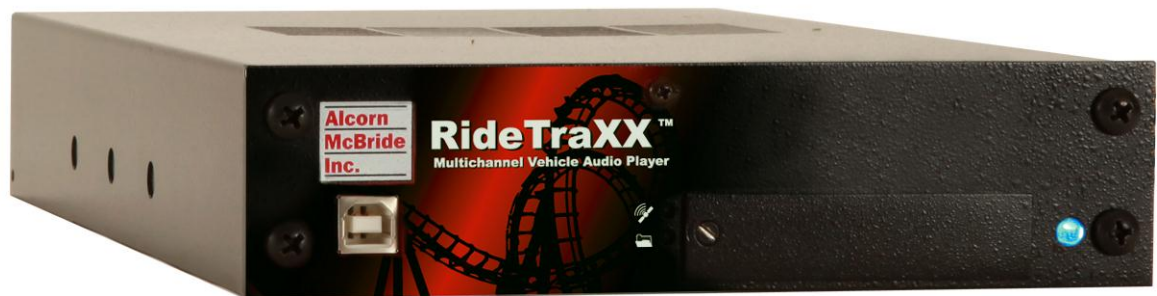


# *RideTraXX*

**16 Channel Synchronous MP3 Playback**

*User's Guide*



**Alcorn**  
**McBride**  
**Inc.**



Every effort has been made to assure the accuracy of the information contained in this manual, and the reliability of the Alcorn McBride RideTraXX hardware and software. Errors can sometimes go undetected, however. If you find one, please bring it to our attention so that we can correct it for others. Alcorn McBride welcomes comments and suggestions on the content and layout of its documentation.

Applications described herein are for illustrative purposes only. Alcorn McBride Inc. assumes no responsibility or liability for the use of these products, and makes no representation or warranty that the use of these products for specific applications will be suitable without further testing or modification. Alcorn McBride products are not intended for use in applications where a malfunction can reasonably be expected to result in personal injury. Customers using or selling Alcorn McBride products for use in such applications do so at their own risk, and agree to fully indemnify Alcorn McBride for any damages resulting from such improper use or sale.

Some or all of the following open source software were used in the development of this product: u-boot, uClinux, ALSA, Busybox, drivers by Analog Devices Inc, drivers by PLX Technology, Inc., drivers by Alcorn McBride Inc., and the traxxPlay Mp3 player.

Please contact us at [swlicense@alcorn.com](mailto:swlicense@alcorn.com) to request a copy of any license or source code for the previously mentioned open source projects.

Proper steps have been taken to separate proprietary portions of this product in order to satisfy license restrictions. Proprietary designs include the application logic (TraxxOS), hardware design, and documentation of this product. The product's hardware warranty will be voided if, at any time, software that is not released in the form of a firmware update by Alcorn McBride Inc. is executed on the hardware.

RideTraXX™ is a trademark of Alcorn McBride Inc., all rights reserved.

Copyright © 2012 Alcorn McBride, Inc. All rights reserved.

Artwork	Gerry Calixto, Jr.
Hardware Design	Jim Carstensen
Mechanical Design	Martin Chaney
Software Design	Hunter Olson
Firmware Design	Adam Rosenberg
Driver Development	Phil Wilshire

**Part Number: 110-104138.50**

**Document Revision: 1.3**

**March 20, 2013**

---

**Alcorn  
McBride  
Inc.**

---

**Alcorn McBride Inc.  
3300 S. Hiwassee Rd.  
Bldg. 105  
Orlando, FL 32835  
Tel: (407) 296-5800  
FAX: (407) 296-5801  
<http://www.alcorn.com>  
[support@alcorn.com](mailto:support@alcorn.com)**



# Table of Contents

<i>Welcome!</i>	1
<i>Features</i>	1
<i>Technical Support</i>	2
<i>Getting Started</i>	3
<b>In the Box</b>	3
<b>Playing a Test Sound</b>	4
<b>Front Panel Controls and Indicators</b>	5
File Access LED	5
Communication LED	5
USB	6
<b>Rear Panel Controls and Connectors</b>	7
Audio Output Connector	8
Power and Discrete Control Input Connector	9
RS-232C Serial Connector	10
Ethernet RJ-45	10
Configuration DIP Switches	11
<i>The Compact Flash Card</i>	13
<b>Recommended Compact Flash Cards</b>	13
<b>Supported File Formats</b>	13
<b>File Naming</b>	14
<b>Playback</b>	15
<b>Crossfade Mode</b>	16
<b>Transferring Files via FTP</b>	18
<i>Control Methods</i>	19
<b>Serial Control</b>	19
<b>Ethernet Control</b>	19
<b>Discrete Input Control</b>	20
Input Groups	20
TraXX Configuration Software	21
<b>Playlist Control</b>	25
Overview	25
Special Playlist Commands	26
Example Playlist	27
Playlist Builder	28
<b>Scheduler Control</b>	29
Schedule Builder	29
<i>Command Protocol</i>	32
<b>Playback Commands</b>	32
Play - PL	33
Loop Play - LP	33
Sync Play - SP	34

Sync Loop - SL	34
Background Search - BS	35
Play Toggle - PT	35
Search - SE	36
Start Crossfade - XF	36
Play Directory - PD	37
Pause - PA	38
Stop - RJ	38
Add Channels to Channel Group	39
<b>Playlists</b>	<b>40</b>
Play System Playlist	40
Play Channel Playlist	40
Stop Channel Playlist	40
<b>Control Commands</b>	<b>41</b>
Version Request	41
Soft Reset	41
Set Name	41
Mute/UnMute/Volume Control	42
Send String	43
Enable/Disable Crossfade Mode	43
Status Request	44
Filename Status Request	44
Completion Acknowledge Mode	45
<b><i>How To Update Firmware</i></b>	<b>46</b>
<b>Standard Firmware Update</b>	<b>46</b>
<b>Firmware Recovery</b>	<b>46</b>
<b><i>Troubleshooting Guide</i></b>	<b>47</b>
<b><i>Specifications</i></b>	<b>48</b>
<b><i>Index</i></b>	<b>49</b>

# Welcome!

Thank you for purchasing The Alcorn McBride RideTraXX™. The RideTraXX replaces CD and solid state audio players in permanent or mobile audio playback installations, providing hours of high-quality and maintenance-free digital audio.

The RideTraXX lets you synchronously play sixteen independent mono tracks from a single, economical unit. It's perfect for ride vehicles, tour vehicles, trams, and background music or effects. Each track can be individually triggered by a number of methods. Discrete inputs – contact closure or voltage – allow direct connection to proximity sensors or switches. The RideTraXX can also be controlled using Ethernet, RS-232 serial, or USB. It can even be programmed to play on a scheduled-basis using its built-in Real Time Clock or execute a pre-programmed sequence of playback operations using a playlist.

RideTraXX offers the latest in Digital Audio Repeater technology, combining MP3 audio with Compact Flash for a rugged, reliable, maintenance-free solution to your playback needs. Access time is nearly instantaneous! All sixteen tracks are completely independent and offer superb audio quality.

# Features

The RideTraXX offers a wide range of features including:

- 16 Independent Mono Audio Channels
- Synchronous Playback
- Hours of High-quality MP3 Audio
- Compact Flash Media Included
- Ethernet or RS-232 Control
- Ethernet File Transfer
- Scheduled Playback using built-in clock
- Playlists for automated operation
- “Boot-and-Loop” playback capability

# Technical Support

You can obtain technical support or other information about specifying, installing, configuring, updating and programming your Alcorn McBride RideTraXX from several sources:

For...	Contact...	When?...
Firmware Updates	<a href="http://www.alcorn.com/support">http://www.alcorn.com/support</a>	Any Time
Knowledge Base	<a href="http://www.alcorn.com/kb">http://www.alcorn.com/kb</a>	Any Time
E-mail Support	<a href="mailto:support@alcorn.com">support@alcorn.com</a>	Any Time
Telephone Support	(407) 296-5800	M-F 9am-5pm (EST)



# Getting Started

Throughout this manual you will find detailed discussions on all the features of this product. However, you're probably anxious to see your RideTraXX play some audio right away. We'll start with a few easy steps to show you how to play audio from the RideTraXX straight out of the box!

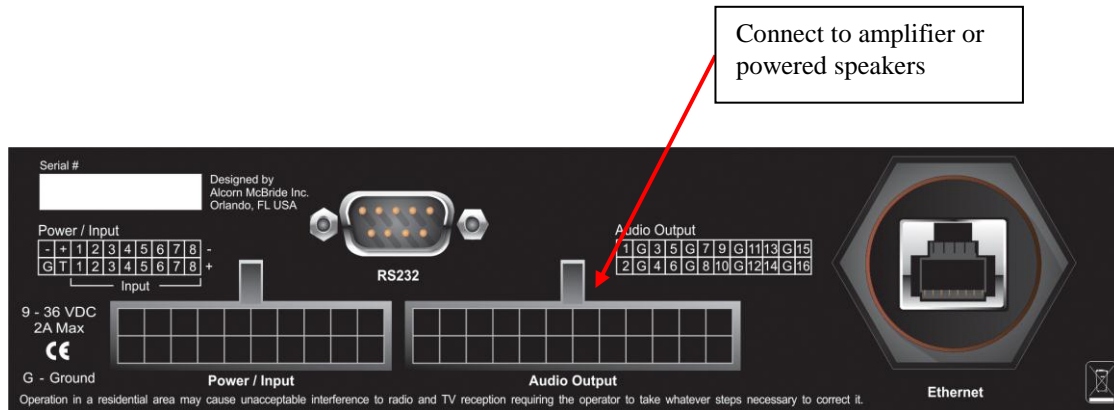
## In the Box

- RideTraXX Multichannel Audio Player (with CF card installed)
- RideTraXX Manual
- Straight-through DB-9 Female to DB-9 Female serial cable
- Crossover Ethernet Cable
- Audio Output Cable Harness
- Discrete Input and Power Cable Harness

# Playing a Test Sound

Every RideTraXX comes with a Compact Flash card preloaded with test sounds already installed in the unit's compact flash slot. To play test sounds, follow these step-by-step directions.

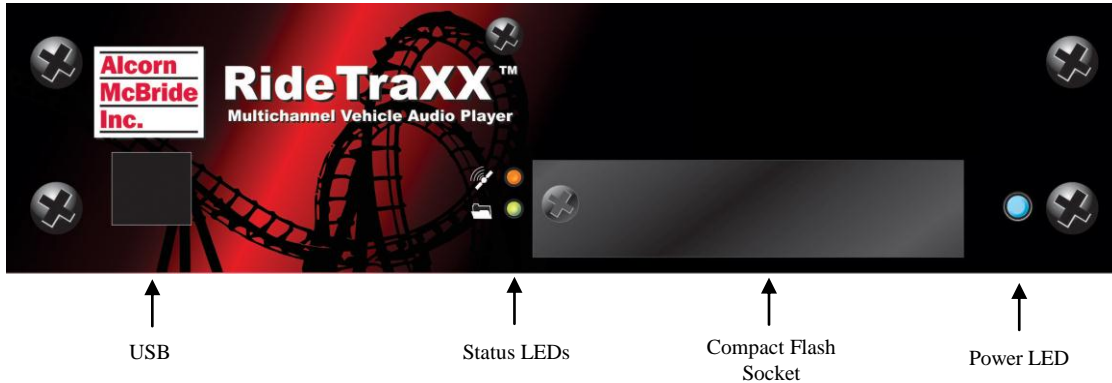
- Using the included Audio Output Cable Harness, connect channel 1 of the RideTraXX to a set of powered speakers or to the input of an amplifier.



- Using either the included Discrete Input/Power cable harness or a custom cable of your own making, connect power to the RideTraXX.
- To quickly test playback, trigger input #1 by touching the Input 1(+) and Input 1(-) wires of the included Discrete Input/Power cable harness. This will trigger playback of file 1 on channel 1.

# Front Panel Controls and Indicators

The RideTraXX includes controls to accommodate a wide variety of installations. The front panel incorporates a blue power indicator, media status/error indicator, an Ethernet status indicator, and a Compact Flash slot covered by a security panel.



## File Access LED



This LED is primarily used to indicate Compact Flash media activity. It illuminates green whenever the Compact Flash media is being accessed. The LED will turn Red if an error condition exists.

File Access LED Behavior	File Access LED Color	LED Meaning
Off	Off	Idle
On	Green	Media activity (Playing, file transfer, etc.)
On	Red	Media Error
On	Orange	Firmware Update (erasing Flash)
Flashing	Green	Firmware Updating
Flashing	Red	Firmware Update Error

## Communication LED



The LED located directly above the File Access LED is used to indicate communication activity. It will blink Green whenever the unit is communicating via Serial or Ethernet. It will blink Red whenever a communication error is detected.

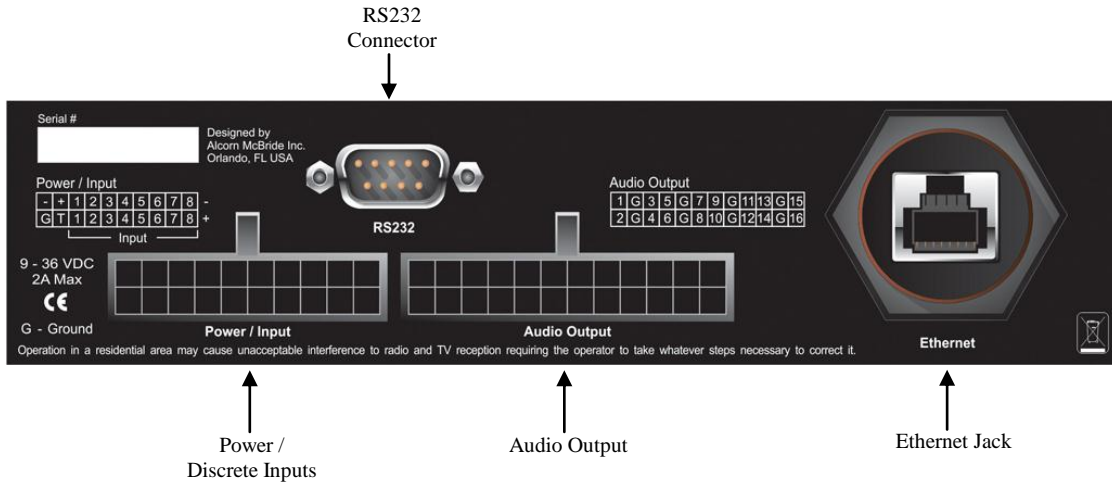
## USB



RideTraXX includes a female type-B USB connector on its front panel. This port can be used to control RideTraXX, using its standard command set, as a USB serial device. While many operating systems will detect and install the driver automatically, some operating systems such as Windows 7 64-bit will need the user to supply the appropriate driver. This driver is available from our website, at <http://www.alcorn.com>.

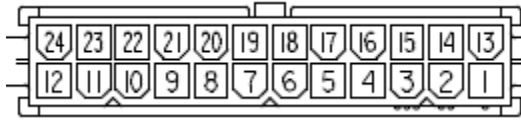
# Rear Panel Controls and Connectors

The rear panel includes Molex type connectors for 16 mono channels of unbalanced audio outputs, Molex control inputs, DB-9 serial RS-232 control connector, and locking RJ45 network jack.



## Audio Output Connector

A 24 pin Molex type connector supplies analog line-level unbalanced audio suitable for connection to a multi-channel amplifier or powered loudspeakers.



Audio Output Pinout			
Pin	Function	Pin	Function
1	Ch. 16	13	Ch. 15
2	Ground	14	Ground
3	Ch. 14	15	Ch. 13
4	Ch. 12	16	Ch. 11
5	Ground	17	Ground
6	Ch. 10	18	Ch. 9
7	Ch. 8	19	Ch. 7
8	Ground	20	Ground
9	Ch. 6	21	Ch. 5
10	Ch. 4	22	Ch. 3
11	Ground	23	Ground
12	Ch. 2	24	Ch. 1

## Power and Discrete Control Input Connector

*Configure the inputs as Voltage Input when triggering over long distances (greater than 10-20 feet).*

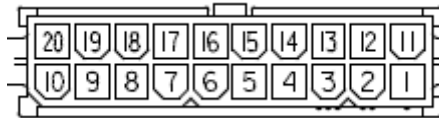
*Inputs configured as Voltage Input trigger on 12-24vdc.*

A 20-pin Molex type connector provides eight discrete control inputs and the power input for the unit.

The control inputs can be configured for either voltage input or contact closure using dipswitch #2 on the side of the chassis. See *Configuration Dipswitches* section of this manual for details.

By default, the rear control inputs are configured to play files 1-8 to channels 1-8. For instance, input 3 will play file 3 to channel 3 when triggered. With the default input configuration, files will play and loop for as long as the input is maintained.

The RideTraXX requires 9-36VDC at approximately 2 Amps.



Input/Power Connector Pinout			
Pin	Function	Pin	Function
1	Input 8 (+)	11	Input 8 (-)
2	Input 7 (+)	12	Input 7 (-)
3	Input 6 (+)	13	Input 6 (-)
4	Input 5 (+)	14	Input 5 (-)
5	Input 4 (+)	15	Input 4 (-)
6	Input 3 (+)	16	Input 3 (-)
7	Input 2 (+)	17	Input 2 (-)
8	Input 1 (+)	18	Input 1 (-)
9	Unused	19	Power(+)
10	Ground	20	Power(-)

## RS-232C Serial Connector

A serial cable is provided with each RideTraXX.

This input is a standard DB-9 male RS-232C connector, intended to be connected to a PC or control system using a straight-through (not null-modem) 9-pin cable. The pinout appears below. Please refer to the *Serial Protocol* section for a list of control commands.



Serial Connector Pinout	
Pin	Function
2	TXD (data from RideTraXX)
3	RXD (data to RideTraXX)
5	Ground

## Ethernet RJ-45

UDP control uses port 2638.

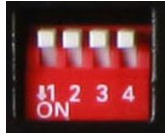
This is a standard RJ-45 jack used to connect the RideTraXX to a network. Control (UDP), file transfer (FTP), and time (NTP) protocols are supported.

FTP uses TCP ports 20, 21.

NTP uses UDP port 123.



## Configuration DIP Switches



Four small DIP switches on the side of the unit are used to configure various hardware related settings.

---

*NOTE: For units that are being installed in equipment racks, it's a good idea to configure the DIP switches before installing the rack mount kit.*

---

DIP Switch Operation		
Number	On	Off
1	Ground Lift Enabled	Ground Lift Disabled
2	Voltage Input	Contact Closure
3	Firmware Recovery Mode	N/A
4	Reset to Factory Defaults	N/A

### **Ground Lift (DIP 1)**

The Ground Lift feature can sometimes help in situations where audio hum or other noise is being caused by potential ground loops. A Ground Lift disconnects the internal ground from the RideTraXX's metal chassis, reducing the chance of ground noise affecting the RideTraXX's electrical systems (i.e. audio outputs/inputs, etc.). This feature is configured using switch #1 on the bank of DIP switches. By default, this switch is set to the OFF position which grounds the RideTraXX to its chassis. You can implement a ground lift by setting this switch to the ON position. After that, the RideTraXX's ground will be isolated from the chassis.

### **Input Mode (DIP 2)**

This switch configures the discrete control inputs for either dry contact closure operation or voltage input operation. For voltage input operation, the input voltage should be at least 12VDC and not more than 24VDC. See the *Discrete Input Control* section of this manual for more details.

### **Firmware Recovery Mode (DIP 3)**

This switch boots the RideTraXX in a recovery mode in which you can update the firmware in the case the unit is unbootable. Use this mode to update to a working firmware version if a regular firmware update fails (due to power failure during update, corrupt firmware file, etc.)

### **Factory Defaults (DIP 4)**

Activating this switch while the RideTraXX is booted will return the unit to its default (factory) state. These defaults are listed below.

<b>Factory Defaults</b>	
<b>Parameter</b>	<b>Default Value</b>
IP Address	192.168.0.254
Subnet Mask	255.255.255.0
Gateway	192.168.0.1
Nameserver (DNS)	127.0.0.1
NTP Server	pool.ntp.org
FTP User	admin
FTP Password	password
Master Volume	100
Legacy Mode	No
NTP	No
Spot Logging	No
Timezone Offset	0
Daylight Savings Mode	None

# The Compact Flash Card

## Recommended Compact Flash Cards

Included with your RideTraXX is a pre-formatted Compact Flash card. The formatting is a standard Microsoft FAT32 format, and any other card you use should be formatted as FAT32 using a PC.

Alcorn McBride recommends SanDisk® brand of Compact Flash. For the latest list of compatible and recommended Compact Flash please visit:

<http://www.alcorn.com/library/appnotes/HowMuchFlash/index.html>

## Supported File Formats

The RideTraXX plays MP3 files which are compliant to the following specifications:

Supported MP3 Formats	
Standard	MPEG-1 Part 3 (or simply, MP3) defined in ISO/IEC-11172-3
Channel Encoding	Mono, Joint Stereo, Stereo, Dual
Sample Rates	44.1KHz, 48KHz
Bitrates	32,48,56,64,80,96,112,128,160,192Kbps

Note: When playing more than 10 files simultaneously, 128Kbps is the maximum supported bitrate.

# File Naming

*It is sometimes easiest to use the “numbered” style of file naming for MP3 files if you want to use the discrete inputs.*

*Details about Playlists and Schedules are contained in separate chapters of this manual.*

A specific file naming convention is used so that the RideTraXX can identify the file type, as well as associate the supported file types with a number. Here is a table showing the naming conventions based on the type of file that is being named:

File Type	File Naming Style
MP3 Audio File (numbered)	SNDnnnnn.MP3
MP3 Audio File (named)	any-filename.MP3
Playlist	PLYnnnnn.LST
Schedule	SCHEDULE.SCL

Files can be referenced in commands by their full filename or their file number. The file number is determined by the 5-digit ‘**nnnnn**’ portion of the file name. The RideTraXX supports files that are assigned numbers between 1 and 99999, so this part of the filename is always represented as **00001** through **99999**.

Just as an example, let’s say that you created an MP3 file that you want to associate as file #123. To do this, you would need to rename the MP3 file to **SND00123.MP3**.

The file number naming style is not needed if you plan to reference files in commands by their full filename.

The one exception to the numbering method is the schedule file. The RideTraXX only supports a single schedule, and therefore there is only one file name permitted for this file. As you can see in the table above, this filename is **SCHEDULE.SCL**.

---

Note: If you use the file numbering scheme, every file on the media must have a unique number in the file name, no duplications are allowed. If you have duplicate file numbers (i.e. such as a playlist and a MP3 file numbered as #1), the RideTraXX would consistently choose only one of those files when told to play the associated number.

---

# Playback

## ***Synchronous Playback***

RideTraXX is capable of sample synced audio playback on all 16 channels.

To start synchronous playback, you first send search commands to queue each file on its desired output channel. Once all of your desired channels are searched(queued), sending an **SP**(Sync Play) or **SL**(Sync Loop) command will start playback of all searched files.

For example, if we wanted to start synchronous playback of files 1-4 on channels 4-7 respectively, we would send the following series of commands:

```
1C4SE  
2C5SE  
3C6SE  
4C7SE  
SP
```

After the **SP** command is received, playback will start in exactly 3 seconds (+/- 10ms). No delay is required between the search commands and the **SP** command. To execute the required actions with a single command, the search and sync play commands can be put into a playlist, and the playlist started as the System Playlist.

More details on these commands can be found in the *Command Protocol* section of this manual.

---

Note: During the 3 seconds between when a **SP** command is received and the audio playback is started, all the audio outputs of RideTraXX will be silent.

---

## ***Interruptible versus Uninterruptible Operation***

In a play command, you can specify uninterruptible playback, meaning that repeated play requests will not cause the sound to start over from the beginning while playing. An example of where this might be used is in a museum kiosk, where a guest trips a break-beam sensor to start an audio presentation. The audio will continue playing through, even though the guest might continually trip the sensor while listening to the presentation.

If not specified otherwise in the play command, files play as interruptible, meaning that repeated play requests during playback will cause the sound to start over. An example of where this might be used is in a shooting gallery, where you want the sound to be heard every time a gun's trigger is pulled.

See the Command Protocol section for details on how to specify interruptible vs. uninterruptible mode in your commands.

## **Stereo vs. Mono MP3 Playback**

RideTraXX is capable of up to 16 independent mono channels, but can also be used to play stereo MP3 files. When specifying playback of a stereo MP3 in a command, play the file to a channel pair (P1-P8) rather than an individual channel (C1-C16). RideTraXX will automatically send the left channel of the MP3 to the odd channel in the pair, and the right channel to the even channel in the pair.

For example, sending command “myMonoFile.mp3”C2PL would play the MP3 file to the Channel 2 output. The command “myStereoFile”P2PL would play to channel pair #2, containing channels 3 and 4, perfect for stereo files.

More details can be found in the *Command Protocol* section of this manual.

## **Crossfade Mode**

RideTraXX is capable of crossfades between two MP3 files on a channel. As one file ends, it begins to fade out. At the same time, the next MP3 file fades in. This allows for smooth transitions between clips where required. In order to Crossfade, RideTraXX must be placed into “Crossfade Mode”. **While in Crossfade Mode, RideTraXX functions as an 8-Channel player.**

---

Note: If channels are playing back synchronously, channel sync will be maintained during and after the crossfade.

---

### **Putting RideTraXX in Crossfade Mode**

RideTraXX can be put into Crossfade Mode using a variant of the Crossfade command (see *Command Protocol* section of this manual for complete details).

For example, “E0XF”h0d turns Crossfade mode off, and “E1XF”h0d turns Crossfade mode on. This setting is saved and will be restored even after a power cycle of the unit.

RideTraXX operates as an 8 channel player while in Crossfade Mode with the 8 channel pairs of RideTraXX representing the 8 usable channels in crossfade mode. The audio outputs of each channel in each channel pair will be identical. For example, channels 1 and 2(channel pair 1) of RideTraXX will output the same audio. While either channel of a channel pair can be used as the audio output, it is probably easiest to use either all of the odd channels (1,3,5,7,9,11,13,15) or all of the even channels (2,4,6,8,10,12,14,16) as the 8 crossfade channels.

Crossfade commands are issued to channel pairs, but remember; in Crossfade Mode a channel pair represents one channel. Playback of stereo MP3’s is not supported in Crossfade mode. You should separate your channels into up to 8 mono MP3’s to take advantage of the 8 channels of synchronous crossfading audio available while in crossfade mode.

## ***Using the Crossfade Command***

There are two main steps to initiating a crossfade one a channel or set of channels:

- 1) Once the initial file(s) is playing, queue up the next file(s) using the “Background Search” command.
- 2) Initiate the crossfade using the Crossfade Command. The Crossfade command allows you to specify the length of the crossfade.

## ***A Crossfade Example***

Let’s look at a scenario where you have two channels of synchronous audio playing on channel pairs 1 and 2. These two channels of audio are MP3 files “ch1scene1.mp3” and “ch2scene1.mp3”. When you issue the crossfade command to RideTraXX, you want a 5 second crossfade to the MP3 files “ch1scene2.mp3” and “ch2scene2.mp3”. The following commands would be issued:

*Start the initial synchronous playback on channel pairs 1 and 2.*

**“ch1scene1.mp3”P1SE**

**“ch2scene1.mp3”P2SE**

**SP**

*At any point during playback, queue the next files in preparation for the crossfade.*

**“ch1scene2.mp3”P1BS**

**“ch2scene2.mp3”P2BS**

*When the crossfade should begin, issue a command for a 5000ms crossfade.*

*We send this command to all channels(P\*) and any channels with queued files will start crossfading. Channels without queued files are unaffected.*

**5000P\*XF**

# Transferring Files via FTP

In addition to transferring the files to the RideTraXX's compact flash card directly with your PC, RideTraXX has a built-in FTP server allowing for quick file transfers over Ethernet.

Before using FTP, be sure that the network settings on your RideTraXX are configured correctly. The default network settings are listed below.

*The nameserver setting is used to resolve hostnames into IP addresses. If you will not be using domain names (for NTP server addresses, etc.) you need not change this value from the default.*

IP Address	192.168.0.254
Subnet Mask	255.255.255.0
Gateway	192.168.0.1
Nameserver (DNS)	127.0.0.1
FTP User	admin
FTP Password	password



# Control Methods

For your convenience, Alcorn McBride distributes an application called **AMI Terminal**, which allows you to control the RideTraXX via RS-232 or Ethernet with the click of a button. This free application can be obtained at <http://www.alcorn.com/support/software.html>.

## Serial Control

The RideTraXX can be controlled using the male DB-9 RS-232 connector on the rear panel. If you are interfacing to the unit with a PC or other type of controller, you will need to use a straight-through DB-9 Female to DB-9 Female serial cable. For your convenience, this cable has been included with your RideTraXX.

This serial interface operates at **9600 baud, no parity, 8 bits per character, and one stop bit (9600,8,N,1)**.

All serial commands are ASCII based, and **are terminated with a carriage return** (hexadecimal value 0x0D). The complete list of supported commands can be found in the **Command Protocol** section of this manual.

## Ethernet Control

The RideTraXX is equipped with an Ethernet interface that allows you to control the unit using a standard Ethernet Protocol called UDP (User Datagram Protocol).

UDP is a very simple Ethernet protocol that allows an Ethernet device to transmit a generic string of data to another Ethernet device. Since UDP is such a common standard, UDP packets can be generated easily by a large variety of controllers and PC applications. The UDP packet handles all of the complexities of getting the message where it needs to go in one piece. The good news is that you do not have to understand those complexities. As a matter of fact, you only need to know the following information to send a UDP message.

1. **Destination IP Address** – This needs to be the IP address of the RideTraXX you intend to control. The default IP address of the RideTraXX is **192.168.0.254**.
2. **Destination Port** – This value needs to be the UDP port that the RideTraXX is listening to for UDP commands. This value is always **2638**.
3. **Source Port** - This is the UDP port the PC or controller will use to send the UDP message. It does not matter what port you choose, as long as it is a legal UDP port. If you have no idea what a legal UDP port is, just use **2638**.
4. **Data** – This is the actual message that you are sending to the RideTraXX. This message needs to be a **Command Protocol** message (i.e. 5C1PL<CR>).

All UDP commands are ASCII based, and **are terminated with a carriage return** (hexadecimal value 0x0D). The complete list of supported commands can be found in the **Command Protocol** section.

# Discrete Input Control

The RideTraXX has a total of 8 discrete inputs on the rear of the unit which can be set to accept Voltage Input (12-24VDC) or Contact Closure. Voltage Input or Contact Closure is selected by configuration dipswitch #2 on the side of the RideTraXX chassis. See the **Configuration Dipswitch** section of this manual for configuration details.

When configured for contact closure, an input is triggered when a connection is made between the positive and negative terminals of that input. No external power supply is needed, so a simple switch or button wired between the terminals can be used to trigger the input. Contact Closure should not be used over long distances, anything over 15 feet should use inputs configured for Voltage Input.

When RideTraXX is configured for voltage input, an external power supply should be used to provide a 12-24vdc signal to the RideTraXX. Voltage Input can trigger from much longer distances than contact closure and because the inputs aren't sharing the power supply of the RideTraXX (as with contact closure), the wiring can be further isolated.

When triggered, inputs execute whatever standard control command has been assigned to them. TraXX Config, the configuration software for your RideTraXX, can be used to assign commands (such as play, stop, etc) to your RideTraXX's discrete inputs. See the **Command Protocol** section of this manual for a list of supported commands.

By default, the rear control inputs are configured to play files 1-8 to channels 1-8. For instance, input 3 will play file 3 to channel 3 when triggered. With the default input configuration, files will play and loop for as long as the input is maintained.

## Input Groups

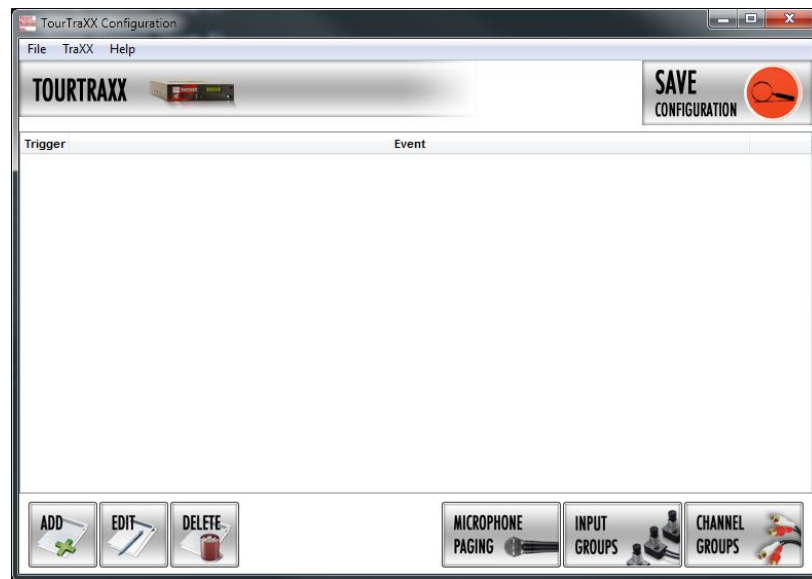
If you need more than 8 discrete inputs for your application, you can set up an Input Group. Input Groups are an advanced feature of RideTraXX that allows you to use combination of inputs to trigger up to 255 different actions. Once you use the TraXX Config software to add inputs to a group, the input group can be assigned to respond to different combinations of inputs triggered simultaneously.

For example, if you add all 8 inputs to input group 1, you can assign commands to respond to the 8bit numbers created by the 255 different combinations of inputs.

# TraXX Configuration Software

TraXX Config can be downloaded from <http://www.alcorn.com/support/software.html>

TraXX Config allows you to easily group inputs together, assign them to specific playback or operational commands, and configure them for active-on or active-off operation.

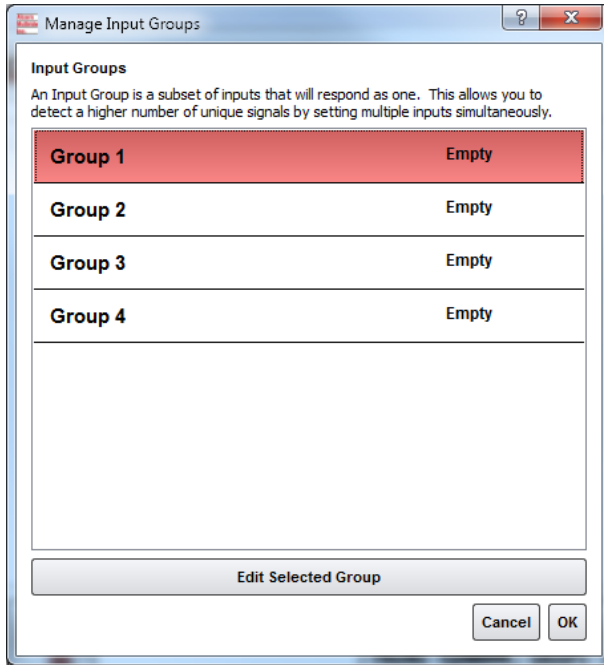


## Manage Input Groups

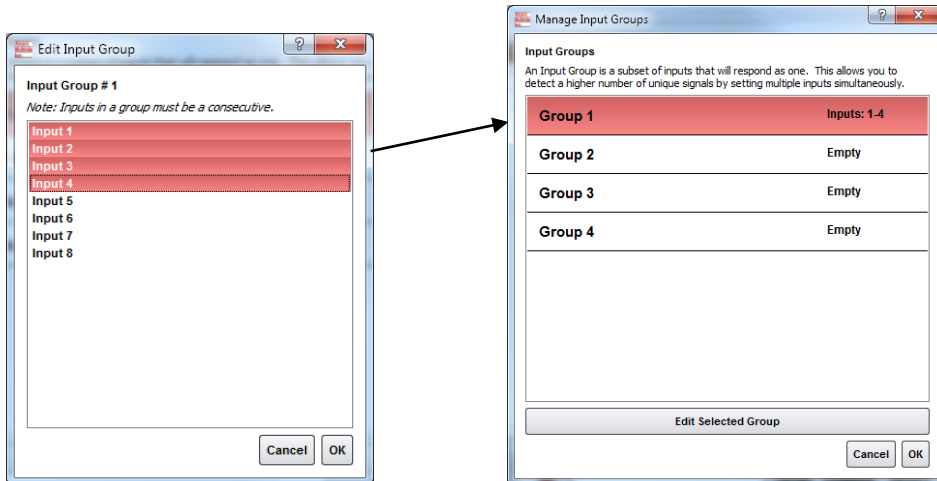
By default, the 8 discrete inputs on the RideTraXX act individually and are constrained to a total of 8 separate operations – one for each input.

However, if you need to perform more than 8 operations using the parallel inputs, you can combine the 8 inputs into “Input Groups” so that they can be activated using binary codes to achieve up to 255 operations.

In the TraXX Config program select *Input Groups* to open this window:



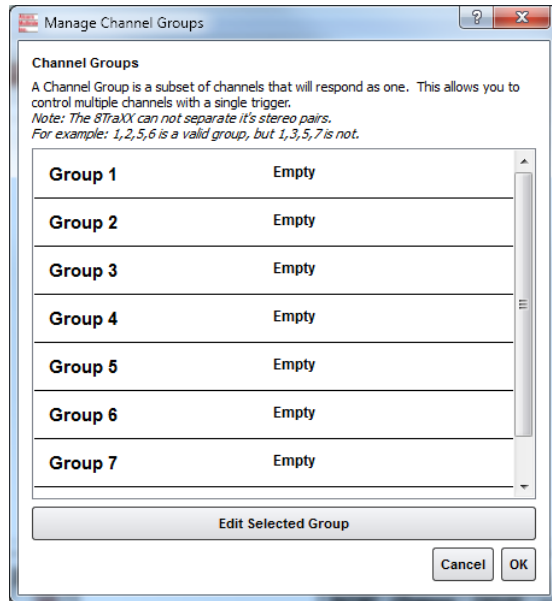
Select one of the Groups and click *Edit Selected Group* to assign inputs to that group. For this example we have assigned Inputs 1-4 to Group #1. Click *OK* to save the assignments.



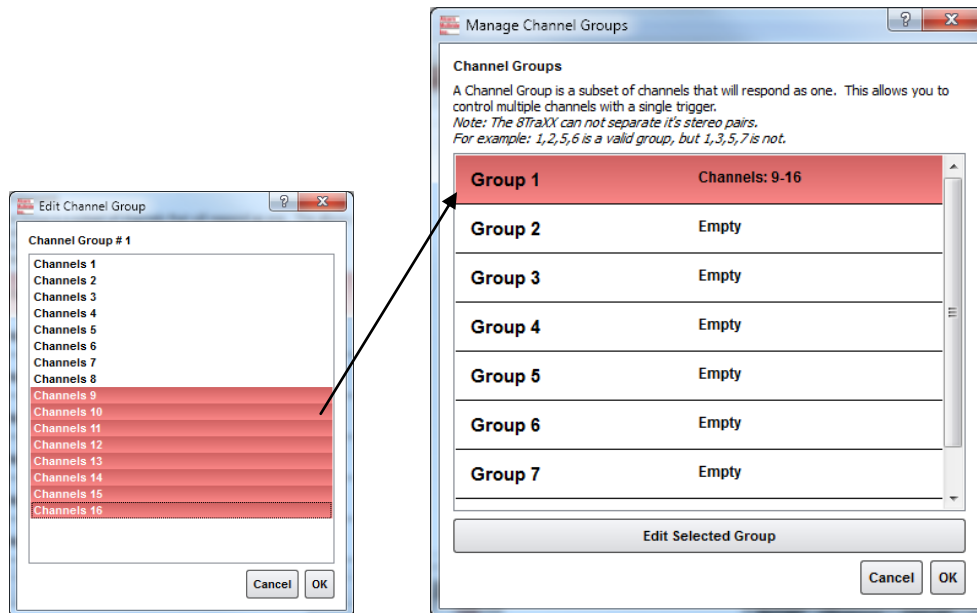
## Manage Channel Groups

You can have a total of 13 *Channel Groups* in the RideTraXX. Channel Groups allow you to control multiple audio channels using a single input or Serial/Ethernet command.

In the TraXX Config program select *Channel Groups* to open this window:

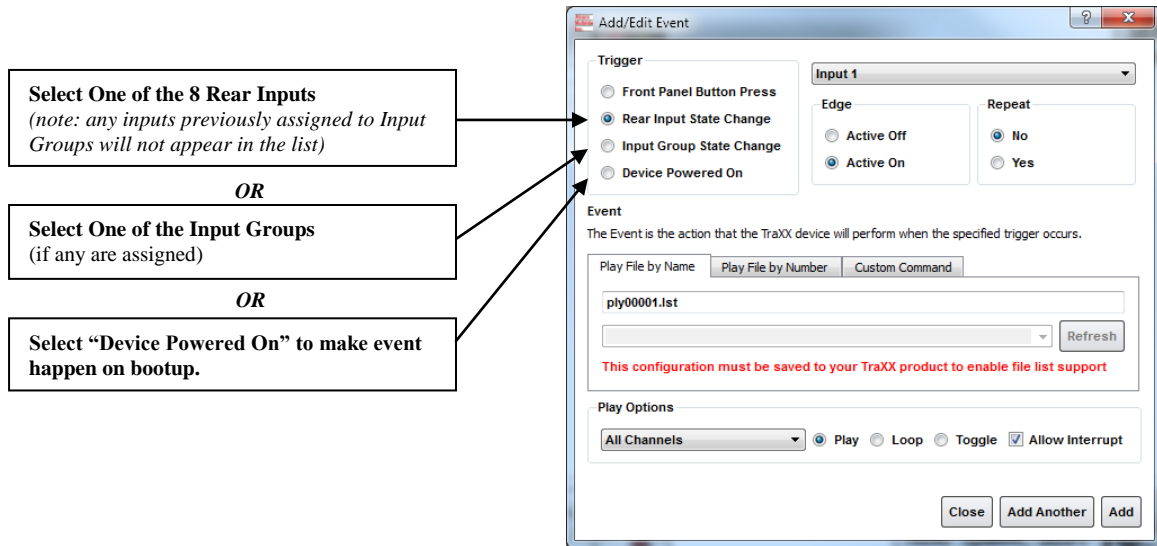


Select a Group and click *Edit Selected Group* to assign channels to that group. For this example we have assigned channels 9-16 to Group #1. Click *OK* to save the assignments.



## Add/Edit Event

Now for the fun part! Click on *Add* to add a trigger event to one of the inputs, buttons, or input groups. The Add/Edit Event window will appear.



**The best way to make the RideTraXX to play a file on startup is to select the *Device Powered On* Trigger Type.**

Once you have chosen where the event is being triggered from, you can select either an *Active On* or *Active Off* Trigger Edge. This allows the RideTraXX to respond to either normally-open or normally-closed input sources.

In the *Event* field you specify exactly what action you want the Event to perform. You can have it play a file or you can specify a *Custom Command*, where you can use any command out of the Command Reference in this manual.

*The Configuration File is named "CONFIG.LST" which the RideTraXX always loads on startup.*

Click *Add* once you've finished making your choices and the trigger will be added to the RideTraXX configuration file.

Now you need to save your file. To do this select *Save Configuration* and choose the RideTraXX you want to send this configuration to. If you don't have access to the RideTraXX at the time, you can also save the configuration to your hard drive and then later copy it to the RideTraXX Compact Flash and insert it into the unit.

# Playlist Control

*You can have a total of 17 simultaneously-running playlists in the RideTraXX. One system playlist plus one per channel.*

Playlists provide a mechanism for the RideTraXX to automatically execute commands in a predefined sequence. Since Playlists can execute any of the commands listed in the **Command Protocol** section, they can perform a variety of functions. These functions could be something as simple as looping a single channel or groups of channels, or as complex as user-initiated pacing of playback.

## Overview

*For an easy way to make a playlist, use **Playlist Builder** described on Page 28 and available at the software section of <http://www.alcorn.com>*

*A playlist zero (PLY00000.LST) will automatically play on power-up.*

*It is not a good idea to have both a playlist file and an audio file with the same number (like PLY00002.LST and SND00002.MP3). If you do, commanding the RideTraXX to play file #2 may not produce the desired result (whatever that may be) since the RideTraXX is forced to prioritize which file type it should play in that circumstance.*

**A playlist file is nothing more than a list of RideTraXX commands listed in a text file.** These commands, which are listed in the **Command Protocol** section, are the same commands that are used by all of the RideTraXX's control interfaces. Although any of the supported commands can be used, playlists usually contain **Play** commands, which cause the RideTraXX to play audio clips from the Compact Flash card. When a playlist is being used, the commands are processed in the order in which they appear in the list. The playlist is an ASCII text file and can be made using any ASCII text editor (like notepad) or using the Playlist Builder program available from <http://www.alcorn.com/support/software.html>.

Each channel of the RideTraXX can have its own independent playlist. There is one additional system playlist that is not tied to any particular channel.

Playlists are triggered the same way as sound files - by Serial or Ethernet commands, or with discrete inputs. The system playlist is started by a **PL** command and the individual channel playlists are started with **PO** commands.

For example, the following serial command would start the playlist file "ply00002.lst" as the system playlist:

**2PL**

The following command would start the playlist file "ply00004.lst" as the channel 5 playlist:

**4C5PO**

It is a very common requirement in applications for the RideTraXX to power up and launch a playlist on its own. This can easily be done by naming the playlist file **PLY00000.LST**. This filename directs the RideTraXX to play the file as the system playlist upon power up.

It is also common practice to have one playlist launch another playlist. This is done in the same fashion as playing an audio file; by using a command like **2PL** (where 2 is the number of another playlist).

## Special Playlist Commands

In addition to regular RideTraXX commands listed in the **Command Protocol** section, there are a few additional commands for the Playlist Control interface. These single-character commands further modify the behavior of the playlists to provide additional features that only apply to the Playlist Control interface of the RideTraXX. These characters and their functions are listed in the following table:

Command	Description
<b>W</b>	Wait. Playlist pauses in the current position until another 'Play' command occurs, at which time it resumes processing of the playlist. This is useful for applications where a user interactively controls the progression of play.
<b>&lt;</b>	Loop. Tells the playlist to loop upon completion. This looping will continue indefinitely until a Stop condition occurs, or another audio or playlist file is played on that channel.
<b>;</b>	Comment. Ignore all characters until the next line.

Commands may be entered into the Playlist in either upper or lower case. All text following a semicolon (;) is interpreted as comments, until the next line in the file. You may put anything you want in comments, including the special command characters normally used in the Playlist.



## Example Playlist

Here's an example playlist that will be discussed in detail below to explain the concept of how playlists work in the RideTraXX.

```
1C1PL ;You can comment your Playlist file using
6P2LP ;semicolons. All text after a semicolon is
3G2PL ;ignored until the end of the line where a
      ;carriage return is encountered.
5P1PL
<
```

The first command, which is “**1C1PL**”, causes the RideTraXX to play file #1 to channel #1. If the user issues a Play command during the playback of file #1, it will be interrupted and the playlist will move on to the next command. If a Play command is not received, the RideTraXX will not advance to the next playlist command until file #1 has played to completion.

When file #1 has finished playing, the “**6P2LP**” command is executed. This command causes the RideTraXX to loop file #6 continuously to channel pair #2. This file will continue looping indefinitely until the user issues a Play command to advance the playlist.

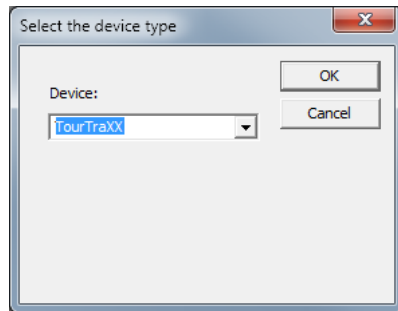
If advanced by the user, file #3 plays to channel group #2 followed by file #5 playing to channel pair #1. Once file #5 finishes, the “<” character causes the playlist to loop back to the beginning. This means that file #1 will start playing once again.

## Playlist Builder

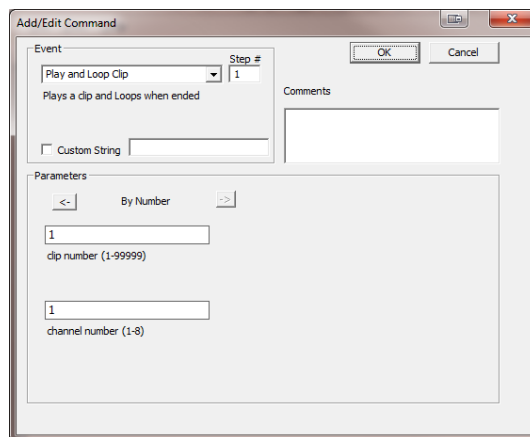
For convenience, Alcorn McBride provides a free Windows application for making playlists using a graphical interface. This program is called Playlist Builder, and it can be downloaded from the software section the Alcorn McBride website at <http://www.alcorn.com/support/software.html>

Here is a step-by-step guide to get started making playlists with Playlist Builder.

1. Open the Playlist Builder application
2. Select “RideTraXX” in the list of supported devices and click OK.



3. Select the event you wish to insert into the playlist. For this example we'll choose “Play and Loop Clip”.
4. Using the arrows in the ‘parameters’ box, you can select which type of Play command you wish to use. For example, “By Number” would allow you to enter a clip number. “By File” would allow you to enter the actual file name. If you are using the compact flash card that shipped inside your RideTraXX, select ‘By Number’ and enter the number ‘1’ in the field.
5. Also choose which channel pair you would like the file to play to. For this example we will choose Channel Pair 1.
6. Click OK to add the event to the playlist



7. Add as many commands as you like! Once you are done, you can save the playlist file and transfer it to the RideTraXX Compact Flash card!

# Scheduler Control

For an easy way to make a schedule, use **Schedule Builder** available at <http://www.alcorn.com/support/software.html>

If you plan on using the Scheduler, you should verify that the RideTraXX's time, date, timezone, and daylight savings mode are set correctly.

The RideTraXX has a built-in real-time clock, which can be used to trigger events based on time and/or date. For example, a schedule could be created to play a sound at exactly 3:00PM on January 15th 2015, or start looping a sound at 8AM every morning and stop looping it at 8PM every night.

The RideTraXX can accomplish this type of task using a feature called the **Scheduler**. With a user-created schedule file, the Scheduler can execute any event listed in the **Command Protocol** section of this manual. These scheduled events are accurate down to the minute, and can be programmed to execute once, or repeat periodically (i.e. repeat every Monday, every January 1<sup>st</sup>, every morning, etc.).

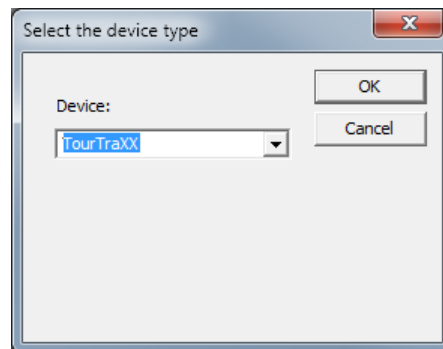
The RideTraXX obtains schedule information from a file on the Compact Flash card named **schedule.scl**. This file is created using either a text editor (i.e. Notepad) or a free Alcorn McBride application called **Schedule Builder**. The remainder of this section contains information on how to make this schedule file using either method.

## Schedule Builder

To make creating schedules more convenient, Alcorn McBride provides a free Windows application for making schedules using a graphical interface. This program is called **Schedule Builder**, and it can be downloaded from the software section the Alcorn McBride website at <http://www.alcorn.com/support/software.html>

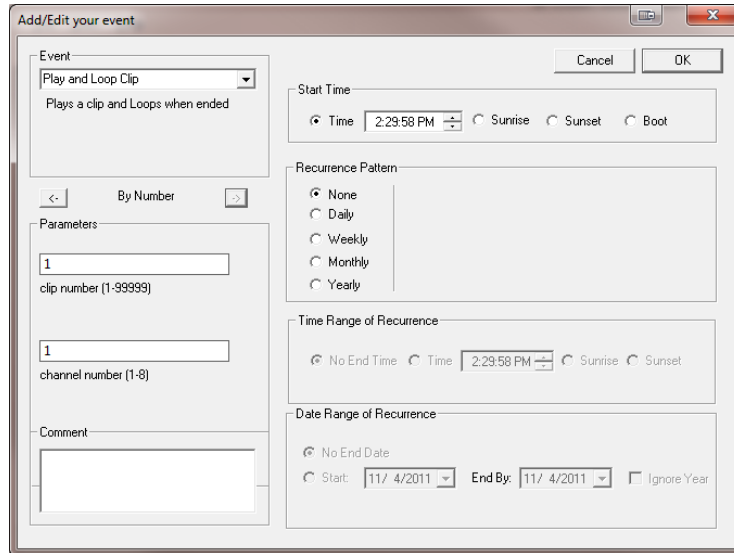
Here is a step-by-step guide to get started making schedules with Schedule Builder.

1. Open the Schedule Builder application
2. Select the RideTraXX in the list of supported devices and click OK.

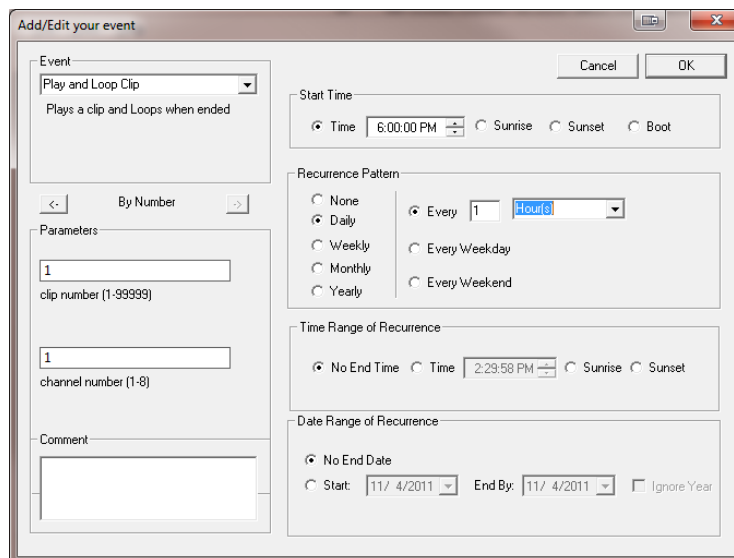


3. The Add/Edit Event window will appear automatically. This is where you select the event you wish to insert into the schedule. Since "Play and Loop Clip" is an event used to play an audio file, we will use it for this example.
4. Using the arrows in the 'parameters' box, you can select which type of Play command you wish to use. For example, "By Number" would allow you to enter a clip number. "By File" would allow you to enter the

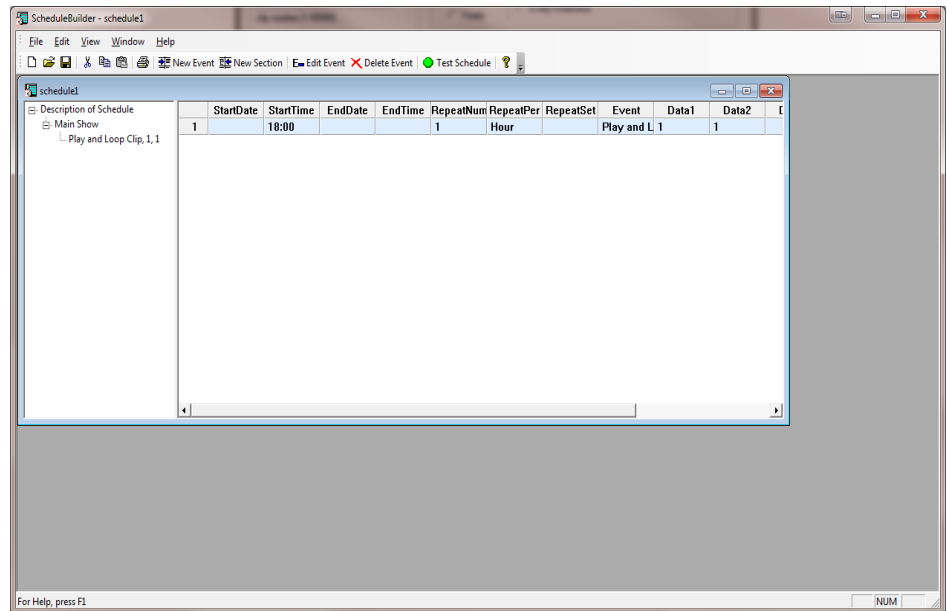
actual file name. If you are using the compact flash card that shipped inside your RideTraXX, select 'By Number' and enter the number '1' in the field.



5. Select the time and date you wish to execute the command in the Start Date/Time section on the right side of this screen.
6. If you only want the command to be executed once, just click OK. However, if you want the command to be repeated (say every hour or every day) you can specify that in the repeat box on the right side of the screen. For the sake of this example, we are going to repeat this command every hour.



7. Click OK and the event will be added to the schedule.



8. In this screen, you are free to Add/Edit/Delete Scheduled events as needed.
9. Once you are finished creating the schedule, all you need to do is **save** the schedule file as **schedule.scl** and copy it to the RideTraXX's Compact Flash card.

# Command Protocol

The Serial, Ethernet, Playlist, and Scheduler control interfaces use the Pioneer Alcorn McBride Enhanced protocol. This section discusses the most commonly used commands in detail.

The protocol is ASCII-based. Upper or lower case characters can be used interchangeably. **All commands and responses are terminated with the Carriage Return character. This character is represented by the hexadecimal value of 0x0D.**

## Playback Commands

These commands control the playback operation of the RideTraXX. These commands can be directed at an individual **Channel, Channel Pair, or Channel Group.**

Channel Group numbers range from 1 to 13 and are always preceded by the letter 'G'. For example,

**G5** means Channel Group #5.

Channel Pair numbers range from 1 to 8 and are always preceded by the letter 'P'. For example,

**P4** means Channel Pair #4.

Individual Channel numbers range from 1 to 16 and are always preceded by the letter 'C'. For example,

**C3** means Channel #3

## Play - PL

Description:	This command causes the RideTraXX to play a sound file.	
Command Bytes:	<b>&lt;n&gt;&lt;d&gt;&lt;i&gt;PL</b> or " <b>&lt;filename&gt;&lt;d&gt;&lt;i&gt;PL</b>	
Parameters:	<b>&lt;n&gt;</b> is the ASCII file number between 0-99999 <b>&lt;filename&gt;</b> is the ASCII file name <b>&lt;d&gt;</b> is either <b>C</b> followed by a channel number in ASCII, <b>P</b> followed by a channel pair number in ASCII, or <b>G</b> followed by a channel group number in ASCII <b>&lt;i&gt;</b> is either <b>U</b> for uninterruptible playback or <b>I</b> for interruptible playback. This parameter is optional.	
Message Response:	<b>R</b>	
Comments:	If no sound number or filename is specified, the file specified with the last Search command (see below) will be used. Channel numbers range from 1 through 16, Channel Pairs from 1 through 8, and Channel Groups from 1 through 13. Use "*" for All Channels. This command can also start the system playlist or can be used to force a playlist to continue by using the '>' character in place of a file number.	
Examples:	Play file 21 to Channel Pair 4.	<b>21P4PL</b>
	Play file 9 to Channel 3.	<b>9C3PL</b>
	Play file 1 to Channel Group 12.	<b>1G12PL</b>
	Play file "music.mp3" to all channels.	<b>"music.mp3"*PL</b>
	Move the playlist on Channel Pair 1 to the next entry	<b>&gt;P1PL</b>

## Loop Play - LP

Description:	This command causes the RideTraXX to play a sound file, automatically looping it back to the start when the sound finishes.	
Command Bytes:	<b>&lt;n&gt;&lt;d&gt;&lt;i&gt;LP</b> or " <b>&lt;filename&gt;&lt;d&gt;&lt;i&gt;LP</b>	
Parameters:	<b>&lt;n&gt;</b> is the ASCII file number between 0-99999 <b>&lt;filename&gt;</b> is the ASCII file name <b>&lt;d&gt;</b> is either <b>C</b> followed by a channel number in ASCII, <b>P</b> followed by a channel pair number in ASCII, or <b>G</b> followed by a channel group number in ASCII <b>&lt;i&gt;</b> is either <b>U</b> for uninterruptible playback or <b>I</b> for interruptible playback. This parameter is optional.	
Message Response:	<b>R</b>	
Comments:	If no sound number is specified, the number specified with the last Search command (see below) will be used. Channel numbers range from 1 through 16, Channel Pairs from 1 through 8, and Channel Groups from 1 through 13. Use "*" for All Channels.	
Examples:	Play and loop file 21 to Channel Pair 4.	<b>21P4LP</b>
	Play and loop file 9 to Channel 3.	<b>9C3LP</b>
	Play and loop file 1 to Channel Group 12.	<b>1G12LP</b>
	Play and loop file "music.mp3" to all channels	<b>"music.mp3"*LP</b>

## Sync Play - SP

Description:	This command causes the RideTraXX to start synchronous playback of searched files on the specified channel pair or channel group. If no channel pair or channel group is specified, it will start synchronous playback of all searched channels. Synchronous playback will begin 3 seconds after the <b>SP</b> command is received (+/- 10ms). During the 3 seconds between receipt of command and start of playback, all audio outputs will be silent.
Command Bytes:	<b>&lt;d&gt; SP</b>
Parameters:	<b>&lt;d&gt;</b> is either <b>C</b> followed by a channel number in ASCII, <b>P</b> followed by a channel pair number in ASCII, or <b>G</b> followed by a channel group number in ASCII
Message Response:	<b>R</b>
Comments:	Channel numbers range from 1 through 16, Channel Pairs from 1 through 8, and Channel Groups from 1 through 13. Use "*" for All Channels.
Examples:	Sync Play searched files on channel group 2. <b>G2SP</b> Sync Play all searched files. <b>SP</b>

## Sync Loop - SL

Description:	This command causes the RideTraXX to start looping synchronous playback of searched files on the specified channel pair or channel group. If no channel pair or channel group is specified, it will start looping synchronous playback of all searched channels. Synchronous playback will begin 3 seconds after the <b>SL</b> command is received (+/- 10ms). During the 3 seconds between receipt of command and start of playback, all audio outputs will be silent.
Command Bytes:	<b>&lt;d&gt; SL</b>
Parameters:	<b>&lt;d&gt;</b> is either <b>C</b> followed by a channel number in ASCII, <b>P</b> followed by a channel pair number in ASCII, or <b>G</b> followed by a channel group number in ASCII
Message Response:	<b>R</b>
Comments:	Channel numbers range from 1 through 16, Channel Pairs from 1 through 8, and Channel Groups from 1 through 13. Use "*" for All Channels.
Examples:	Sync Loop searched files on channel group 2. <b>G2SL</b> Sync Loop all searched files. <b>SL</b>



## Background Search - BS

Description:	This command is used to queue a file on a channel pair in preparation for a crossfade. Upon issuing a crossfade command, the file specified by this command will be crossfaded to. This command can only be issue while RideTraXX is in Crossfade Mode.	
Command Bytes:	<b>&lt;n&gt;&lt;d&gt; BS</b> or <b>"&lt;filename&gt;"&lt;d&gt;BS</b>	
Parameters:	<b>&lt;n&gt;</b> is the ASCII file number between 0-99999 <b>&lt;filename&gt;</b> is the ASCII file name <b>&lt;d&gt;</b> is either <b>P</b> followed by a channel pair number in ASCII, or <b>G</b> followed by a channel group number in ASCII	
Message Response:	<b>R</b>	
Comments:	Channel Pairs from 1 through 8, and Channel Groups from 1 through 13. Use "*" for All Channels.	
Examples:	Queue file 2 on channel pair 3.	<b>2P3BS</b>
	Queue "scene3.mp3" on channel pair 5.	<b>"scene3.mp3" P5BS</b>

## Play Toggle - PT

Description:	This command causes the RideTraXX to toggle playback of a file. If currently stopped, the file will play. If currently playing, the file will stop.	
Command Bytes:	<b>&lt;n&gt;&lt;d&gt;&lt;i&gt;PT</b> or <b>"&lt;filename&gt;"&lt;d&gt;&lt;i&gt;PT</b>	
Parameters:	<b>&lt;n&gt;</b> is the ASCII file number between 0-99999 <b>&lt;filename&gt;</b> is the ASCII file name <b>&lt;d&gt;</b> is either <b>C</b> followed by a channel number in ASCII, <b>P</b> followed by a channel pair number in ASCII, or <b>G</b> followed by a channel group number in ASCII <b>&lt;i&gt;</b> is either <b>U</b> for uninterruptible playback or <b>I</b> for interruptible playback. This parameter is optional.	
Message Response:	<b>R</b>	
Comments:	If no sound number or filename is specified, the file specified with the last Search command (see below) will be used. Channel numbers range from 1 through 16, Channel Pairs from 1 through 8, and Channel Groups from 1 through 13. Use "*" for All Channels.	
Examples:	Play toggle file 21 to Channel Pair 4.	<b>21P4PT</b>
	Play toggle file 9 to Channel 3.	<b>9C3PT</b>
	Play toggle file 1 to Channel Group 12.	<b>1G12PT</b>
	Play toggle file "music.mp3" to all channels.	<b>"music.mp3"*PT</b>

## Search - SE

Description:	This command causes the RideTraXX to preload a file to a specific Channel, Channel Pair, or Channel Group. If the next Play or Loop command is sent without a sound number or file specified, the preloaded sound number specified by this command is used. This is especially useful for All Channel play commands, where you can leave the sound number unspecified, and the channel number is specified as “*”. The RideTraXX will start playback of all channels at the same time, using the assigned files for each channel specified by a previous Search command.
Command Bytes:	<n><d><i>SE or "<filename>"<d><i>SE
Parameters:	<n> is the ASCII file number between 0-99999 <filename> is the ASCII file name <d> is either <b>C</b> followed by a channel number in ASCII, <b>P</b> followed by a channel pair number in ASCII, or <b>G</b> followed by a channel group number in ASCII <i> is either <b>U</b> for uninterruptible playback or <b>I</b> for interruptible playback. This parameter is optional.
Message Response:	<b>R</b>
Comments:	Channel numbers range from 1 through 16, Channel Pairs from 1 through 8, and Channel Groups from 1 through 13. Use “*” for All Channels.
Examples:	Search file 21 to Channel Pair 4. <b>21P4SE</b> Search file 1 to Channel Group 12. <b>1G12SE</b> Search file “music.mp3” to all channels <b>"music.mp3"*SE</b>

## Start Crossfade - XF

Description:	This command starts crossfade between a currently playing file and the queued file as preloaded by a <i>Background Search</i> command.
Command Bytes:	<l><d>XF
Parameters:	<l> is the desired length of the crossfade in milliseconds(0 to 5000000) <d> is either <b>C</b> followed by a channel number in ASCII, <b>P</b> followed by a channel pair number in ASCII, or <b>G</b> followed by a channel group number in ASCII
Message Response:	<b>R</b>
Comments:	Channel numbers range from 1 through 16, Channel Pairs from 1 through 8, and Channel Groups from 1 through 13. Use “*” for All Channels.
Examples:	Start a 5 second crossfade on all channels with queued files. <b>5000P*XF</b> Start a 3.5 second crossfade on channel pair 2. <b>3500P2XF</b>

## Play Directory - PD

Description:	This command causes the RideTraXX to play the files located in a specified directory by automatically generating a playlist with the contents of the directory and starting the playlist on the specified channel, channel pair, or channel group.
Command Bytes:	<b>"&lt;dir&gt;"&lt;d&gt;&lt;options&gt;PD</b>
Parameters:	<p><b>&lt;dir&gt;</b> is the directory path. Leave blank with <i>no quotes</i> for root directory. <b>&lt;d&gt;</b> is either <b>C</b> followed by a channel number in ASCII, <b>P</b> followed by a channel pair number in ASCII, or <b>G</b> followed by a channel group number in ASCII <b>&lt;options&gt;</b> is one or more of the following:</p> <ul style="list-style-type: none"><li><b>A</b> – sort ascending</li><li><b>D</b> – sort descending (overrides ascending)</li><li><b>C</b> – sort case-sensitive</li><li><b>#</b> – sort by number</li><li><b>?</b> – randomize each directory (overrides all sorting options)</li><li><b>E</b> – embed playlists contained in directory into generated playlist</li><li><b>R</b> – recurse into subdirectories</li><li><b>L</b> – loop the directory by refreshing playlist at end</li></ul>
Message Response:	<b>R</b>
Minimum OS Version:	TraXXOS 1.3.0
Comments:	Channel numbers range from 1 through 16, Channel Pairs from 1 through 8, and Channel Groups from 1 through 13. Use "*" for All Channels.
Examples:	<p>Play all the files in the "sounds" directory, sorted by number, on ch pair 5. <b>"sounds"P5#PD</b></p> <p>Play all the files in the "music" directory, sorted ascending, on ch. 2. <b>"music"C2APD</b></p> <p>Play all the files in all directories, sorted randomly, on all channels. <b>*?RPD</b></p>

## Pause - PA

Description: This command causes the RideTraXX to pause playback of the specified Channel, Channel Pair or Channel Group. Playback is resumed whenever a Play command is issued to the paused Channel Pair or Channel Group

Command Bytes: **<d>PA**

Parameters: **<d>** is either **C** followed by a channel number in ASCII, **P** followed by a channel pair number in ASCII, or **G** followed by a channel group number in ASCII

Comments: Channel numbers range from 1 through 16, Channel Pairs from 1 through 8, and Channel Groups from 1 through 13. Use “\*” for All Channels.

Message Response: **R**

Examples: Pause Channel Pair 2.                   **P2PA**  
Pause Channel Group 1.                   **G1PA**

## Stop - RJ

Description: This command causes the RideTraXX to stop playback of the specified Channel, Channel Pair or Channel Group.

Command Bytes: **<d>RJ**

Parameters: **<d>** is either **C** followed by a channel number in ASCII, **P** followed by a channel pair number in ASCII, or **G** followed by a channel group number in ASCII

Comments: Channel numbers range from 1 through 16, Channel Pairs from 1 through 8, and Channel Groups from 1 through 13. Use “\*” for All Channels.

Message Response: **R**

Examples: Stop Channel Pair 2.                   **P2RJ**  
Stop Channel Group 1.                   **G1RJ**

## Add Channels to Channel Group

Description:	This command will create a group from the specified channels. This allows you to control the grouped channels as one.	
Command Bytes:	<b>A&lt;n&gt;C&lt;#,#,#&gt;GR</b>	
Parameters:	<n> is the channel group number(1-16) <#,#,#> are the channels to be added to the group, separated by commas. (up to 16)	
Comments:	To clear the group, issue the command with no channels listed after <b>C</b>	
Message Response:	<b>R</b>	
Examples:	Add channels 2,3,6 to Group 3.	<b>A3C2,3,6GR</b>
	Clear group 6.	<b>A6CGR</b>

# Playlists

## Play System Playlist

Description:	This command causes the RideTraXX to start a playlist as the system playlist (not tied to any particular channel).
Command Bytes:	<n>PL or "<filename>"PL
Parameters:	<n> is the ASCII playlist number between 0-99999 <filename> is the ASCII playlist file name
Message Response:	<b>R</b>
Comments:	Playlists named ply00000.lst will automatically be started as the system playlist on startup.
Examples:	Start playlist 3 as the system playlist <b>3PL</b> Start playlist "cmds.lst" as the system playlist <b>"cmds.lst"PL</b>

## Play Channel Playlist

Description:	This command causes the RideTraXX to start a playlist on a specified channel pair
Command Bytes:	<n><d>PO or "<filename>"<d>PO
Parameters:	<n> is the ASCII playlist number between 0-99999 <filename> is the ASCII playlist file name <d> is <b>P</b> followed by a channel pair number in ASCII
Minimum OS Version:	TraXXOS 1.2.3
Message Response:	<b>R</b>
Examples:	Start playlist 3 as the channel 2 playlist <b>3C2PO</b> Start playlist "cmds.lst" as the channel pair 3 playlist <b>"cmds.lst"P3PO</b>

## Stop Channel Playlist

Description:	This command stops a channel playlist
Command Bytes:	<d>RJ
Parameters:	<d> is <b>P</b> followed by a channel pair number in ASCII
Message Response:	<b>R</b>
Examples:	Stop the channel 4 playlist <b>C4RJ</b>

# Control Commands

These commands control some of the operational modes of the RideTraXX.

## Version Request

Description:	This command is used to request the operating system version.	
Command Bytes:	<b>?V</b>	
Message Response:	<b>RideTraXX x.x.x</b> Where x.x.x is the current version of the installed firmware.	
Example:	Get current version.	<b>?V</b>

## Soft Reset

Description:	This command issues a software reset to the unit.	
Command Bytes:	<b>XX</b>	
Message Response:	<b>R</b>	
Example:	Reset Unit.	<b>XX</b>

## Set Name

Description:	This command gets/sets the user-readable name of the unit.	
Command Bytes:	<b>&lt;n&gt;NM</b>	
Parameters:	<b>&lt;n&gt;</b> is the desired ASCII name of the unit	
Message Response:	<b>R</b>	
Comments:	To get the current name of the unit, send <b>NM</b> without the <b>&lt;n&gt;</b> parameter.	
Minimum OS Version:	TraXXOS 1.2.5	
Examples:	Set unit name to "myTraXX".	<b>myTraXXNM</b>
	Get the current unit name.	<b>NM</b>

## Mute/UnMute/Volume Control

Description:	This command causes the RideTraXX to mute, unmute, or set the volume level of the specified Channel, Channel Pair, or Channel Group.	
Command Bytes:	<b>&lt;v&gt;&lt;d&gt;AD</b>	
Parameters:	<b>&lt;v&gt;</b> is one of the following (in ASCII): <b>M</b> - Mute <b>U</b> - Unmute <b>T</b> - Toggle Mute <b>#</b> - number between 0(silent) and 10(loudest) <b>&lt;d&gt;</b> is either <b>C</b> followed by a channel number in ASCII, <b>P</b> followed by a channel pair number in ASCII, or <b>G</b> followed by a channel group number in ASCII	
Comments:	Channel numbers range from 1 through 16, Channel Pairs from 1 through 8, and Channel Groups from 1 through 13. Use "*" for All Channels.	
Message Response:	<b>R</b>	
Examples:	Mute Channel 2.	<b>MC2AD</b>
	Unmute Channel Group 1.	<b>UG1AD</b>
	Set Volume of Channel Pair 3 to 7/10	<b>7P3AD</b>



## Send String

Description:	This command sends an ASCII string from RideTraXX to the specified destination.	
Command Bytes:	<b>"&lt;s&gt;"&lt;d&gt;&lt;o&gt;SS</b>	
Parameters:	<s> is the string to be sent <d> is the destination, one of: <b>S – Serial</b> <b>U – UDP(over Ethernet)</b> <b>B – USB Serial</b> <o> is only required for UDP, and is the destination info in the form of: <b>&lt;XXX.XXX.XXX.XXX&gt;:&lt;PPPPP&gt;:&lt;SSSS&gt;</b> where, <b>&lt;XXX.XXX.XXX.XXX&gt;</b> is destination IP <b>&lt;PPPPP&gt;</b> is destination port <b>&lt;SSSS&gt;</b> is source port	
Comments:	The <o> field is not necessary if using Serial or USB Serial destination. You must escape all special characters in the string to be sent(such as quotation marks). For example, "State=\"Playing\"\\r" would send: State="Playing"\\r	
Message Response:	<b>R</b>	
Examples:	Send 'Done' out the serial port: <b>"Done"SSS</b> Send 'Done Playing' to 192.168.0.254 port 2639 from port 2640: <b>"Done Playing"U192.168.0.254:2639:2640SS</b> Send 'State="Playing"\\r' to 192.168.0.254 port 2639 from port 2640: <b>"State=\"Playing\"\\r"U192.168.0.254:2639:2640SS</b>	

## Enable/Disable Crossfade Mode

Description:	This command enabled or disables Crossfade Mode	
Command Bytes:	<b>E&lt;s&gt;XF</b>	
Parameters:	<s> is either 1 (to enable) or 0 (to disable)	
Comments:	Crossfade Mode is saved and applied on bootup. This command only needs to be issued once, and the setting will remain after a power cycle.	
Message Response:	<b>R</b>	
Examples:	Enable Crossfade Mode	<b>E1XF</b>
	Disable Crossfade Mode	<b>E0XF</b>

## Status Request

Description:	This command requests the status of the specified channel, channel pair, or channel group.	
Command Bytes:	<b>&lt;d&gt;?P</b>	
Parameters:	<b>&lt;d&gt;</b> is either <b>C</b> followed by a channel number in ASCII, <b>P</b> followed by a channel pair number in ASCII, or <b>G</b> followed by a channel group number in ASCII	
Comments:	If a channel group or channel pair status is requested, a status for each individual channel will be returned, separated by line breaks. As with all responses, the status request is terminated with a final line break.	
Message Response:	<b>&lt;s&gt;</b> or <b>&lt;c&gt; &lt;s&gt;</b> <b>&lt;c&gt;</b> is the channel number <b>&lt;s&gt;</b> is one of: <b>P</b> - Playing <b>S</b> - Stopped <b>A</b> - Paused	
Examples:	Get Channel 3 Status	<b>C3?P</b> <i>Example Response: S</i>
	Get Channel Group 2 Status	<b>G2?P</b> <i>Example Response: 2 P&lt;h0d&gt;3 P&lt;h0d&gt;4 P&lt;h0d&gt;</i>

## Filename Status Request

Description:	This command requests the filenames that are currently playing on the specified channel, channel pair, or channel group.	
Command Bytes:	<b>&lt;d&gt;?C</b>	
Parameters:	<b>&lt;d&gt;</b> is either <b>C</b> followed by a channel number in ASCII, <b>P</b> followed by a channel pair number in ASCII, or <b>G</b> followed by a channel group number in ASCII	
Comments:	If a channel group or channel pair status is requested, a status for each individual channel will be returned, separated by line breaks. As with all responses, the status request is terminated with a final line break.	
Minimum OS Version:	TraXXOS 1.3.0	
Message Response:	<b>&lt;f&gt;</b> or <b>&lt;c&gt; &lt;f&gt;</b> <b>&lt;c&gt;</b> is the channel number <b>&lt;f&gt;</b> is the filename	
Examples:	Get Channel Pair 3 Status	<b>P3?C</b> <i>Example Response: 5 music1.mp3&lt;h0d&gt;6 music2.mp3&lt;h0d&gt;</i>
	Get Channel Group 2 Status	<b>G2?C</b> <i>Example Response: 2 yell.mp3&lt;h0d&gt;3 sing.mp3&lt;h0d&gt;4 hi.mp3&lt;h0d&gt;</i>
	Get Channel 12 Status	<b>C12?C</b> <i>Example Response: 12 soundfx.mp3</i>

## Completion Acknowledge Mode

**Description:** This command is used to select the Completion Acknowledge mode. When the RideTraXX finishes playing a file, it can send a Completion Acknowledge response or trigger a command.

There are 2 different completion acknowledge modes possible. This *Completion Acknowledge Mode* remains as long as the RideTraXX is powered. **Note:** The unit defaults to no completion acknowledge whenever the unit first powers up.

**Message Response:** **R**

**Command Bytes:** `<s>"<c>"<d>AK` Selects advanced channel response mode. This allows you to specify a command to be executed when the status of a channel pair or group changes. `<s>` is the channel status to trigger on  
**P** for play started  
**S** for stopped  
**L** for looped  
**A** for paused  
`<c>` is the command to perform when this status occurs (such as `\”CH1 Stopped\”SS`)  
`<d>` is either **C** followed by a channel number in ASCII, **P** followed by a channel pair number in ASCII, or **G** followed by a channel group number in ASCII

**RAK** Selects simple channel response mode.

**CH** is sent out the serial port upon a channel pair stopping, where CH = ASCII representation of the Channel pair number (0 through 7). For a completion ack via Ethernet, use the above advanced mode with a Send String command.

**Examples:** Select simple command response mode:

**RAK**

Use advanced channel response mode to send “ChPair1 Started” over UDP to 192.168.1.100 port 2639 from port 2640 whenever channel pair 1 starts:  
`P”\”ChPair1Started\”U192.168.5.150:2639:2640SS”P1AK`

# How To Update Firmware

Under certain circumstances, it may be necessary to update the RideTraXX's firmware. Do not update the firmware unless there is a specific need to do so.

## Standard Firmware Update

1. Download the latest firmware for the RideTraXX product from <http://www.alcorn.com/support/firmware.html>
2. Copy the new firmware file to your Compact Flash card. The file must be named "OS.NEW" if updating only one unit or "FACTORY.NEW" if updating multiple units using the same CF card. You may copy it directly using a CF card reader or over a network via FTP.
3. Reset this product by either power-cycling or sending the reset command.
4. When the CF card is detected after loading, the new firmware file will be located and the update process will begin. **Do not remove power from the unit during the update.**
5. When file access LED remains off, the firmware update is complete. If your firmware file was named "OS.NEW" it will be renamed to "OS.SAV" and the unit will automatically restart.
6. If the firmware update fails (i.e. power was interrupted during update), please follow the directions under "Firmware Recovery".

## Firmware Recovery

1. Remove power from the unit.
2. Set DIP Switch 3 to the ON position.
3. Download the latest firmware for this product from <http://www.alcorn.com/support/firmware.html>
4. Copy the new firmware file to your Compact Flash card. The file must be named "OS.NEW" if updating only one unit or "FACTORY.NEW" if updating multiple units using the same CF card. You must copy the file directly to the CF card using a CF card reader.
5. Insert the CF card, with the firmware update file, into the unit.
6. Apply power to the unit.
7. The new firmware file will be located and the update process will begin.
8. When the file access LED remains off, the firmware update is complete.
9. Remove power from the unit.
10. Set DIP Switch 3 to the OFF position.
11. Apply power to the unit.

# Troubleshooting Guide

**Q:** Triggering using the discrete inputs, the unit does not execute the assigned command.

**A:** Verify that the Voltage/Contact Closure DIP switch is properly configured.

**Q:** I've copied several files from my PC onto a Compact Flash, but they do not play.

**A:** The Compact Flash may have been pre-formatted in a way that is incompatible with the RideTraXX operating system. Be sure you have saved your files, then format the card as Microsoft FAT32.

**Q:** Audio playback is too fast or too slow.

**A:** All files played on the RideTraXX must be encoded to use the same sample rate (i.e all 44.1KHz or all 48KHz, etc.). You cannot mix and match sample rates.

**Q:** I hear snaps and pops in the playback audio, especially when playing all of the channels.

**A:** RideTraXX supports up to 10 mono or 8 stereo files at 192kbps and up to 16 mono at 128kbps. Make sure that your files do not exceed this bitrate. If playing more than 8 files at once, please verify that they are mono files. RideTraXX supports a simultaneous maximum of 8 stereo mp3s, or 16 mono.

For the answers to other questions, check out the Alcorn McBride Knowledge Base at <http://www.alcorn.com/kb>

If you don't find it there, please email our technical support staff at [support@alcorn.com](mailto:support@alcorn.com)

# Specifications

<b>Audio</b>	Tracks: 16 Mono or 8 Stereo, Synchronous Format: MPEG-1 Part 3 (or simply, MP3) defined in ISO/IEC-11172-3 Maximum Bitrate: 192kbps for 8 simultaneous stereo, 128kbps for 16 simultaneous mono Sample Rate: 44.1Khz or 48Khz DAC Resolution: 16 Bits Frequency Response: 10 to 20 Khz +/-0.1 dB (dependent on MP3 encoding) Line Output Level: Unbalanced, 1.15Vrms into 20K Line Output Load: 10K Ohm Minimum, 100pF Maximum
<b>Playback Time:</b>	Approximately 1 MByte per minute per channel with excellent audio quality at 128kbps
<b>Controls:</b>	Discrete: 8 programmable and individually configurable contact or voltage (12-24vdc) inputs Serial: DB9M RS-232 9600,N,8,1 Ethernet (10/100Base-T, UDP control, FTP file transfer, NTP time protocol) USB Serial
<b>Connectors:</b>	DB-9M serial RS-232 connector USB type-B connector for USB serial RJ45 jack for Ethernet connection 20-Pin Molex Type connector for discrete inputs and Power 24-Pin Molex Type connector for unbalanced audio output
<b>Front Panel:</b>	Power LED & Status LEDs. Compact Flash Socket
<b>Power:</b>	9 VDC to 36 VDC at 2 Amp max
<b>Physical:</b>	Size: 7" W x 1.719" H x 10.88" D (17.78 cm W x 4.44 cm H x 27.64 cm D) Weight: 4lbs. (1.8Kg) Environment: 0°C (32°F) to 38°C (100°F) 0-90% Relative Humidity
<b>Compliance:</b>	CE, UL, CSA, WEEE, RoHS

# Index

- AMI Terminal, 19
- Bitrates, 13
- Channel Encoding, 13
- Command Protocol, 32
- Communication LED, 5
- Compact Flash, 5, 13
- Completion Acknowledge Mode, 45
- Configuration DIP Switches, 11
- Control Commands, 41
- Control Methods, 19
- Defaults, 12
- DIP Switch
- Ground Lift, 11
- Discrete Input Control, 20
- Ethernet Control, 19
- File Access LED, 5
- file naming, 14
- Firmware, 46
- front panel, 5
- FTP, 12, 18
- Gateway, 11, 12, 18
- Ground Lift, 11
- interruptible, 15
- IP Address, 12, 18
- ISO/IEC-11172-3, 13
- Loop Play, 33
- Manage Channel Groups, 23
- Manage Input Groups, 21
- MPEG-1 Part 3, 13
- Mute, 42
- Noise, 11
- On Startup Trigger Type, 24
- Password, 12, 18
- Pause, 38
- Play, 33, 34, 35, 37, 40
- Playback Commands, 32
- Playlist Builder, 28
- Playlists, 25
- rear panel, 7
- RJ45, 7
- RS-232C connector, 10
- Sample Rates, 13
- SanDisk®, 13
- Schedule Builder, 29
- Search, 36
- Serial Control, 19
- Soft Reset, 41
- Specifications, 48, 49
- Status Request, 44
- Stop, 38
- Subnet Mask, 10, 11, 12, 18
- technical support, 2
- test sounds, 4
- TraXX Configuration Software, 21
- Troubleshooting, 47
- uninterruptible, 15
- UnMute, 42
- Update Firmware, 46
- Version Request, 41