card loading and scanning messages may be viewed. The serial data format defaults to 9600 baud, 8 data bits, 1 stop bit and no flow control. Samples of these messages are shown in the Appendix.

Background Audio Tracks

Sounds labeled with the “**B**” token in their filename after the hex number are played in numerical sequence as background sounds when no other audio tracks are being played, continuously repeating the sequence. A background audio track will cease playing when any other sound file is triggered to play by contact closure or serial command. Once that sound file is finished playing, the background audio file that was playing will resume playing from the point where it was interrupted and continue to play until again interrupted.

CFSOUND.INI File

Support is provided to control various operating parameters by the inclusion of a CFSOUND.INI file on the SD card. This file is a text file that is created using Notepad or other text editor and then is placed onto the SD card along with the sound files.

When a card is inserted and after all sound files have been loaded, the CFSound-IV unit looks for a file with this name, and reads initialization parameter entries from it. Parameter entries are organized into sections with a section name in brackets followed by one or more PARAMETER=VALUE lines. The Section Names and Parameter entries are case sensitive. A section or parameter may be commented out by placing a '#' character as the first character on the line. **The last line in the file should be a blank line.**

THE CFSOUND.INI FILE IS NOT REQUIRED IF YOU ARE NOT ALTERING A PARAMETER FROM THE DEFAULT VALUE SHOWN IN THE TABLE.

SECTIONS AND PARAMETERS THAT ARE NOT BEING CHANGED FROM THEIR DEFAULT VALUES IN THE TABLE ARE ALSO NOT REQUIRED TO BE IN THE FILE.

INCLUDING A [DMX] SECTION AND PARAMETERS WHEN THE DMX MODULE IS NOT INSTALLED WILL GENERATE AN ERROR INDICATION.

Here is a list of currently supported Sections and Parameter entries:

[Section] / Parameter	Description
[Comm]	Communications Port Section
BaudRate=dddd	Sets the serial port baudrate to the decimal value dddd. Default=9600.
[DEBUG]	Debug Section
ShowStartStop=TRUE/FALSE	Enables RS-232 message display of sound start/stop events. Default=FALSE.
[Background]	Background Section
BackgroundDelay=dddd	Sets the delay in seconds between background sound play-outs to the decimal value dddd. The delay starts when a background sound is started in order to maintain the delay interval regardless of the sound(s) length. Default=0.
BackgroundRestart=TRUE/FALSE	Enables interrupted background sound to restart from the beginning instead of where it was interrupted. Default=FALSE.
[Quiz]	Quiz Section
QuizMode=TRUE/FALSE	Enables Quiz/Kiosk mode of operation. Default=FALSE.
QuestionContacts=dd	Sets the number of question contacts to the decimal value dd. Default=4.
AnswerContacts=dd	Sets the number of answer contacts to the decimal value dd. Default=4.
NoAnswerTimeout=dd	Sets the delay in seconds between the end of the question sound and the timeout answer sound to the decimal value dd. Default=5.
AwaitAnswerSound=xx	Sets the hexadecimal sound number xx to play after the question sound before the timeout answer sound.

[Section] / Parameter	Description
	Default=0 (no sound).
QuestionTimeout=dd	Sets the delay in seconds between the end of the quiz question while waiting for a answer. Default=30.
AnswerWithoutQuestionSound=xx	Sets the hexadecimal sound number xx to play if an answer contact is activated before a question contact. Default=0 (no sound).
[Contacts]	Contacts Section
Force=TRUE/FALSE	Setting this value to TRUE restores the original CFSound contact behavior wherein the contact's active status is 'forced' upon reset, power-up or card-insertion. This will cause associated sound activation if the contact was active. Setting this value to FALSE (the default) causes the new behavior wherein the contact's current status is sampled upon reset, power-up or card-insertion. This will cause no associated sound activation until the contact is re-activated. Default=FALSE.
SequenceContactNumber=dd	Sets the number of the contact that will play sounds in sequence to the decimal value dd. Default=0 (no sequencing)
FirstSoundNumber=dd	Sets the first sound number that will be played in sequence to the decimal value dd. Default=1 (sound #1)
LastSoundNumber=dd	Sets the last sound number that will be played in sequence to the decimal value dd. Default=127 (sound #127)
SaveNIContacts=TRUE/FALSE	Setting this value to TRUE will remember any contact events that occur while a non-interruptible sound is playing. Note that this can cause a non-interruptible sound to play again if its contact is re-activated while it is playing. Default=FALSE
OutputContactModulus=dd	Setting this value to non-zero will cause the output contacts associated with sounds to repeat on the modulo value if QuizMode=FALSE . Example: OutputContactModulus=4 activates contact outputs 1 through 4 for sounds 1 through 4, contact outputs 1 through 4 for sounds 5 through 8, etc. Default=0
RandomSequence=TRUE/FALSE	Setting this value to TRUE will cause each activation of the SequenceContactNumber to play a random sound from the range FirstSoundNumber to LastSoundNumber. Default=FALSE
OffsetContactNumber=dd	Sets the number of the contact that will offset the sounds associated with the other contacts by ContactOffsetAmount to the decimal value dd. Does not affect Sequence or Quiz mode. Default=0 (no offset)

[Section] / Parameter	Description
ContactOffsetAmount=dd	Sets the value that will be added the the input contact number when the OffsetContactNumber input is active, to offset the actual sound number that will play to the decimal value dd. Does not affect Sequence or Quiz mode. Default=0 (no offset amount)
AutoplayEntireSequence=TRUE/FALSE	Setting AutoplayEntireSequence to TRUE causes the entire sequence of sounds to be played once whenever the SequenceContactNumber activates one time. Default=FALSE (no autoplay)
LineInputEnableContactNumber=dd	Sets the number of the contact that will stop any sound currently playing and enable the Line level Input to the decimal value dd. Audio on the Line level Input is amplified to the current volume setting and appears on the speaker and Line level Output. Default=0 (no Line In control contact)
PttOutputWithLineInputContact=TRUE/FALSE	Setting this value to TRUE will cause the PTT relay to follow the non-zero LineInputEnableContactNumber state, otherwise the PTT relay activation is controlled by sounds with the Relay attribute in their filename. Default=FALSE (PTT for sounds w/Relay attr)
[LineIn]	LineIn Section
LineInputAlwaysEnabled=TRUE/FALSE	Setting this value to TRUE enables the Line level Input always when no sound is playing. When this is FALSE, the Line level Input is controlled by the LineInputEnableContactNumber. Default=FALSE (Line level Input disabled)
[DMX]	DMX Section
IsMaster=TRUE/FALSE	Setting this value to TRUE enables the CFSound as a DMX controller. When this is FALSE, the CFSound is a DMX device. Default=FALSE (DMX device)
SlaveBaseChannel=ddd	Sets the base channel number for the group of channels that the CFSound monitors when configured as a DMX device (IsMaster=FALSE). Default=1
FrameDelay=dd	Sets the delay times 20mSEC between transmitted DMX frames. Default=0 (no delay)
NumberOfChannels=ddd	Sets the number of transmitted channels per DMX frame, always a multiple of 2. Default=512
BeginFadeValue=ddd	Sets the DMX channel value for the start of the Fade Up when the associated sound plays, and the value for the end of the Fade Down when the sound ends. Default=0
EndFadeValue=ddd	Sets the DMX channel value for the end of the Fade Up when the associated sound plays, and the value for the start of the Fade Down when the associated sound ends. Default=255
FadeDuration=dd	Sets the length of the DMX channel Fade Up

<u>[Section] / Parameter</u>	<u>Description</u>
	/ Fade Down when the associated sound plays, 0 = full on/off. Default=1

Sample CFSOUND.INI files

These three lines (blank line at end) typed into a text file with a text editor and saved on the SD card as CFSOUND.INI along with sounds named with the Background filename attribute provides a 5 minute (300 second) delay between played background sounds – the delay interval starts from when a background sound is played:

```
[Background]
BackgroundDelay=300
```

These six lines (blank line at end) typed into a text file with a text editor and saved on the SD card as CFSOUND.INI along with sounds named 01C.WAV, 02C.WAV, ... , 08C.WAV will play in a sequence with the next sound in the sequence played whenever that contact #25 input is momentarily closed.

```
[Contacts]
SequenceContactNumber=25
FirstSoundNumber=1
LastSoundNumber=8
AutoplayEntireSequence=FALSE
```

Only a single sequence triggered from a single contact input is supported in CFSound mode using the CFSOUND.INI file – for multiple sequences controlled by multiple contacts the unit operation must be scripted in Arbitrary Precision Basic.

Quiz / Kiosk Mode

The CFSound-IV may be configured to operate in a Quiz or Kiosk mode. This mode is enabled by including the requisite **[Quiz]** section entries in a CFSOUND.INI file on the card with the **QuizMode=TRUE** parameter.

In Quiz mode, Question contacts trigger the Question sounds, then Answer contacts trigger the Answer sounds related to the question that was activated. The Question contacts start at contact 1 and are contiguous through the number of **QuestionContacts=dd** parameter. The Question sounds would be named 01C.WAV, 02C.WAV, ... , xxC.WAV where xx is the hexadecimal equivalent of the **QuestionContacts=dd** parameter.

The Answer contacts immediately follow the Question contacts and are contiguous through the number of **AnswerContacts=dd** parameter. Each Answer contact is associated with an array of Answer sounds, with the offset into the array controlled by the Question contact that was activated. There is also an additional timeout Answer sound that will be played if an Answer contact is not activated within the **NoAnswerTimeout=dd** seconds interval.

Due to the maximum number of sounds being limited to 255 the following limitation must be observed:

$$\text{QuestionContacts} \times (\text{AnswerContacts} + 1) \leq 254$$

The next table shows a sample of Quiz mode sound file names and their contact associations given the following values for the [Quiz] section parameters: **QuestionContacts=4**, **AnswerContacts=4**, **AnswerWithoutQuestionSound=7E** and **AwaitAnswerSound=7F** in the CFSOUND.INI file. Notice how the Answer contacts play different sounds depending upon what Question contact was activated:

Sound Filename	Contact #	Description
01CS.WAV	1	Question #1 sound
02CS.WAV	2	Question #2 sound
03CS.WAV	3	Question #3 sound
04CS.WAV	4	Question #4 sound
05CS.WAV	5	Question #1 Answer #1 sound
06CS.WAV	6	Question #1 Answer #2 sound
07CS.WAV	7	Question #1 Answer #3 sound
08CS.WAV	8	Question #1 Answer #4 sound
09.WAV	-	Question #1 no Answer Timeout sound
0ACS.WAV	5	Question #2 Answer #1 sound
0BCS.WAV	6	Question #2 Answer #2 sound
0CCS.WAV	7	Question #2 Answer #3 sound
0DCS.WAV	8	Question #2 Answer #4 sound
0E.WAV	-	Question #2 no Answer Timeout sound
0FCS.WAV	5	Question #3 Answer #1 sound
10CS.WAV	6	Question #3 Answer #2 sound
11CS.WAV	7	Question #3 Answer #3 sound
12CS.WAV	8	Question #3 Answer #4 sound
13.WAV	-	Question #3 no Answer Timeout sound
14CS.WAV	5	Question #4 Answer #1 sound
15CS.WAV	6	Question #4 Answer #2 sound
16CS.WAV	7	Question #4 Answer #3 sound
17CS.WAV	8	Question #4 Answer #4 sound
18.WAV	-	Question #4 no Answer Timeout sound
7EC.WAV	5, 6, 7, 8	Answer Without Question Sound
7F.WAV	-	Awaiting Answer Sound

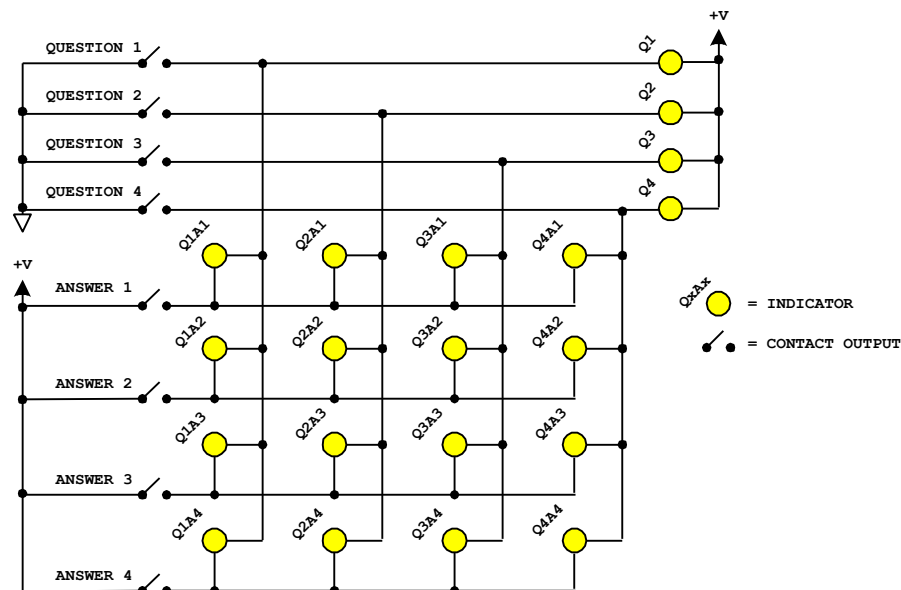
Note that while these sample files all have the 'C' contact closure token in the filename, the 'O' contact open token may be used as well. Sound files that are played without a direct contact closure do not require these tokens.

There are two additional, optional sounds that may be played in Quiz mode. The **AwaitAnswerSound=xx** parameter, if xx is non-zero, causes sound file xx.WAV to be played after the question sound while awaiting an Answer contact activation. In the preceding example, this would play sound file 7F.WAV. The duration of this optional sound or the **NoAnswerTimeout=dd** seconds parameter, whichever is greater, determines the interval before the associated no answer timeout sound file is played. The **AwaitAnswerSound** should not be marked as non-interruptible.

The **AnswerWithoutQuestionSound=xx** parameter if xx is non-zero causes sound file xx.WAV to be played if an Answer contact is activated before a Question contact to prompt user to select question first. In the preceding example, this would play sound file 7EC.WAV when any Answer contact is activated before a Question contact.

A Kiosk mode of operation may be obtained by setting the **NoAnswerTimeout=dd** parameter to zero. This effectively disables the **AwaitAnswerSound** and all of the **No Answer Timeout sounds** simply allowing the Question contact to select different sets of Answer contacts sounds such as various languages or features. After a Question contact is activated, it remains active selecting a given set of Answer contact sounds until another Question contact is activated.

Contact outputs may also be activated for Question and Answer sounds by using the 'S' token in the filename. In Quiz mode, Question contact outputs remain active for the duration of the Question/Answer cycle. In Kiosk mode, the last Question (answer selection) contact output remains active until another Question input is activated. Note that the Answer contact outputs associated with the Answer contact inputs are used for each set of answers, independent of which Question contact is activated. Unique Answer contact outputs may be achieved by constructing a matrix using the Question contact outputs as Rows and the repeating Answer contact outputs as Columns (or vice versa). This is shown in the following diagram:



Matrix Wiring of Question/Answer Contact Outputs

The **BackgroundDelay=dddd** parameter can be used in conjunction with one or more background sounds to provide a delay between sounds, so that they can be used as an attention getting attract mode.

Sequential Sound Play-out

The CFSound-IV may be configured to allow a single contact closure (or opening) to play several sounds in sequence; one each time it is closed (or opened). This mode is enabled by including the requisite

[**Contacts**] section entries in a CFSOUND.INI file on the card with the **SequenceContactNumber=dd** parameter set to a non-zero, valid contact number from 1 to 8 (1 to 24 with Contact Sense 24 module) or 25 and 26 (rear of unit contact inputs).

Only a single sequence triggered from a single contact input is supported in CFSound mode – for multiple sequences triggered by multiple contacts the unit must be scripted in Arbitrary Precision Basic.

The range of sounds that may be played in sequence may be constrained by the addition of two other parameter entries in the [**Contacts**] section. The **FirstSoundNumber=dd** parameter controls the starting sound number of the sequence, and the **LastSoundNumber=dd** parameter controls the ending sound number in the sequence. Setting the **FirstSoundNumber** parameter to a higher number than the **LastSoundNumber** will cause the sounds to be played in reverse sequence, higher number to lower number. The sequence wraps around to the beginning or end.

For example to play normal sounds on the first 8 contacts, and sequence through 16 different sounds for the 25th contact, a card would be prepared with the 24 sound files named 01C.WAV – 18C.WAV. Then a CFSOUND.INI file would be added to the card with the entries:

```
[Contacts]
SequenceContactNumber=25
FirstSoundNumber=9
LastSoundNumber=24
```

The first eight contacts would each play their associated sound. Each closure of contact 25 would play a sound from the 09C.WAV – 18C.WAV group then repeat from 09C.WAV – 18C.WAV in sequence. To reverse the order of the sound play-out set the **FirstSoundNumber=24** and the **LastSoundNumber=9**.

To cause the entire sequence of sounds to be played once whenever the **SequenceContactNumber** is activated, set the **AutoplayEntireSequence** parameter to TRUE. For example a museum exhibit needs to play a sequence of 8 sounds when triggered with each sound activating a relay output while it is playing. A SD card would be prepared with the 8 sound files in the sequence named 01CNS.WAV – 08CNS.WAV. Then a CFSOUND.INI file would be added to the card with the entries:

```
[Contacts]
SequenceContactNumber=25
FirstSoundNumber=1
LastSoundNumber=8
AutoplayEntireSequence=TRUE
```

Whenever contact 25 is closed, the eight sounds 01CNS.WAV – 08CNS.WAV would play in sequence, each sound activating its output relay. This feature only works for non-random sequences. The addition of the non-interruptible attribute to the file names prevents multiple contact 1 closures from prematurely advancing the sequence.

Please see the videos on the [SUPPORT](#) page of the [CFSOUND.COM](#) website.

RS-232 Serial Port Protocol

Sound play-out may also be controlled via commands received via the serial port. The data format is the current baud rate (default is 9600 baud), 8 data bits, 1 stop bit and no flow control.

Sounds may be started, queued and stopped using a simple ASCII protocol. The SOH/ETX protocol starts each command with a single ASCII Start of Header control character = 01₁₆ or **CTRL-A**, represented herein as **<SOH>** and ends each command with a single ASCII End of Text control character = 03₁₆ or **CTRL-C**, represented herein as **<ETX>**.

The actual command is determined by the next lower-case ASCII character following the **<SOH>**:

Command character:	Commands to:
p	Play or queue sounds
v	Control volume
a	Control amplifier
l	Control line input

The command character is followed by a single character command modifier that is command specific.

Command	Command modifier:	Modifies command to:
p	+	Start Sound
v		Increase Volume
a		Un-Mute Amplifier
l		Enable Line Input
p	-	Stop Sound
v		Decrease Volume
a		Mute Amplifier
l		Disable Line Input
p	&	Queue sound
	~	Flush queue
	!	Stop sound and flush queue
v	=	Set Volume Remembered
	<	Fade Volume Down
	>	Fade Volume UP
	^	Set Temporary Volume

The command modifier for some commands is followed by a data argument that is sent as ASCII Hex strings two characters in length.

The individual RS-232 commands are detailed on the following pages.

In the SOH/ETX command descriptions that follow there are spaces shown between the different parts of each command.

THE EMBEDDED SPACES IN THE FOLLOWING DESCRIPTIONS ARE FOR CLARITY ONLY, ARE NOT PART OF THE TRANSMITTED / RECEIVED COMMANDS AND SHOULD NOT BE INCLUDED!

Play / Queue Sound Command

Sound playback can be controlled via commands received via the serial port. Sounds may be started, stopped, queued to play sequentially and the queue flushed using variations of the 'p' command:

<SOH> p {+/-/&/~/!} XX <ETX>

where:

<SOH>	ASCII Start of Header character 01 ₁₆ (Ctrl-A)
p	ASCII lower case letter 'p'
{+/-}	ASCII plus character '+' to start a sound, minus character '-' to stop a sound
{&/~/!}	ASCII ampersand character '&' to queue a sound, tilde character '~' to flush the queued sounds, exclamation character '!' to stop the current playing sound and flush the queued sounds
XX	ASCII two digit hexadecimal number XX ₁₆ of the sound (01, 02, . . . , 7E, 7F, . . . , FE (maximum))
<ETX>	ASCII End of Text character 03 ₁₆ (Ctrl-C)

Examples:

Start playing sound #25:

<SOH>p+19<ETX>

Stop playing sound #25 (if it was playing):

<SOH>p-19<ETX>

Stop playing all sounds (background sounds will resume):

<SOH>p-00<ETX>

Queue sounds #26 and #48 to be played in sequence:

<SOH>p&1A<ETX><SOH>p&30<ETX>

Stop playing the current sound and flush the queued sounds:

<SOH>p!00<ETX>

Up to 128 sounds may be queued. Sounds that are queued are played in succession in the order that they were queued only when no other sound is playing until the queue is exhausted or flushed. Note that this can occur at the end of the current background sound file before it repeats. A background or other sound that is playing may be stopped by issuing a "<SOH>p-00<ETX>" serial command sequence to allow queued sounds to be heard. The background sound will be resumed when the queue has emptied.

Volume Command

Sound volume may also be controlled via commands received via the serial port. The volume is essentially decibel linear in sixty-four, 1db increments from -66db (value of 0) → 0db (value of 63). The current volume level is ‘remembered’ in non-volatile memory on the CFSound-IV and is restored to the last setting upon power-up.

Sound volume may be set, increased or decreased, faded up or down using variations of the ‘v’ command:

<SOH> v {=/+/-/</^/>} XX <ETX>

where:

<SOH>	ASCII Start of Header character 01 ₁₆ (Ctrl-A)
v	ASCII lower case letter ‘v’
{=}	ASCII equals character ‘=’ to set and remember the volume to the absolute value XX
{+/-}	ASCII plus character ‘+’ to increase the volume by XX, minus character ‘-’ to decrease the volume by XX
{<}	ASCII less than ‘<’ to fade the volume to 0 over XX seconds
{^}	ASCII caret character ‘^’ to set without remembering the volume to the absolute value XX
{>}	ASCII greater than ‘>’ to fade the volume up to the remembered value over XX seconds
XX	ASCII two digit hexadecimal value XX ₁₆ of the volume modification (00, 01, . . . , 3E, 3F) = (0 → 63)
<ETX>	ASCII End of Text character 03 ₁₆ (Ctrl-C)

Examples:

Set and remember the volume to maximum:

<SOH>v=3F<ETX>

Increase and remember the volume by 3 steps:

<SOH>v+03<ETX>

Fade the volume from its current setting to zero over 10 seconds:

<SOH>v<0A<ETX>

Set the volume temporarily to zero and then fade up to the remembered value over 5 seconds:

<SOH>v^00<ETX><SOH>v>05<ETX>

Amplifier Command

The speaker amplifier may also be muted via commands received via the serial port. Muting does not affect the current volume setting and is automatically turned off whenever the CFSound is Reset. The speaker amplifier may be muted / un-muted using the 'a' command:

<SOH> a {+/-} <ETX>

where:

<SOH>	ASCII Start of Header character 01 ₁₆ (Ctrl-A)
a	ASCII lower case letter 'a'
{+/-}	ASCII plus character '+' to un-mute the speaker amplifier, minus character '-' to mute the speaker amplifier
<ETX>	ASCII End of Text character 03 ₁₆ (Ctrl-C)

Examples:

Mute (silence) the amplifier:

<SOH>a-<ETX>

Un-mute the amplifier:

<SOH>a+<ETX>

Line Input Command

The line input may also be controlled via commands received via the serial port. The Line Input may be enabled / disabled using the 'l' command:

<SOH> l {+/-} <ETX>

where:

<SOH>	ASCII Start of Header character 01 ₁₆ (Ctrl-A)
l	ASCII lower case letter 'l' (el not one)
{+/-}	ASCII plus character '+' to enable the line input, minus character '-' to disable the line input
<ETX>	ASCII End of Text character 03 ₁₆ (Ctrl-C)

Examples:

Enable the Line Input:

<SOH>l+<ETX>

Disable the Line Input:

<SOH>l-<ETX>

DMX Operation

The CFSound-IV can be configured to be a DMX controller (Master) or DMX device (Slave). DMX operation is provided via the ArtNet™ protocol over Ethernet and an external Ethernet to DMX converter. DMX operation is controlled by the CFSOUND.INI file **IsMaster** parameter as well as the presence of the DMX Scene or Fade filename attributes.

DMX Controller (Master)

The CFSound-IV becomes a DMX controller (Master) when it finds files on the SD card with the DMX Scene or Fade filename attributes, OR, when it reads the CFSOUND.INI file and finds a **[DMX]** section with the **IsMaster** parameter set to TRUE (see the Filename Formats and CFSOUND.INI sections above). As a DMX controller the CFSound produces the ArtNet Ethernet packets that commands connected DMX devices to brighten/dim, move, pan or otherwise respond to their DMX channel values.

When a sound is played with the filename **Fade** attribute, the CFSound begins a fade-up of the channel value with the same number as the sound. When the sound stops playing a fade-down of the channel value occurs. The beginning and ending channel fade values as well as the fade duration can be controlled by entries in the CFSOUND.INI file.

When a sound is played with the filename **Dmx** scene attribute, the CFSound sets the first 32 channels to values obtained from an associated **##.DMX** file – the **##** must match the two character sound number of the associated **##CDx.WAV** or **##ODx.WAV** file. The content of the **.DMX** text files follow a similar format as the CFSOUND.INI file:

[Section] / Parameter	Description
[SoundStart]	Sound Start section
1=ddd	Sets the value of channel 1 to ddd (0 to 255) when this sound starts. Default=0
2=ddd	Sets the value of channel 2 to ddd (0 to 255) when this sound starts. Default=0
...	...
16=ddd	Sets the value of channel 16 to ddd (0 to 255) when this sound starts. Default=0
[SoundStop]	Sound Stop section
1=ddd	Sets the value of channel 1 to ddd (0 to 255) when this sound stops. Default=0
2=ddd	Sets the value of channel 2 to ddd (0 to 255) when this sound stops. Default=0
...	...
16=ddd	Sets the value of channel 16 to ddd (0 to 255) when this sound stops. Default=0

DMX Device (Slave)

The CFSound-IV becomes a DMX device (Slave) when it finds no files on the SD card with the **DMX** Scene or **Fade** filename attributes, OR, when it reads the CFSOUND.INI file and finds a **[DMX]** section with the **IsMaster** parameter set to FALSE (this is the default, see the Filename Formats and CFSOUND.INI sections above). Using the ArtNet™ protocol and external Ethernet to DMX device the CFSound monitors the received ArtNet data packets for DMX channel values.

The CFSound-IV monitors a group of three channels, starting at the channel selected by the **SlaveBaseChannel** parameter of the CFSOUND.INI file for changes in value. Sounds may be stopped, started and queued by setting the DMX Sound Number channel value to the desired sound number and toggling the DMX Command channel value between 0 (no operation) and one of the other command values:

DMX Channel Number	Register	Description
SlaveBaseChannel + 0	Command	0 = No operation 1 = Play Sound Number 2 = Stop playing Sound Number 3 = Queue Sound Number 4 = Clear queue 5 = Stop playing and clear queue 6 = Mute amplifier 7 = Un-mute amplifier
SlaveBaseChannel + 1	Sound Number	0 to 254 When stopping 0 = all sounds
SlaveBaseChannel + 2	Volume	0 to 255

Up to 128 sounds may be queued. Sounds that are queued are played in succession in the order that they were queued only when no other sound is playing until the queue is exhausted or cleared. Note that this can occur at the end of the current background sound file before it repeats. A background or other sound that is playing may be stopped by issuing a 'stop playing all sounds' command sequence to allow queued sounds to be heard. The background sound will be resumed when the queue has emptied.

Sound volume may also be controlled by changing the value of the DMX Volume channel. The 256 channel values are mapped to the 64 volume values by dividing the channel value by 4. The resulting volume is essentially decibel linear in sixty-four, 1db increments from -66db (value of 0) → 0db (value of 63). The current volume setting is not 'remembered' by the CFSound between power-up, reset, or SD card insertions. Note that the volume push buttons on the front of the unit will appear to not work since their 'settings' are constantly being over-ridden by received DMX frames containing a received volume setting.

Here's an example - CFSOUND.INI file contains:

```
[DMX]
SlaveBaseChannel=50
```

Initially DMX channels SlaveBaseChannel+0 through SlaveBaseChannel+2 should be set to zero. This is the no operation command, with the volume set to minimum:

DMX Channel	Value	Description
50	0	No Operation Command
51	0	Sound Number
52	0	Volume

To start a sound playing, set the SlaveBaseChannel+1 to the sound number, set the SlaveBaseChannel+2 to the desired volume, then set the SlaveBaseChannel+0 to 1. To play sound number 10 at full volume:

DMX Channel	Value	Description
50	0	No Operation Command
51	10	Sound Number
52	255	Volume

Then issue the Play Sound command by changing the Command channel's value from No Operation to Play Sound— the transition from No Operation to a command value causes the command to take effect:

DMX Channel	Value	Description
50	1	Play Sound Command
51	10	Sound Number
52	255	Volume

After the DMX frame has been sent, the SlaveBaseChannel+0 can be set back to zero – no operation – in preparation for the next command:

DMX Channel	Value	Description
50	0	No Operation Command
51	10	Sound Number
52	255	Volume

The SlaveBaseChannel+2 continually updates the CFSound-IV volume as DMX frames are received – this channel's value can be tied to a potentiometer or slider on the DMX master's console to adjust the volume.

To stop all sounds that have been queued and are playing:

DMX Channel	Value	Description
50	5	Stop Playing Clear Queue Command
51	n/a	Sound Number (don't care)
52	255	Volume

After the DMX frame has been sent, the SlaveBaseChannel+0 can be set back to zero – no operation – in preparation for the next command:

DMX Channel	Value	Description
50	0	No Operation Command
51	n/a	Sound Number (don't care)
52	255	Volume

Basic Mode

In Basic Mode, the operation of the CFSound-IV is controlled by the interpreted execution of a user written program that is written in the Arbitrary Precision Basic language. **BASIC** is the acronym for **B**eginners' **A**ll-purpose **S**ymbolic **I**nstruction **C**ode – an easy to learn programming language developed by Kemeny & Kurtz at Dartmouth in 1964. Programs consist of numbered lines with one or more English keywords describing the operations that the user wants to have performed.

The BASIC interpreter built-in to the CFSound-IV implements most of the language elements for working with integers, strings and files as well as providing access to the CFSound-IV hardware features such as contacts, timers, clock and playing sounds. The language elements are outlined in the separate document [“CFSound-IV Basic Programming”](#) available for download on the CFSound website.

Programs may be developed two different ways:

1. Interactively using a terminal emulator such as Windows HyperTerminal or TeraTerm connected to the CFSound-IV's serial port.

See the [“Upgrading the Firmware”](#) section at the end of this manual for a description of the required cabling and terminal emulator settings.

This is the preferred development methodology.

2. Offline using a text editor such as Windows Notepad. The files are then saved onto the SD card for debugging when inserted into the CFSound.

This methodology is slow, cumbersome and error-prone since it is not interactive and errors cannot be viewed or addressed in-situ and is not recommended.

Once a program has been interactively developed and tested, it may then be placed or saved onto the SD card named as CFSOUND.BAS along with any other required files. The CFSound will then load and execute this file upon power-up, reset or SD card insertion, effectively customizing the unit's function.

Diagnostic LED Codes (Red & Green LEDs)

Green LED

- Flashes three times a second while scanning or loading SD Card files.
- Steady on when loaded, ready or playing in CFSound mode.
- Flashes once a second while running Arbitrary Precision Basic mode
- Toggles three times a second alternating with the Red LED while self-recording.
- Flashes three times a second in sync with the Red LED while the DEFAULT jumper is installed.

Red LED Error Flashes

The Red LED can indicate up to 4 error conditions by flashing 1 to 4 unique numeric code patterns from the following table in a continuously repeating sequence. The Red LED flashes at two and a half times a second for the value of the pattern, followed by a gap of six tenths of a second, then the next pattern is presented.

When self-recording the Red LED alternates with the Green LED three times a second while the recording is armed, and is Steady On while recording.

Flash Count	Error Condition	CFSound Stops
1	Problem w/SD interface, non-compatible SD card, bad SD card, execution stops.	•
2	Error opening .WAV file for that input, incorrect filename, file does not exist.	•
3	Error when reading file, file corrupted, unsupported file type, the sound doesn't start or stops playing.	•
4	No valid "*.WAV" files of a CFSound Mode Filename Format found on the SD Card – starting Arbitrary Precision Basic. This code will continue until a Basic RUN command is issued.	
5	No valid FAT or FAT32 file system found on the SD Card, execution stops.	•
6	Error seeking within file when starting or resuming sound, the sound doesn't start or resume.	•
7	Error initializing CODEC, execution stops.	•
8	Invalid .WAV file internal structure, file is ignored.	•
9	Unsupported .WAV format (not Windows PCM), file is ignored.	•
10	Unsupported .WAV sample rate (not 8KHz, 11.025KHz, 16KHz, 22.05KHz, 32KHz or 44.1KHz), file is ignored.	•
11	Unsupported .WAV sample size (not 8-bit or 16-bit), file is ignored.	•
12	Contact closure without matching sound file, a contact closure with matching sound file stops this code from flashing.	
13	Speaker is muted by RS-232 command or Arbitrary Precision Basic command.	
14	Problem with CFSOUND.INI file entries.	•
15	CFSOUND.INI [DMX] section with no DMX I/O module detected.	
19	Error writing file when self-recording	
20	Error closing file when self-recording	
21	Error renaming .TMP file at end of self-recording	

Power Requirements

The idle current of a CFSound-IV unit, no expansion modules with a 4GB SanDisk SD card installed is ~80mA @ 12VDC input.

With a background sound playing at a moderate level, the current consumption averages ~200mA @ 12VDC.

Digital Audio Recording

Initially make all recordings (either music or voice) with a sample rate of at least 44.1KHz and 16-bit mono or stereo. This high quality, first-generation recording will later provide the greatest bandwidth and produce the best sounding final audio.

Recommended Recording Procedure

1. Record any audio clip of your choosing @ 44KHz, 16-bit, Mono or Stereo. Make sure that your samples do not clip (go above or below the sample window). The peaks of your audio clip, should, however, fill vertically at least 95% of the sample window. If they do not, you will need to boost your input signal by using a pre-amp or by some other means.
2. Edit the clip until satisfied. You can use a sound editor or filter program, such as Syntrillium's Cool Edit to normalize the amplitude of the sample. Use this command cautiously though, since it also tends to amplify noise levels. It is always preferable to re-record your sound clip at a higher level to achieve better fidelity.

It is imperative that all initial editing and filtering be done to the audio clip while it is formatted at 16-bit stereo and 44.1KHz. Please be certain that your recording environment is absolutely quiet. Recordings at 8-bit or other sample rates can be played as-is or up-converted using suitable software.

Also remember that digital play-out devices do not introduce any static of their own other than quantization noise. They only play back exactly what was recorded. If there is excessive static in the audio clip or sample, then there was probably static in it originally.

Please see the videos on the [support](#) page of the CFSOUND.COM website for additional information.

Self-Recording Procedure

The CFSound-IV has the ability to self-record audio presented at the line-level inputs into a contact associated .WAV file on the SD card. This requires a freshly formatted relatively fast SD card to operate properly – slow or fragmented SD cards may cause glitches in the recordings.

1. Connect your audio source to the CFSound line-level inputs. The audio level must be adjusted externally to avoid low volume and/or clipping in the recording.
2. Press and hold both CFSound volume up/down buttons. After about half a second the Green and Red LEDs underneath the SD card should begin toggling back and forth indicating that the recording process is Armed. Once the LEDs start toggling, the volume buttons may be released. On a freshly formatted SD card, the buttons would need to be held during Reset/Power Up to prevent Basic from starting since there are no contact named .WAV files on the card.
3. Now close and hold the contact that the recording is to be associated with. The Red LED should stay on solid while the Green LED continues to blink - indicating that the CFSound is recording. When the contact is opened the recording will stop, the Red LED will go out and the file will be renamed to associate it with the contact closure. Problems during or after the recording are indicated by a flashing Red LED error code. Pressing the contact again should play back the recorded sound.

4. Repeat this process for any other sounds you wish to record and associate with contact closures.

Optional Plug-in Modules

The **Compact FLASH IV Digital Audio System** supports optional plug-in modules to provide Contact I/O capability. Up to two modules may be installed along the front and back of the unit.

CFSound-IV Contact Sense 8 Module

This module provides eight contact inputs. The contacts on the rear module are numbered 1 through 8 and are associated with sound files named 01x.WAV through 08x.WAV. The contacts on the front module are numbered 33 through 40 and are associated with sound files named 21x.WAV through 28x.WAV. The x denotes additional file control attribute characters. See the **Filename Formats** section above.

The Input connector INP1 consists of a 10 pin, 2 piece terminal block style connector. See the Contact Wiring section below for a discussion of the electrical input structure and connections. The pinout is shown in the following rear diagram and table:

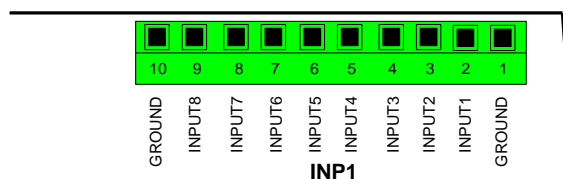


INP1 Pin #	Rear Signal	Rear Filename	Front Signal	Front Filename
1	GROUND		GROUND	
2	INPUT 1	01x.WAV	INPUT 33	21x.WAV
3	INPUT 2	02x.WAV	INPUT 34	22x.WAV
4	INPUT 3	03x.WAV	INPUT 35	23x.WAV
5	INPUT 4	04x.WAV	INPUT 36	24x.WAV
6	INPUT 5	05x.WAV	INPUT 37	25x.WAV
7	INPUT 6	06x.WAV	INPUT 38	26x.WAV
8	INPUT 7	07x.WAV	INPUT 39	27x.WAV
9	INPUT 8	08x.WAV	INPUT 40	28x.WAV
10	GROUND		GROUND	

CFSound-IV Contact I/O 8 Module

This module provides eight contact inputs and eight contact outputs. The contacts on the rear module are numbered 1 through 8 and are associated with sound files named 01x.WAV through 08x.WAV. The contacts on the front module are numbered 33 through 40 and are associated with sound files named 21x.WAV through 28x.WAV. The x denotes additional file control attribute characters. See the **Filename Formats** section above. If an 'S' token is present in the sound filename, the associated output contact and on-board PTT relay will be activated whenever the sound is played.

The Inputs connector INP1 is a 10 pin, 2 piece terminal block style. See the Contact Wiring section below for a discussion of the electrical input structure and connections. The pinout is shown in the following rear diagram and table:



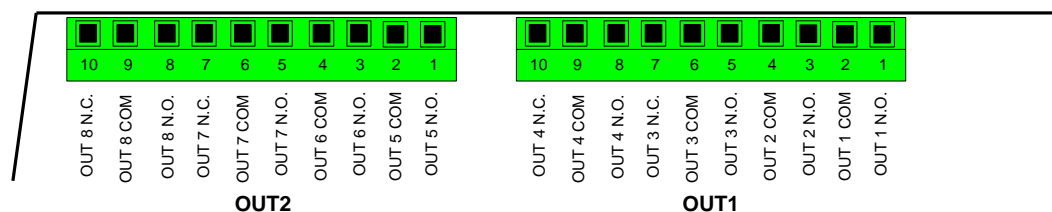
INP1 Pin #	Rear Signal	Rear Filename	Front Signal	Front Filename
------------	-------------	---------------	--------------	----------------

1	GROUND		GROUND	
2	INPUT 1	01x.WAV	INPUT 33	21x.WAV
3	INPUT 2	02x.WAV	INPUT 34	22x.WAV
4	INPUT 3	03x.WAV	INPUT 35	23x.WAV
5	INPUT 4	04x.WAV	INPUT 36	24x.WAV
6	INPUT 5	05x.WAV	INPUT 37	25x.WAV
7	INPUT 6	06x.WAV	INPUT 38	26x.WAV
8	INPUT 7	07x.WAV	INPUT 39	27x.WAV
9	INPUT 8	08x.WAV	INPUT 40	28x.WAV
10	GROUND		GROUND	

The CFSound-IV Contact I/O 8 Module provides eight normally open (N.O.) relay contact outputs. Outputs 3, 4, 7 and 8 also provide the normally closed (N.C.) relay contact outputs.

Each relay contact is rated at 1A @ 30VDC or 0.5A @ 125VAC.

The Output connectors OUT1 and OUT2 consist of two 10 pin, 2 piece terminal block style mounted adjacently. The pinout is shown in the following rear diagram and table:

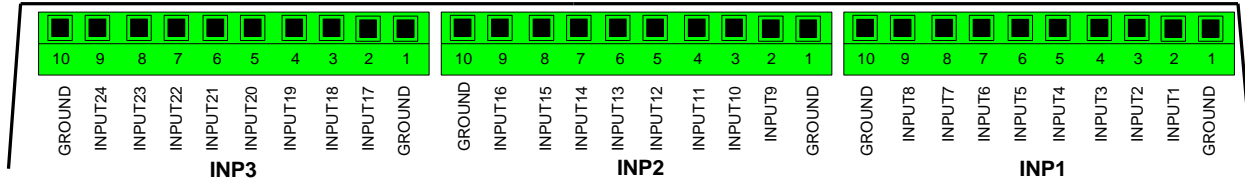


OUT2 Pin #	Rear Signal	Front Signal	OUT1 Pin #	Rear Signal	Front Signal
1	OUT 5 N.O.	OUT 37 N.O.	1	OUT 1 N.O.	OUT 33 N.O.
2	OUT 5 COM	OUT 37 COM	2	OUT 1 COM	OUT 33 COM
3	OUT 6 N.O.	OUT 38 N.O.	3	OUT 2 N.O.	OUT 34 N.O.
4	OUT 6 COM	OUT 38 COM	4	OUT 2 COM	OUT 34 COM
5	OUT 7 N.O.	OUT 39 N.O.	5	OUT 3 N.O.	OUT 35 N.O.
6	OUT 7 COM	OUT 39 COM	6	OUT 3 COM	OUT 35 COM
7	OUT 7 N.C.	OUT 39 N.C.	7	OUT 3 N.C.	OUT 35 N.C.
8	OUT 8 N.O.	OUT 40 N.O.	8	OUT 4 N.O.	OUT 36 N.O.
9	OUT 8 COM	OUT 40 COM	9	OUT 4 COM	OUT 36 COM
10	OUT 8 N.C.	OUT 40 N.C.	10	OUT 4 N.C.	OUT 36 N.C.

CFSound-IV Contact Sense 24 Module

This module provides twenty four contact inputs. The contacts on the rear module are numbered 1 through 24 and are associated with sound files named 01x.WAV through 18x.WAV. The contacts on the front module are numbered 33 through 56 and are associated with sound files named 21x.WAV through 38x.WAV. The x denotes additional file control attribute characters. See the **Filename Formats** section above.

The Input connectors INP1, INP2 and INP3 consist of three 10 pin, 2 piece terminal block style connectors mounted adjacently. See the Contact Wiring section below for a discussion of the electrical input structure and connections. The pinout is shown in the following rear diagram and table:

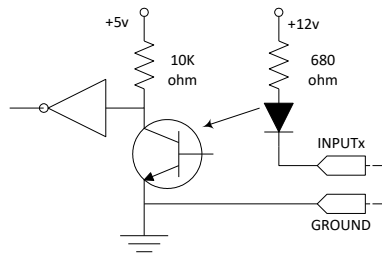


INP3 Pin #	Rear Signal	Front Signal	INP2 Pin #	Rear Signal	Front Signal	INP1 Pin #	Rear Signal	Front Signal
1	GROUND	GROUND	1	GROUND	GROUND	1	GROUND	GROUND
2	INPUT17	INPUT49	2	INPUT9	INPUT41	2	INPUT1	INPUT33
3	INPUT18	INPUT50	3	INPUT10	INPUT42	3	INPUT2	INPUT34
4	INPUT19	INPUT51	4	INPUT11	INPUT43	4	INPUT3	INPUT35
5	INPUT20	INPUT52	5	INPUT12	INPUT44	5	INPUT4	INPUT36
6	INPUT21	INPUT53	6	INPUT13	INPUT45	6	INPUT5	INPUT37
7	INPUT22	INPUT54	7	INPUT14	INPUT46	7	INPUT6	INPUT38
8	INPUT23	INPUT55	8	INPUT15	INPUT47	8	INPUT7	INPUT39
9	INPUT24	INPUT56	9	INPUT16	INPUT48	9	INPUT8	INPUT40
10	GROUND	GROUND	10	GROUND	GROUND	10	GROUND	GROUND

INP3 Pin #	Rear Filename	Front Filename	INP2 Pin #	Rear Filename	Front Filename	INP1 Pin #	Rear Filename	Front Filename
2	11x.WAV	31x.WAV	2	09x.WAV	29x.WAV	2	01x.WAV	21x.WAV
3	12x.WAV	32x.WAV	3	0Ax.WAV	2Ax.WAV	3	02x.WAV	22x.WAV
4	13x.WAV	33x.WAV	4	0Bx.WAV	2Bx.WAV	4	03x.WAV	23x.WAV
5	14x.WAV	34x.WAV	5	0Cx.WAV	2Cx.WAV	5	04x.WAV	24x.WAV
6	15x.WAV	35x.WAV	6	0Dx.WAV	2Dx.WAV	6	05x.WAV	25x.WAV
7	16x.WAV	36x.WAV	7	0Ex.WAV	2Ex.WAV	7	06x.WAV	26x.WAV
8	17x.WAV	37x.WAV	8	0Fx.WAV	2Fx.WAV	8	07x.WAV	27x.WAV
9	18x.WAV	38x.WAV	9	10x.WAV	30x.WAV	9	08x.WAV	28x.WAV

Contact Wiring

On the optional Contact Modules, switch contacts are wired between the Input pin number and a ground located on either end of the connector. The inputs are optically isolated using optocouplers. The cathodes of the LEDs in the optocouplers are connected to the Input connector port pins. The Anodes of the LEDs in the optocouplers are connected to an internal 12VDC power supply, with a 680 ohm current limiting resistor in series. The output transistor of each optocoupler has a 10K pullup resistor on its collector, with the emitter connected to ground and is buffered by an inverting gate. The following diagram is representative of one input:

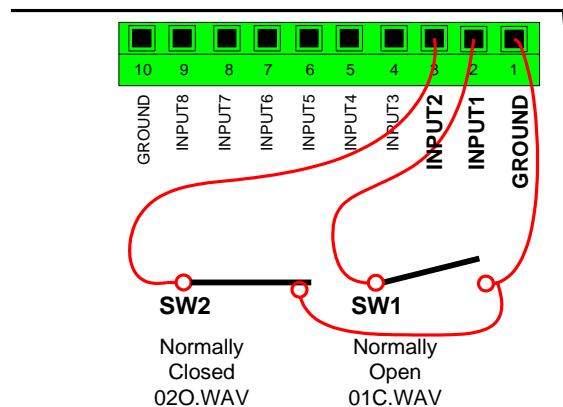


An input is activated by sinking current from the corresponding input pin to ground. A Ground connection is supplied on pins 1 and 10 of each Input connector for this purpose. The input current sink requirement is approximately 15 mA.

Switches may be one of two forms: Normally Open (N.O.) or Normally Closed (N.C.). Switches that are Normally Open have no electrical connection between the switch terminals unless the switch is activated closed. Switches that are Normally Closed have an electrical connection between the switch terminals unless the switch is activated open.

The CFSound-IV Contact Modules can work with either form of switch by using the appropriate filename letter token in the filename. A sound can be played when a Normally Open switch *closes* by placing the 'C' letter token after the sound number in the sound filename. In a similar fashion, a sound can be played when a Normally Closed switch *opens* by placing the 'O' letter token after the sound number in the sound filename.

In the following diagram, Input1 is wired to a Normally Open switch SW1, and Input2 is wired to a Normally Closed switch SW2. If this contact module is installed on the rear expansion connector, the sound file to be played when SW1 closes would be named 01C.WAV. The sound file to be played when SW2 opens would be named 02O.WAV.

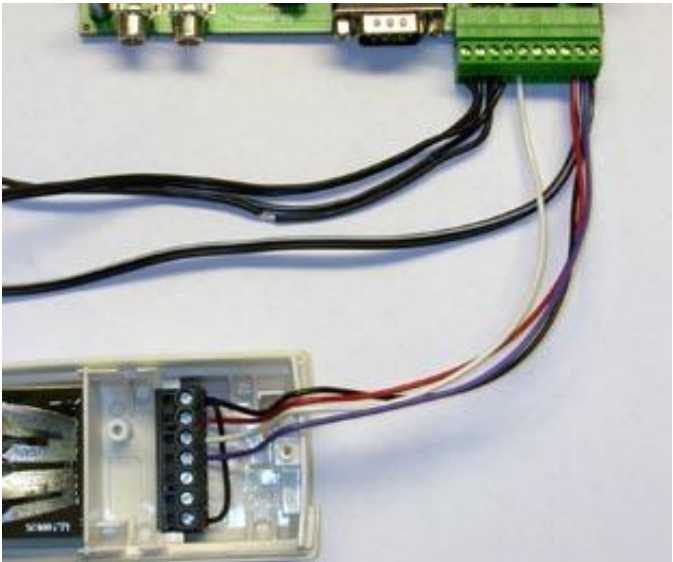


Connecting a Motion Sensor

The CFSound-IV can be connected to a motion sensor similar to those used with wired security systems.

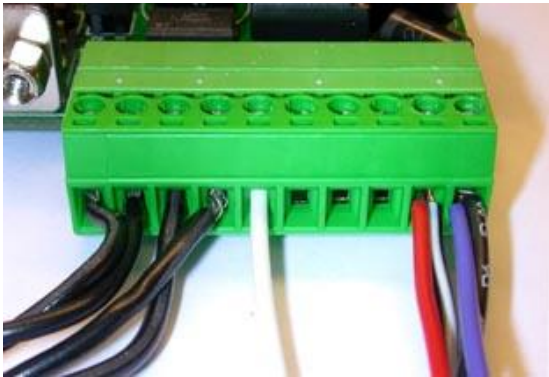
Normal Sized Motion Sensor

The photos below show the CFSound-IV wired to the GE AP100 Motion Sensor on contact input #25. Arbitrary Precision sells this sensor online as the "Normal Size Motion Sensor".



CFSound-IV MAIN

GE AP100 Motion Sensor



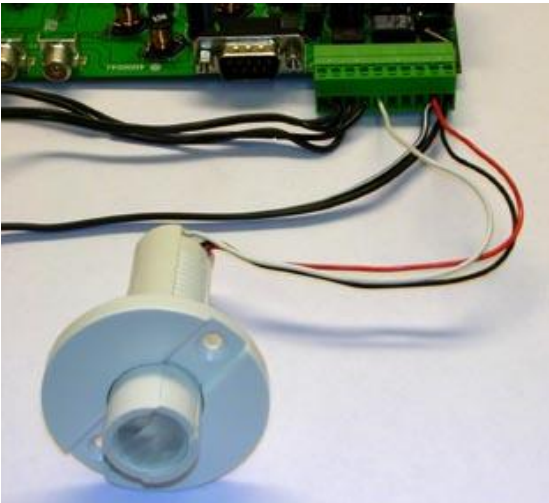
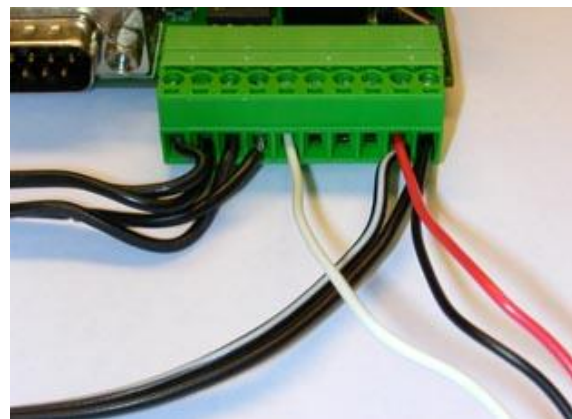
Pins 1-4	BLK	Speakers
Pin 5	WHT	Contact Input #25
Pin 9	BLK/WHT	+12VDC from transformer +12VDC
	RED	power to motion sensor
	BLK	Ground from transformer
Pin 10	BLK	Ground to motion sensor power
	BLUE	Ground to motion sensor loop

Pin 1	BLK	Ground to motion sensor
Pin 2	RED	+12VDC power to motion sensor
Pin 3	WHT	Alarm loop out from sensor
Pin 4	BLUE	Alarm loop in to sensor
Pin 5	BLK	Enables sensor visible LED

Name the file that you want to play when the motion sensor is triggered "19O.WAV", since the motion sensor output is normally closed, and it opens when it sees motion. To enable playing the sound all the way through when triggered add the 'N' non-interruptible attribute character: "19ON.WAV".

Small Sized Motion Sensor

The photos below show the CFSound-IV wired to the SPY4RTE-B Motion Sensor on contact input #25. Arbitrary Precision sells this sensor online as the "Small Size Motion Sensor".



Pins 1-4	BLK	Speakers
Pin 5	WHT	Contact Input #25
Pin 9	BLK/WHT	+12VDC from transformer +12VDC
	RED	power to motion sensor
Pin 10	BLK	Ground from transformer
	BLK	Ground to motion sensor power

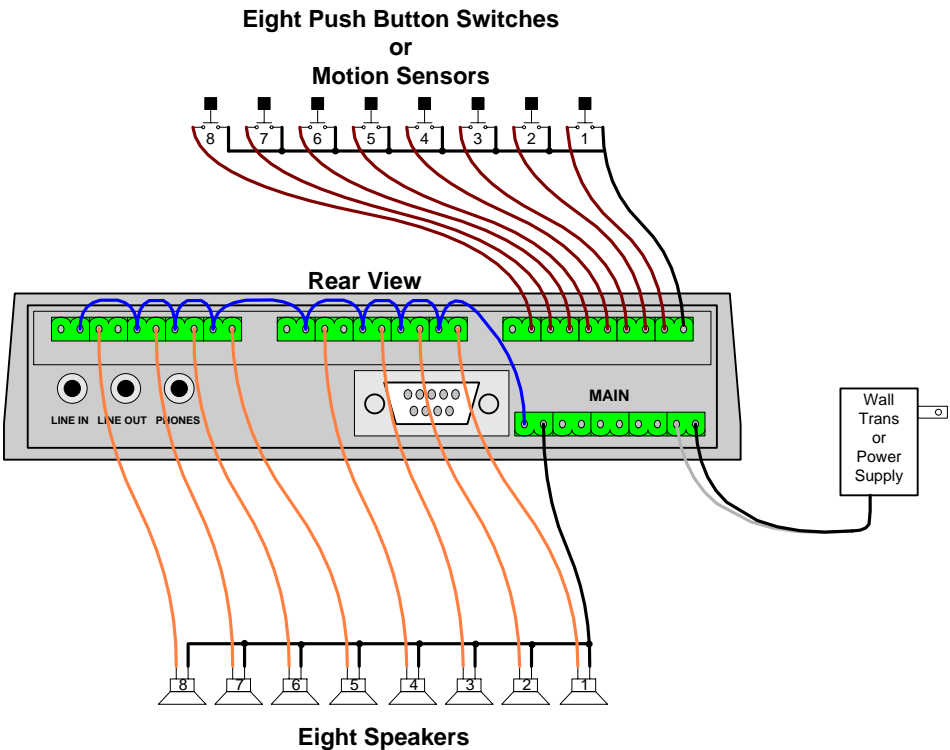
BLK	Ground to motion sensor
WHT	Open collector sensor output
RED	+12VDC power to motion sensor

Name the file that you want to play when the motion sensor is triggered "19C.WAV", since the motion sensor output is normally open, and it closes when the sensor sees motion. . To enable playing the sound all the way through when triggered add the 'N' non-interruptible attribute character: "19CN.WAV".

Connecting Multiple Speakers

The CFSound-IV can play different audio clips to different speakers, one at a time. This requires monaural audio clips and one or two Contact I/O 8 modules.

The sound files have the ‘S’ attribute in the filename to activate the associated output contact. One speaker output is then directed to the desired speaker by connecting it through the output contact. The other side of all the speakers is wired in common but may not be grounded. Here’s a wiring diagram:



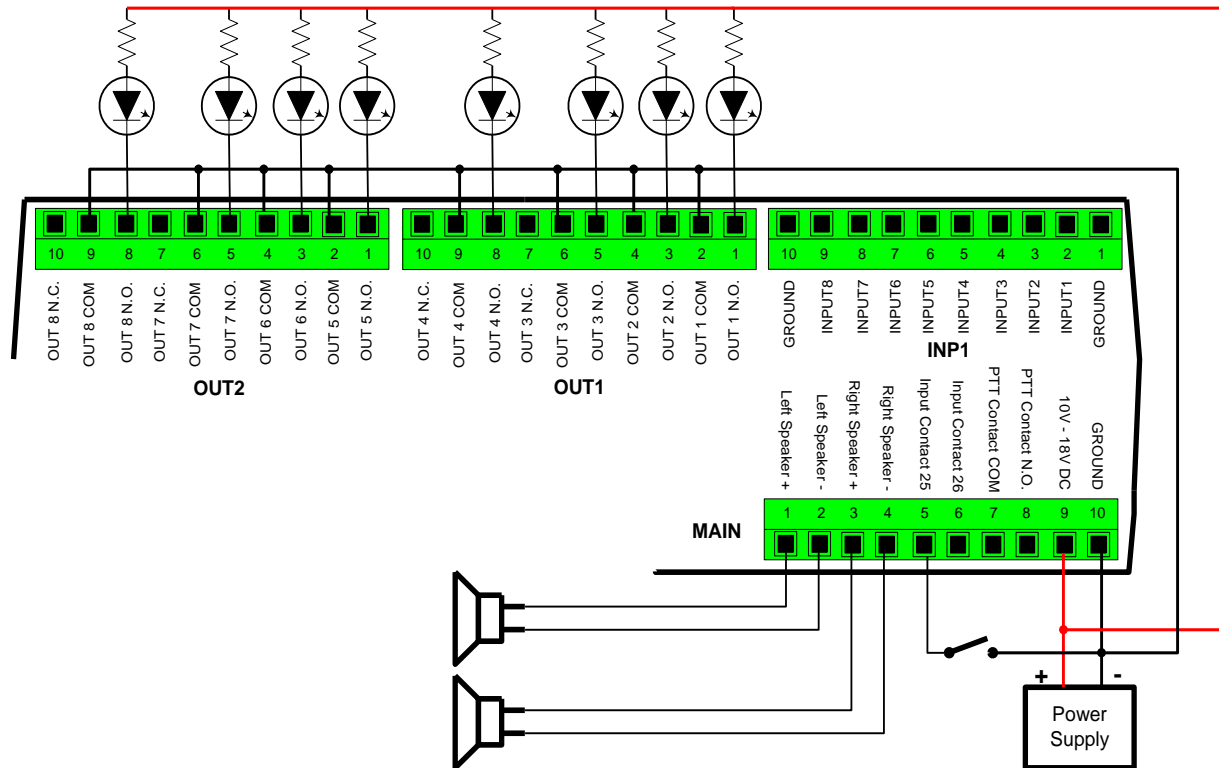
And here’s the table of contact / sound names:

Switch #	Speaker #	Filename
Rear Contact I/O 8	Rear Contact I/O 8	Rear Contact I/O 8
#1(Rear INP1 Pin #2)	#1(Rear OUT1 Pin #1)	#1(01CS.WAV)
#2(Rear INP1 Pin #3)	#2(Rear OUT1 Pin #3)	#2(02CS.WAV)
#3(Rear INP1 Pin #4)	#3(Rear OUT1 Pin #5)	#3(03CS.WAV)
#4(Rear INP1 Pin #5)	#4(Rear OUT1 Pin #8)	#4(04CS.WAV)
#5(Rear INP1 Pin #6)	#5(Rear OUT2 Pin #1)	#5(05CS.WAV)
#6(Rear INP1 Pin #7)	#6(Rear OUT2 Pin #3)	#6(06CS.WAV)
#7(Rear INP1 Pin #8)	#7(Rear OUT2 Pin #5)	#7(07CS.WAV)
#8(Rear INP1 Pin #9)	#8(Rear OUT2 Pin #8)	#8(08CS.WAV)
Front Contact I/O 8	Front Contact I/O 8	Front Contact I/O 8
#9(Front INP1 Pin #2)	#9(Front OUT1 Pin #1)	#9(21CS.WAV)
#10(Front INP1 Pin #3)	#10(Front OUT1 Pin #3)	#10(22CS.WAV)
#11(Front INP1 Pin #4)	#11(Front OUT1 Pin #5)	#11(23CS.WAV)
#12(Front INP1 Pin #5)	#12(Front OUT1 Pin #8)	#12(24CS.WAV)
#13(Front INP1 Pin #6)	#13(Front OUT2 Pin #1)	#13(25CS.WAV)
#14(Front INP1 Pin #7)	#14(Front OUT2 Pin #3)	#14(26CS.WAV)
#15(Front INP1 Pin #8)	#15(Front OUT2 Pin #5)	#15(27CS.WAV)
#16(Front INP1 Pin #9)	#16(Front OUT2 Pin #8)	#16(28CS.WAV)

Controlling Eight LEDs with a Sound Sequence

The CFSound-IV can play a sequence of sounds, one for every button push, turning on a LED associated with each sound. This requires a Contact I/O 8 module.

The LEDs, button and speakers are wired as shown. Select the series resistor for the LED to set the current with the LED forward voltage drop and power supply:



To play a sequence of sounds with associated LED illumination, one every time that the button is pushed, name the sounds 01CSN.WAV, 02CSN.WAV, ... , 08CSN.WAV.

Add a text file **CFSOUND.INI** with the following lines onto the SD card along with the sound files:

```
[Contacts]
SequenceContactNumber=25
FirstSoundNumber=1
LastSoundNumber=8
```

Be sure to leave a blank line at the end of the file so that the CFSound can 'see' the last line of text.

Now, every actuation of the button will play the next sound in the sequence, without being interrupted, and the associated LED illuminates while the sound is playing.

To make the CFSound play the entire sequence of sounds when the button is pushed, add the additional entry to the CFSOUND.INI text file in the **[Contacts]** section:

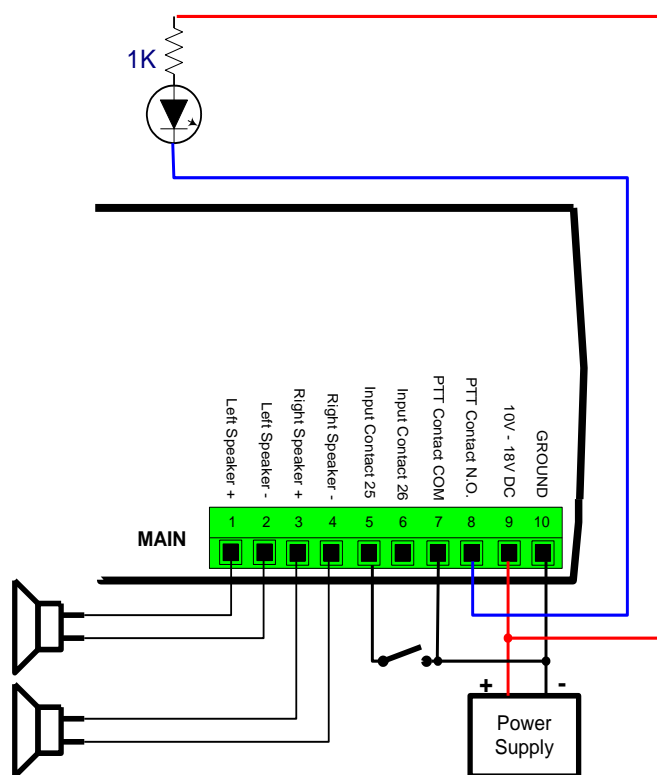
```
AutoplayEntireSequence=TRUE
```

See the *Sequential Sound Payout* section above for more information.

Light LED while Sound is Playing

The CFSound-IV has a PTT (Push-to-Talk) relay that can be optionally activated while a sound is playing. The PTT relay contacts are available on the MAIN connector and can be thought of as a switch that can be closed when a sound is playing.

The LED is wired as shown – note that the PTT contacts are simply wired in series with the LED and its power supply:



Name your sound 19CS.WAV and now whenever the sound is playing, the LED will be lit.

Frequently Asked Questions

Q. The RED LED repeatedly flashes four times at power up?

Q. The RED LED flashes two times when I try to play a sound, and no sound plays? Why is my CFSOUND.INI file not making the CFSound-IV perform using the features I have specified ?:

This is by far our most commonly asked question. 99% of time, it is because your files have double extensions. I.E. "01C.WAV.WAV" or "CFSOUND.INI.INI". Later versions of Windows hide the file extensions in Explorer, so by putting ".WAV" or ".INI" in the filename, you are actually causing a duplicate file extension.

To turn the hiding of file extensions off in Windows XP, do the following:

Click on Start

Then Click on Control Panel

Next, Click on Folder Options

Click on the View tab on the Folder Options Window

Scroll down to "Hide Extensions for Known File Types"

Click on the Check box to the left of it to remove the check mark

Click on the Apply Button

Now the complete file names will be displayed in Explorer, and any folder, and you can remove the duplicate file extensions. Please see the videos on the [SUPPORT](#) page of the [CFSOUND.COM](#) website.

Q. What Inputs and outputs are available on the main CFSound board, without adding I/O Modules ?:

The CFSound main board has a single PTT(Push To Talk) Relay for an output and two contact inputs that appears as Contact #25 & #26 for inputs. Additional Inputs and outputs are added by adding I/O Modules. The CFSound-IV revision 2 can accept one I/O module, the CFSound-IV revision 3 can accept two I/O modules.

Q. I have a motion sensor tied to an old industrial looping tape deck. The first time the sensor is broken it will play the first of three tracks, the second time the same sensor is broken it plays the second track on the tape and the third time it plays the third track. Can your machine duplicate this task ?:

Yes, the CFSound-IV can function as you described. You make three entries into the [CONTACTS] section of the CFSOUND.INI file that you place on the Compact Flash Card.

[Contacts]

SequenceContactNumber=dd

FirstSoundNumber=ff

LastSoundNumber=ll

--> Setting the SequenceContactNumber to anything greater than zero, causes the input # dd to operate in a sequencing mode.

--> FirstSoundNumber sets the first wav file number to be played in the sequence as # ff.

--> LastSoundNumber sets the last wav file number to be played in the sequence as # ll.

You would replace dd with the contact input you wanted to function this way, ff with the first sound that you wanted it to play, and ll with the last sound you want it to play. Every time there is a contact closure on the input specified in dd, it will play the files from ff to ll in sequence, then start over again at ff, and keep repeating. See the Sequential Sound Playout section above for more information.

Q. Can I use a momentary contact closure to initiate a message, but the message needs to stop by using another momentary closure (or open?). So, a momentary to initiate, and a momentary to stop any

message that is in progress ?:

Yes, the CFSound-IV can function as you described. A triggered message that is playing could be 'stopped' by activating another short silent message associated with another contact input. Alternatively a short Acs Basic program can be utilized.

Q. Can the Input ports be momentary, or do they need to remain in a closed or open ?:

It depends upon what you're trying to do. If you want a message that is triggered by a contact closure (or opening) you only need a momentary switch.

If you want the sound to only play while the input is closed (or open) then you would need a latching switch. Sounds are associated with inputs by the filenames of the sounds on the SD card. How they are activated is also controlled by 'decorating' the filenames with additional characters.

For example:

To play a sound whenever input 1 momentary closes, the sound file would be named 01C.WAV.

To play a sound whenever input 2 momentary opens the sound file would be named 02O.WAV.

To play a sound only while input 1 is closed, the sound file would be named 01CP.WAV.

To play a sound only while input 1 is open, the sound file would be named 01OP.WAV.

To repeat a sound only while input 1 is closed, the sound file would be named 01CPR.WAV.

Sounds are normally interruptible whenever another input is activated, but you can also add the character "N" to the filename and make the sound(s) non-interruptible.

You can also control the unit through the RS-232 port to start and stop sounds.

Q. Can a background sound or music be played until another input is triggered ?:

You can have one or more background sounds that play (with an optional delay in between). The background sound is interrupted by the contact input activated sound(s), and resumes (if active) when the contact sound is finished playing.

Q. Can we use basic switches to control our various functions. There are no logic circuits, or anything "hi-tech" in this design ?:

You shouldn't need any external logic circuits. We use 'dry' contact closures for the inputs and provide 'dry' contacts for the output(s). Simple off the shelf switches will work just fine. Water tight / weather proof switches should be used if exposed to elements of weather.

Q. Can individual audio files/channels play simultaneously? i.e. can a file be triggered to play, overlapping a file or files that are already playing ?:

The CFSound cannot play files simultaneously, but it does offer several features that get around this limitation.

1) When the a background sound/file is playing, and it is interrupted by a contact input, it pauses the background sound, plays the sound triggered by the contact input, then continues playing the background sound from where it was interrupted.

2) When sounds are played by an external contact input/trigger, they can be configured by:

Characters in their filename to be:

- a. Not interruptible
- b. Interruptible (the default)

There are settings controllable by entries in the CFSOUND.INI file to:

- a. Store inputs/triggers, and play them when the current sound is finished
- b. Not store inputs/triggers while a sound is being played

Q. I am having addressing problems. With the 'old' units I was able to run a lead from common to 'OnCPR' and have the unit start playing a track "n" automatically on power up. This does NOT happen with these units. It seems there is something wrong with addressing operation. Is there any adjustment I can make ?:

We did indeed make a software change that pre-reads the contact inputs on power-up to prevent automatic sound activation if contacts were already closed. We had a customer complain about this 'feature'.

To support this un-documented feature, we added an additional CFSOUND.INI file entry of:

[Contacts]

Force=TRUE/FALSE

--> Setting this value to TRUE restores the original behavior wherein the contact's active status is 'forced' upon reset, power-up or card-insertion. This will cause associated sound activation if the contact was active.

--> Setting this value to FALSE (the default) causes the new behavior wherein the contact's current status is sampled upon reset, power-up or card-insertion. This will cause no associated sound activation until the contact is re-activated.

Q. Do you have more detailed spec's on your amplifier module ?:

The stereo amplifier is based upon Analog Devices SSM3302 Class D Amplifier and provides 10W per channel to bridge tied speakers (no ground reference).

Q. How do I use the Windows Sound Recorder to make recordings ?:

1. Click on the Start Button, then Programs, then Accessories, then Entertainment, then Sound Recorder.
 2. Click on Files, then Properties. A new window opens named "Properties For Sound".
 3. Change the "Format Conversions" "Choose From" drop down box to "All Formats" or "Recording Formats".
 4. Next, click on "Convert Now", a new window named "Sound Selection" opens.
 5. Make sure "Format" is PCM, 44.1KHz, Stereo or Mono.
 7. Click on the OK Button to close the "Sound Selection" window.
 8. Click on the OK Button to close the "Properties For Sound" window.
 9. To start recording, click on the button with the red dot(record).
 10. To stop recording, click on the button with the black square(Stop).
 11. Once you have finished recording, Click on File, then Save As, and name your file how you want it to play.
 12. I.E. "01B.WAV" to play when there is no switch closures.
 13. I.E. "01C.WAV" to play when input #1 closes.
 14. Copy the file to the Compact Flash Card.
- You're done.

NOTE: If you click on File, then click new, you will have to set the File properties again before recording. Please see the videos on the [SUPPORT](#) page of the CFSOUND.COM website.

Q. Can a output contact be used to operate a lamp to indicate that the message has been triggered ?:

If you need individual lights that illuminate when the associated input is active, you would need the Contact I/O 8 module in the CFSound-IV. If you only need a single light that is illuminated whenever any sound is playing then you can use the PTT relay that is installed on the main unit and use a Contact Sense 8 or Contact Sense 24 module.

Q. We have a client who has a specific need -- an interval timer function. They want their very short message to go off every 30 minutes (not seconds). Does your equipment have a provision for an interval timer ?:

By placing a file named CFSOUND.INI on the compact flash card along with your sound files, you can control several operating features of the unit.

For your application the following entry would provide the 30 minute delay between the background sound:

[Background]

BackgroundDelay=1800

This .INI file option sets the delay in seconds between background sound play-outs to the decimal value of 1800 seconds (30 minutes). The default is 0 seconds (no delay). This is not a high-resolution timer and may be affected by other sounds playing while the timer is active. The specified time is between background sound starts.

Q. We need to have a light associated with an individual "message initiate" momentary switch when that message is playing. Do we need only the Contact I/O 8 module, or do we also need the Contact Sense 8 ?:

The Contact I/O 8 has 8 contact sense inputs and 8 contact outputs. The Contact Sense 8 has 8 contact sense inputs and the Contact Sense 24 has 24 contact sense inputs. The CFSound-IV revision 2 can accept one I/O module, the CFSound-IV revision 3 can accept two I/O modules.

Q. I ordered an enclosed CFSound-IV, and no volume control knob was installed/received ?:

We do not ship a volume knob with the CFSound-IV, since the volume may be adjusted by pushbuttons on the front of the unit.

Q. When wiring the power supply connector to the supplied power supply, which side is plus and which side is ground ?:

Please see the MAIN connector description earlier in the manual for correct power supply wiring.

Q. Will it finish playing the current file even if another input is tripped half way through the file ?:

If you add an "N" in the filenames of your files, it marks the files as non-interruptible. Then optionally setting SaveNIContacts= TRUE in the [CONTACTS] section of the ini file, will cause contact closures occurring during play out of the Non-Interruptible file to be saved until it completes playing, triggering the subsequent file(s).

Q. How long does it take the CFSound-IV to power-up and be ready to play sounds ?:

The unit must either scan the card upon power-up, or scan the card every time a sound is initiated. We opted to scan the card upon power-up to enable each sound to be played quicker. Many of the delays upon power-up are due to having to support the variety of Compact Flash cards on the market - some of which require several seconds upon power-up before they are accessible. The firmware is designed to accommodate the worst case card, and we've had to slow it down in production as people encountered problems with various cards that couldn't be read. There is no firmware modification available that will shorten this delay. A freshly formatted small card with a limited number of sounds will scan the fastest. We have measured this delay at 5-15 seconds.

Q. How long does it take the CFSound-IV to play a sound in response to a contact closure ?:

The unit debounces the contact inputs by sampling them every 20mSEC. It requires a three sample sequence of open, open, closed to qualify a contact input as valid. Then the file must be opened and buffering started. We have measured this delay at ~100mSEC.

Sample Messages

The following sample messages were captured from the CFSound-IV serial port using Windows HyperTerminal. The default connection serial format is 9600 baud, 8 data bits, one stop bit.

Sample Power Up Message, no SD card

```
CFSound-IV #0 v0.1 on May  7 2014 15:14:46
DMX I/O via Art-Net
no SD card

Basic v3.0 May  7 2014 15:14:44
Ready
```

Sample Power Up Message, CFSound Mode

```
CFSound-IV #0 v0.1 on May  7 2014 15:14:46
DMX I/O via Art-Net

Scanning card directory for .WAV files...
Sound Table:
```

#	Sound Filename	Sample Rate	Sound Offst	Sound Length	C	O	P	R	S	B	N	F	D
1	01CS.WAV	44100 Hz	M 44	44,016	C	-	-	-	S	-	-	-	-
2	02CS.WAV	44100 Hz	M 44	40,900	C	-	-	-	S	-	-	-	-
3	03CS.WAV	44100 Hz	M 44	44,016	C	-	-	-	S	-	-	-	-
4	04CS.WAV	44100 Hz	M 44	38,124	C	-	-	-	S	-	-	-	-
5	05CS.WAV	44100 Hz	M 44	44,708	C	-	-	-	S	-	-	-	-
6	06CS.WAV	44100 Hz	M 44	54,760	C	-	-	-	S	-	-	-	-
7	07CS.WAV	44100 Hz	M 44	53,028	C	-	-	-	S	-	-	-	-
8	08CS.WAV	44100 Hz	M 44	39,856	C	-	-	-	S	-	-	-	-
9	09CS.WAV	44100 Hz	M 44	52,684	C	-	-	-	S	-	-	-	-
10	0ACS.WAV	44100 Hz	M 44	40,552	C	-	-	-	S	-	-	-	-
11	0BCS.WAV	44100 Hz	M 44	55,456	C	-	-	-	S	-	-	-	-
12	0CCS.WAV	44100 Hz	M 44	42,284	C	-	-	-	S	-	-	-	-
13	0DCS.WAV	44100 Hz	M 44	51,988	C	-	-	-	S	-	-	-	-
14	0ECS.WAV	44100 Hz	M 44	52,684	C	-	-	-	S	-	-	-	-
15	0FCS.WAV	44100 Hz	M 44	55,108	C	-	-	-	S	-	-	-	-
16	10CS.WAV	44100 Hz	M 44	62,388	C	-	-	-	S	-	-	-	-
17	11CS.WAV	44100 Hz	M 44	63,080	C	-	-	-	S	-	-	-	-
18	12CS.WAV	44100 Hz	M 44	55,800	C	-	-	-	S	-	-	-	-
19	13CS.WAV	44100 Hz	M 44	57,884	C	-	-	-	S	-	-	-	-
20	14CS.WAV	44100 Hz	M 44	45,752	C	-	-	-	S	-	-	-	-
.													
40	28CS.WAV	44100 Hz	M 44	42,632	C	-	-	-	S	-	-	-	-
41	29CS.WAV	44100 Hz	M 44	65,160	C	-	-	-	S	-	-	-	-
42	2ACS.WAV	44100 Hz	M 44	66,200	C	-	-	-	S	-	-	-	-
43	2BCS.WAV	44100 Hz	M 44	72,092	C	-	-	-	S	-	-	-	-
44	2CCS.WAV	44100 Hz	M 44	70,360	C	-	-	-	S	-	-	-	-
45	2DCS.WAV	44100 Hz	M 44	72,440	C	-	-	-	S	-	-	-	-
46	2ECS.WAV	44100 Hz	M 44	70,708	C	-	-	-	S	-	-	-	-
47	2FCS.WAV	44100 Hz	M 44	68,280	C	-	-	-	S	-	-	-	-
48	30CS.WAV	44100 Hz	M 44	57,884	C	-	-	-	S	-	-	-	-
49	31CS.WAV	44100 Hz	M 44	68,628	C	-	-	-	S	-	-	-	-
50	32CS.WAV	44100 Hz	M 44	52,336	C	-	-	-	S	-	-	-	-
51	33CS.WAV	44100 Hz	M 44	70,704	C	-	-	-	S	-	-	-	-
52	34CS.WAV	44100 Hz	M 44	72,440	C	-	-	-	S	-	-	-	-
53	35CS.WAV	44100 Hz	M 44	70,012	C	-	-	-	S	-	-	-	-
54	36CS.WAV	44100 Hz	M 44	78,680	C	-	-	-	S	-	-	-	-
55	37CS.WAV	44100 Hz	M 44	77,984	C	-	-	-	S	-	-	-	-
56	38CS.WAV	44100 Hz	M 44	89,404	C	-	-	-	S	-	-	-	-
126	7EB.WAV	44100 Hz	S 44	306,016	-	-	-	-	-	B	-	-	-
127	7FB.WAV	44100 Hz	S 44	1,164,188	-	-	-	-	-	B	-	-	-

Sample Power Up Message, Basic Mode

```
CFSound-IV #0 v0.1 on May  7 2014 15:14:46  
DMX I/O via Art-Net
```

```
Scanning card directory for .WAV files...  
Basic v3.0 May  7 2014 15:14:44  
Ready
```

SDSound-I
Digital Audio Unit
Copyright(c)1997-2013 by ACS
ACS, Sarasota, FL 34240
ALL RIGHTS RESERVED

SD Card

DOWN UP

RS-232

ETHERNET

PHONES LINEOUT LINEIN

CONTACTS 1 - 24

PNT_EXP1 PNT_EXP2 PNT_EXP3 PNT_EXP4

BT1 BT2 BT3 BT4

BT5 BT6

BT7 BT8

BT9 BT10

BT11 BT12

BT13 BT14

BT15 BT16

BT17 BT18

BT19 BT20

BT21 BT22

BT23 BT24

BT25 BT26

BT27 BT28

BT29 BT30

BT31 BT32

BT33 BT34

BT35 BT36

BT37 BT38

BT39 BT40

BT41 BT42

BT43 BT44

BT45 BT46

BT47 BT48

BT49 BT50

BT51 BT52

BT53 BT54

BT55 BT56

BT57 BT58

BT59 BT60

BT61 BT62

BT63 BT64

BT65 BT66

BT67 BT68

BT69 BT70

BT71 BT72

BT73 BT74

BT75 BT76

BT77 BT78

BT79 BT80

BT81 BT82

BT83 BT84

BT85 BT86

BT87 BT88

BT89 BT90

BT91 BT92

BT93 BT94

BT95 BT96

BT97 BT98

BT99 BT100

BT101 BT102

BT103 BT104

BT105 BT106

BT107 BT108

BT109 BT110

BT111 BT112

BT113 BT114

BT115 BT116

BT117 BT118

BT119 BT120

BT121 BT122

BT123 BT124

BT125 BT126

BT127 BT128

BT129 BT130

BT131 BT132

BT133 BT134

BT135 BT136

BT137 BT138

BT139 BT140

BT141 BT142

BT143 BT144

BT145 BT146

BT147 BT148

BT149 BT150

BT151 BT152

BT153 BT154

BT155 BT156

BT157 BT158

BT159 BT160

BT161 BT162

BT163 BT164

BT165 BT166

BT167 BT168

BT169 BT170

BT171 BT172

BT173 BT174

BT175 BT176

BT177 BT178

BT179 BT180

BT181 BT182

BT183 BT184

BT185 BT186

BT187 BT188

BT189 BT190

BT191 BT192

BT193 BT194

BT195 BT196

BT197 BT198

BT199 BT200

BT201 BT202

BT203 BT204

BT205 BT206

BT207 BT208

BT209 BT210

BT211 BT212

BT213 BT214

BT215 BT216

BT217 BT218

BT219 BT220

BT221 BT222

BT223 BT224

BT225 BT226

BT227 BT228

BT229 BT230

BT231 BT232

BT233 BT234

BT235 BT236

BT237 BT238

BT239 BT240

BT241 BT242

BT243 BT244

BT245 BT246

BT247 BT248

BT249 BT250

BT251 BT252

BT253 BT254

BT255 BT256

BT257 BT258

BT259 BT260

BT261 BT262

BT263 BT264

BT265 BT266

BT267 BT268

BT269 BT270

BT271 BT272

BT273 BT274

BT275 BT276

BT277 BT278

BT279 BT280

BT281 BT282

BT283 BT284

BT285 BT286

BT287 BT288

BT289 BT290

BT291 BT292

BT293 BT294

BT295 BT296

BT297 BT298

BT299 BT300

BT301 BT302

BT303 BT304

BT305 BT306

BT307 BT308

BT309 BT310

BT311 BT312

BT313 BT314

BT315 BT316

BT317 BT318

BT319 BT320

BT321 BT322

BT323 BT324

BT325 BT326

BT327 BT328

BT329 BT330

BT331 BT332

BT333 BT334

BT335 BT336

BT337 BT338

BT339 BT340

BT341 BT342

BT343 BT344

BT345 BT346

BT347 BT348

BT349 BT350

BT351 BT352

BT353 BT354

BT355 BT356

BT357 BT358

BT359 BT360

BT361 BT362

BT363 BT364

BT365 BT366

BT367 BT368

BT369 BT370

BT371 BT372

BT373 BT374

BT375 BT376

BT377 BT378

BT379 BT380

BT381 BT382

BT383 BT384

BT385 BT386

BT387 BT388

BT389 BT390

BT391 BT392

BT393 BT394

BT395 BT396

BT397 BT398

BT399 BT400

BT401 BT402

BT403 BT404

BT405 BT406

BT407 BT408

BT409 BT410

BT411 BT412

BT413 BT414

BT415 BT416

BT417 BT418

BT419 BT420

BT421 BT422

BT423 BT424

BT425 BT426

BT427 BT428

BT429 BT430

BT431 BT432

BT433 BT434

BT435 BT436

BT437 BT438

BT439 BT440

BT441 BT442

BT443 BT444

BT445 BT446

BT447 BT448

BT449 BT450

BT451 BT452

BT453 BT454

BT455 BT456

BT457 BT458

BT459 BT460

BT461 BT462

BT463 BT464

BT465 BT466

BT467 BT468

BT469 BT470

BT471 BT472

BT473 BT474

BT475 BT476

BT477 BT478

BT479 BT480

BT481 BT482

BT483 BT484

BT485 BT486

BT487 BT488

BT489 BT490

BT491 BT492

BT493 BT494

BT495 BT496

BT497 BT498

BT499 BT500

BT501 BT502

BT503 BT504

BT505 BT506

BT507 BT508

BT509 BT510

BT511 BT512

BT513 BT514

BT515 BT516

BT517 BT518

BT519 BT520

BT521 BT522

BT523 BT524

BT525 BT526

BT527 BT528

BT529 BT530

BT531 BT532

BT533 BT534

BT535 BT536

BT537 BT538

BT539 BT540

BT541 BT542

BT543 BT544

BT545 BT546

BT547 BT548

BT549 BT550

BT551 BT552

BT553 BT554

BT555 BT556

BT557 BT558

BT5

CFSound-IV Beta Software Changes

Fixes and Corrections

-

Improvements

- Changed network stack to split large TCP/IP packets instead of sending them twice to circumvent delayed ACK algorithm and improve performance.
- Doubled network stack polling rate from 20Hz to 40Hz.
- Corrected operation of DMX slave mode to allow CFSound sound playout to be controlled via DMX input from ArtNet™.

New Features

- Detect any file system writes via FTP and stop CFSound and/or Basic processes, then reset CFSound after FTP disconnect.
- Added detection of Ethernet link down and force restart of network stack.
- Added configuration items to provide [configurable logging](#) of network protocol events to NETWORK.LOG file.
- Renamed the TCP/IP Raw configuration items to TELNET. Made the TELNET protocol operate full-duplex and send the <IAC> <DO> <suppress go ahead> command upon connection.

Firmware Revisions

Version	Date	Notes
0.01	6-Jun-13	<ul style="list-style-type: none"> Development version for Prototype boards.
1.00	15-Jul-14	<ul style="list-style-type: none"> First release.
1.10	9-Aug-14	<ul style="list-style-type: none"> Added configuration setting via HTTP page. Added official Art-Net and ESTA codes. Restructured code to provide commonality between CFSound-IV and ColorLCD.
1.20	4-Sep-14	<ul style="list-style-type: none"> Corrected handling of paths with sub-directories.
1.2.1	15-Oct-14	<ul style="list-style-type: none"> Fixed bug in handling of baud rate changes via CFSOUND.INI file that would crash the player. Changed configuration via built-in http server to use POST redirect GET for POSTs so browser refresh won't repost. Corrected bug in contact controlled Line Input switching that would occasionally produce white noise instead of sound when resuming interrupted background sound. Upgrade Basic to v3.1.0.
1.2.2	28-Oct-14	<ul style="list-style-type: none"> Changes for internal usage. Upgrade Basic to v3.1.1.
1.2.3	29-Oct-14	<ul style="list-style-type: none"> Changes for internal usage.
1.2.4	30-Oct-14	<ul style="list-style-type: none"> Changes to random number generator. Added advance of random number in SOUND_Start() to improve random sound playout.
1.2.5	22-Nov-14	<ul style="list-style-type: none"> Changes for internal usage. Upgrade Basic to v3.1.2.
1.2.6	8-Dec-14	<ul style="list-style-type: none"> Added lower case support for sound attribute filename characters.
1.2.7	12-Dec-14	<ul style="list-style-type: none"> Upgrade Basic to v3.1.3.
1.2.8	9-Jan-15	<ul style="list-style-type: none"> Upgrade Basic to v3.1.4.
1.2.9	24-Jan-15	<ul style="list-style-type: none"> Fixed a stack overrun when attempting to save more than 9 copies of a file where the first 8 characters of the filename are identical resulting in a short filename tail greater than ~9. Fixed real-time clock binary to bcd conversion of the year resulting in a year value that was three years beyond what was set.
1.2.10	26-Jan-15	<ul style="list-style-type: none"> Upgrade Basic to v3.1.5.
1.2.11	30-Mar-15	<ul style="list-style-type: none"> Corrected playout of sound files that have extra non-audio information at the end. Fixed bug where output contacts on the second expansion slot were not being reset when a new sound was started.
1.2.12	5-May-15	<ul style="list-style-type: none"> Fixed PLAY WHILE operation.
1.3.0	BETA 10-Sep-14	<ul style="list-style-type: none"> Detect any file system writes via FTP and stop CFSound and/or Basic processes, then reset CFSound after FTP disconnect. Added detection of Ethernet link down and force restart of network stack. Added configuration items to provide configurable logging of network protocol events to NETWORK.LOG file.
1.3.1	BETA 11-Apr-15	<ul style="list-style-type: none"> Upgrade Basic to v3.2.0. Changed network stack to split large TCP/IP packets instead of sending them twice to address delayed ACK behavior. FTP server code cleanup/speedup.
1.3.2	BETA	<ul style="list-style-type: none"> Internal version for testing.
1.3.3	BETA 20-Apr-15	<ul style="list-style-type: none"> Upgrade Basic to v3.2.1. Added API for network stack state detection.
1.3.4	BETA 19-May-15	<ul style="list-style-type: none"> Upgrade Basic to v3.2.2. Doubled network stack polling rate from 20Hz to 40Hz.
1.3.5	BETA 21-May-15	<ul style="list-style-type: none"> Upgrade Basic to v3.2.3.
1.3.6	BETA 3-Jun-15	<ul style="list-style-type: none"> Upgrade Basic to v3.2.4.
1.3.7	BETA 4-Jun-15	<ul style="list-style-type: none"> Renamed TCP/IP Raw configuration items to TELNET. Modified TELNET protocol to work full-duplex and send <IAC><DO><suppress go ahead> on connection.

Version	Date	Notes
1.3.8	BETA 4-Sep-15	• Upgrade Basic to v3.2.5.
1.4.0	BETA 1-Oct-15	• Upgrade Basic to v3.3.0.
1.4.1	BETA 1-Dec-15	• Upgrade Basic to v3.3.1.
1.4.2	BETA 4-Jan-16	• Upgrade Basic to v3.3.2.
1.4.3	BETA 2-Feb-16	• Upgrade Basic to v3.3.3.
1.4.4	BETA 29-Feb-16	• Upgrade Basic to v3.3.4. • Fixed FAT file system to correctly support filename extensions of less than 3 characters.
1.4.5	BETA 10-Aug-16	• Corrected operation of CFSound DMX when used as a DMX slave device via ArtNet™.
1.4.6	BETA 17-Aug-16	• Corrected ArtNet™ startup race condition introduced in v1.4.5 that would lockup unit. • Added turn off of Green LED when problem encountered during self-recording. • Improved control of Codec to minimize pops during audio powerup/start/stop, further improved by change to Lattice CPLD program. • Added logic to background sound interrupt/resume to prevent left/right channel flipping on resume.
1.4.7	BETA 23-Aug-16	• Upgrade Basic to v3.3.5.
1.4.8	BETA 9-Sep-16	• Upgrade Basic to v3.3.6.
1.4.9	BETA 21-Nov-16	• Upgrade Basic to v3.3.7.
1.4.10	BETA 30-Nov-16	• Upgrade Basic to v3.3.8.
1.4.11	BETA 6-Feb-17	• Upgrade Basic to v3.3.9.
1.4.12	BETA 4-Mar-17	• Corrected FTP PASV response to report configured value of FTP Port less one instead of hardcoded value of 20.

Upgrading the Firmware

Please see the [CFSound-IV Updating Firmware Manual](#) for instructions detailing how to obtain and update the firmware in the CFSound-IV.

Please Read Carefully:

Information in this document is provided solely in conjunction with Arbitrary Precision products. Arbitrary Precision reserves the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time without notice.

All Arbitrary Precision products are sold pursuant to Arbitrary Precision's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the Arbitrary Precision products and services described herein, and Arbitrary Precision assumes no liability whatsoever relating to the choice, selection or use of the Arbitrary Precision products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license or grant by Arbitrary Precision for the use of such third party products and services, or any intellectual property contained therein or considered a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ARBITRARY PRECISION'S TERMS AND CONDITIONS OF SALE ARBITRARY PRECISION DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ARBITRARY PRECISION PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

ARBITRARY PRECISION PRODUCTS ARE NOT AUTHORIZED FOR USE IN WEAPONS. NOR ARE ARBITRARY PRECISION PRODUCTS DESIGNED OR AUTHORIZED FOR USE IN: (A) SAFETY CRITICAL APPLICATIONS SUCH AS LIFE SUPPORTING OR SYSTEMS WITH PRODUCT FUNCTIONAL SAFETY REQUIREMENTS; (B) AERONAUTICAL APPLICATIONS; (C) AUTOMOTIVE APPLICATIONS OR ENVIRONMENTS, AND/OR (D) AEROSPACE APPLICATIONS. THE PURCHASER SHALL USE PRODUCTS AT PURCHASER'S SOLE RISK, EVEN IF ARBITRARY PRECISION HAS BEEN INFORMED IN WRITING OF SUCH USAGE.

Resale of Arbitrary Precision products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by Arbitrary Precision for the Arbitrary Precision product or service described herein and shall not create or extend in any manner whatsoever, any liability of Arbitrary Precision.

Arbitrary Precision and the Arbitrary Precision logo are trademarks of Arbitrary Precision.
Information in this document supersedes and replaces all information previously supplied.

©1992-2018 Arbitrary Precision – All rights reserved

www.arbitrary-precision.com