

RS-232 to DMX-512 Converter
Model 4200
Firmware Revision 1.2
PC Board Revision 1.0
Instruction Manual

Durand Interstellar, Inc.
219 Oak Wood Way
Los Gatos, California 95032-2523
www.interstellar.com
tel: +1 408 356-3886, fax: +1 408 356-4659

Intended Use

This product will allow computers of almost any make and speed to control lighting in situations where a full lighting board is impractical. While this product does support all 512 channels per USITT DMX512/1990 (8 μ S MARK AFTER BREAK), the number of level changes per second is limited by the speed of the RS-232 port and the number of channels in use. A slow computer (old '286 laptop) running MS-DOS is able to perform smooth fades on 24 channels, so you're not as limited as you might think. For most intended applications (trade shows, hotel events, store windows, museums, amusement park displays, etc.) this limitation will never show up. You may store one static level for each of the 512 possible lighting channels in each of the 64 internal presets (current active and 63 non-volatile). The non-volatile presets (NV-RAM) will be held until you change them, even with the power off. You may choose to have one of the NV-RAM presets sent out to the lighting dimmers automatically on power-up. You may even connect this converter to an external modem (set to auto-answer) and control it over the phone. Wireless modems work well, also.

Please note that the Windows demo program is intended as an example and does not have complete error checking. It also has not been tested on every combination of computer and version of Windows and so it may crash unexpectedly.

Since the DMX protocol itself has no error checking, you should never use it to control anything that might be damaged or cause injury if the wrong data is received.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

As defined by FCC rules (§15.3h), this is a "Class A digital device". A digital device that is marketed for use in a commercial, industrial or business environment, exclusive of a device which is marketed for use by the general public or is intended to be used in the home.

There are no user serviceable parts inside, remove the cover only to change jumper settings. Be sure to re-attach the cover securely and never apply power with the cover removed.

The latest version of this manual and firmware are available from our support site:

<http://www.interstellar.com/support.html>

All of our documentation is in Adobe Acrobat format. You can download a free reader for PCs and Macs at:

<http://www.adobe.com>

All trademarks are the property of their owners and are used for reference only.

Basic Hardware Setup

First, you will need a computer or PLC that has an available serial port (COM Port) that supports RS-232. You will also need a cable to connect your computer to the converter. If you have a PC, a simple 9-pin “straight-through” male to female cable should work. If you have a Mac or other computer, see your local computer dealer for any adaptors or special cables that you might need (the computer and cable MUST support RTS/CTS handshaking). Connect the serial cable to the 9-pin female connector on the end of the converter.

The converters are shipped set for 9600 baud, if you wish to use a different rate, see the Advanced Hardware section of this manual.

Now, connect the output of the converter (5 pin XLR connector on end of converter) to your dimmers (they must support the DMX-512 protocol, see your dimmer instructions for connection and setup information). If possible, set your dimmers to start on the first channel (1 or 0, depending on the dimmer). In any case, note the channel numbers of the dimmers you wish to control and whether they consider 1 or 0 the first channel.

Attach some lights to the dimmers and disconnect anything that may be damaged or cause injury if the dimmer receives bad control information.

If not already on, turn on the power to the dimmers, the computer, and plug in the converter using the supplied adaptor (use of improper adaptor or modifications to the supplied adaptor will void the warranty and may cause damage and/or injury, see the Power Notes below). The red POWER LED on the converter should come on.

If the converter was set to auto-playback mode (factory default), the green OUT LED will start blinking (it may appear to be constantly on) and your lights will be set to whatever levels were stored in the NV-RAM (the levels will be random on a new converter). If auto-playback was not selected, no DMX data will be sent and the lights attached to the dimmers should remain off.

The factory default for the number of channels is 64 (40 hex), but you may change this using the “N” command.

Demo Software

We are currently shipping the demo program from our old converter. It works fine with the new models. You can check here for the latest software:

<http://www.interstellar.com/support.html>

Note: The old Windows demo program only controls lights on the first four channels. If you do not have any dimmers or lights on these channels you will not be able to use the demo program as supplied. This program is intended to demonstrate one way of using this converter, it is not intended to be used in actual applications.

The converter is shipped with the baud rate set to 9600, if you have changed it, you must return it to 9600 to use the demo program.

Run the “demo.exe” program found on the enclosed disk. If the converter is connected to COM:1, you do not need to change the port settings. If it is connected to another port, click on the ComPort button and select the correct port (see your computer manual if you are unsure of which port you are using). The other settings should always be “9600”, “8”, “1”, “none”, and “RTS/CTS”. Select OK to return to the main window.

If the green OUT light is on or blinking, click on the button labeled “Playback Mode 0”, this will instruct the converter to stop sending data from the active memory (Preset 0). You may see the yellow IN light flash momentarily and then the green light should stay out. A short time later, your dimmers should turn off any attached lights (and possibly indicate a DMX no-data error). Now, press the button labeled “Slider Data in Loop” and the yellow IN light should be on (you may notice a flicker, this is fine), this means the computer is sending commands to the converter. The green OUT light should also be on or flickering, this indicates that data is being sent to the dimmers.

The 4 sliders are now controlling dimmer channels 1-4 (or 0-3 in some dimmers). Use your mouse to drag the sliders to new positions and note that you are controlling the lights. Also, any error indication on your dimmers should now be off. If you have changed the number of channels from the default, you may notice a delay between changing a level in the demo program and the lights actually changing. This is a result of the “D” command being sent out without waiting for the prompt. See the “D” command for more information. If you set “N” to a small number (less than about 170, 4 would be good since that’s all the channels the demo supports anyway) you will not see any delay.

Select the button labeled “None” to stop sending data to the dimmers. The yellow and green LEDs will go out and your dimmers will detect the loss of data and should shut off all attached lights.

Click on the button labeled “Playback Mode 1”, the yellow LED will blink once and the

green LED will light. The converter is now sending 64 (or whatever “N” is set to) channels of levels from NV-RAM Preset 1. Your dimmers should clear any error indication and set all attached lights to whatever levels were stored in the NV-RAM.

Set the sliders to four different positions and then click the button labeled “Slider Data To Memory”. The yellow LED will blink once and the level of the sliders will be stored in the first four channels of the active memory (channels 5-512 will not be changed by this demo program). The new levels will be sent out to the dimmers immediately (no delay as there was with the loop mode, this mode is using “W”). Try setting the sliders to various positions and clicking the “Slider Data To Memory” button to see how you can change levels in the active memory. These levels will be maintained until changed, but will be lost when the power is turned off. The demo program does not write to the NV-RAM presets.

If you click on “Playback Mode 1” again, the levels from NV-RAM Preset 1 will be copied back to the active memory, overwriting any changes that you made.

You may leave the program by pressing the “Exit” button.

Advanced Hardware

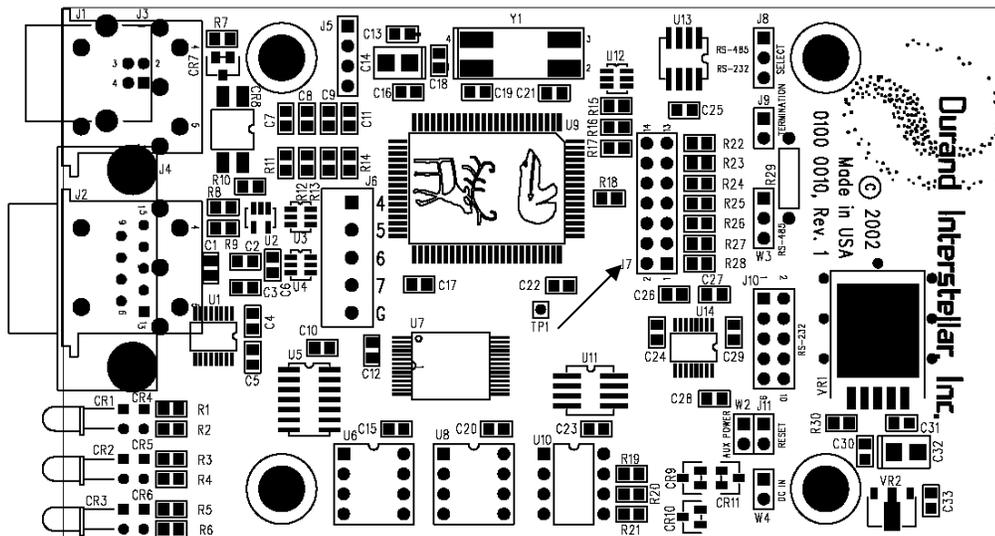
This section only applies to PC Board Revision 1. If you have Revision 0 (shipped before January 2003) please contact us for the old instruction manual.

There are jumpers. To access these jumpers, unplug the power cable and remove the cover from the converter. You need to remove the 4 screws on the bottom of the converter and then the cover will lift off. Inside you will see a PC board with dual-pin post type jumpers (see below). If you remove a shunt you should replace it so that it only covers one post. This way you won't lose it (spares available from any electronics store).

J7 is the large rectangular jumper with 14 pins to the right of the center of the board. The first 4 pins are for baud rate selection, the rest are not used in this model. In the picture below, pin one is to the bottom right of J7, pin 2 is the bottom left, pin 3 is second from the bottom on the right, and pin 4 is second from the bottom on the left.

Baud rate selection

- 9600 baud: shunt between pins 1 & 2, shunt between pins 3 & 4 (default).
- 19200 baud: shunt between pins 3 & 4 only.
- 28800 baud: shunt between pins 1 & 2 only.
- 38400 baud: no shunts installed.



Anyone working with DMX systems should have the booklet “Recommended Practice for DMX512” available from PLASA Ltd. in the UK and USITT in the USA.

If you'd like to know more about how the DMX signal is actually sent to dimmers, you will want to read the actual specification from USITT titled “DMX512/1990”.

Advanced Software

If the demo program doesn't do everything you need, you will need to either write your own control software or purchase a package from us or a third party. Of course, for just setting static lighting levels, you can also use any terminal emulator program and type in the values by hand (slow, but you only have to do it once).

If you'd like to see how the Windows demo program works, the source files are available at:

http://www.interstellar.com/support/Model_0001/windows/

If you haven't yet, read the booklet "Recommended Practice for DMX512" published by PLASA and USITT.

The serial port on the host computer **MUST** be set up as:

- The baud rate must match the speed selected by the jumpers in the converter.
- 8 data bits
- 1 stop bit
- no parity
- RTS/CTS handshaking

If you aren't comfortable with hexadecimal numbers, see pages 72-74 of the above book or any introductory computer programming text. All numbers sent to and returned from the DMX converter are in hex.

Leading zeros are not required, for the number zero you must send at least one "0". Numbers returned from the converter may or may not have leading zeros. If a number is received by the converter with more digits than it is looking for, only the last digits are kept ("12345678" is read as "5678"). Do NOT send the "0x" prefix in front of any number.

All commands and hex numbers may be in upper case, lower case, or a mixture of both.

White space is defined as Horizontal Tab (09 hex), Line Feed (0A hex), Vertical Tab (0B hex), Form Feed (0C hex), and Space (20 hex). Any amount/combination of white space may precede a command or number. White space may also follow the last data on a line and precede the carriage return.

All commands execute as they are received but they **MUST** be terminated with a carriage return (0D hex)...that's '\r' for you C programmers, NOT '\n'. This makes it much easier to use a terminal emulator. If you want to send "\r\n" or "\n\r", that's OK.

Any of these commands (except the "D" command) may be typed in from a terminal emulator program (such as Hyperterminal in Windows). There are no timeouts on

commands (except “D”), you could start a command today and finish it tomorrow and the converter wouldn’t care (unless there was a power failure). Aside from debugging, the main use of manual command entry would be for static (non-changing) lighting setups where you set the levels once, enable auto-playback, unplug the computer, and go home.

If your computer allows you to copy text files to the serial port, you could make up a text file containing anything from a single cue to a long series of commands creating a fade. Assuming your system always copied the files at the same speed, you could create fairly complicated lighting effects with nothing more than a text editor (like Notepad in Windows).

If your terminal emulator can execute scripts, you could program an entire show with cross fades in the scripting language. A sample Procomm script is available at:

http://www.interstellar.com/support/Model_0001/DMX.asp

This script uses the “D” command, “W” would work just as well with the new models and is preferred in most cases since you only need to send changes.

Command Descriptions

There are only 10 commands to master, and you really only need a couple in most applications.

Don't forget the carriage return (0D hex) after every line!

If an error is detected while processing a command, a '!' character is sent to the host. When the converter is ready for a command, it will send a prompt to the host. The prompt consists of a return (0D) and linefeed/newline (0A) followed by the character '>'. In most applications you can safely ignore error checking (you shouldn't ever get any), however you **MUST** set up your software to deal with incoming characters. The demo program discards them, if you turn off that section everything will be fine for about 10 seconds and then the receive buffer will fill up. When this happens Windows sets RTS to FALSE (the host saying "Stop sending characters"). The converter will then be waiting for RTS to go back to TRUE before it sends out the next prompt. Since the converter can't execute the next command, its receive buffer will fill up and it will set CTS to FALSE telling the host to stop sending characters. Both computers will now wait a very long time for something to happen.

If you're sure you never want to receive the return prompt or error messages, you could cut the wire in your serial cable (pin 2 on the DMX converter end). This way you will never receive any characters that need to be discarded.

There is no harm in issuing the next command before the current one has completed. The new command(s) will be buffered and executed as soon as the previous one has completed.

Commands:

I INFORMATION COMMAND

When the converter receives this command it will respond with several lines of text describing the software and what the initial conditions are. Here's an example of what the converter will send:

```
RS-232 to DMX Converter
Copyright 1999-2001, Durand Interstellar, Inc.
Version: 4200 1.0 20011225
Serial Number: 00000000
Playback mode: 01
Start Code: 00
Channels: 040
```

You may wish to set up your software to check that you're talking to the right converter (our other products will also have an 'I' command). The first number after "Version:" is the model number ("4200"), the next number is the firmware version ("1.0") and the last number is the firmware date ("25

December 2001”). The playback mode reflects the current setting (00 = off, 01-3F = which NV-RAM preset was last copied to active memory). Other versions of this product may have higher playback mode numbers (for example, Model 4201 uses 00-7F) and more lines of information.

P SET PLAYBACK MODE

Valid modes are 00 for off and 01-3F for auto-playback. As soon as this command is entered, the DMX output is stopped. If the new mode is not equal to zero, the selected preset is copied from NV-RAM to the active memory (Preset 0) and the DMX output is re-started. This all happens VERY quickly.

Samples:

P0

P4

R READ MEMORY

This command will cause 16 bytes to be read back to the host from memory. The memory is accessed as a single 32K block, so you supply the starting address as a 16 bit hex number where the upper 7 bits are the preset number. Addresses 0000-01FF are channels 1-512 in the active memory (these are what are sent out the DMX port). The rest of the memory block is the NV-RAM. This command always sets the lower 4 bits of the address low before starting the read, so the last digit of the address you send doesn't matter (but is required). So if you send 123, the first byte returned will be from Preset 0, channel 120 (hex). The first line below shows a read back command for channels 120 through 12F, the second line shows a typical response:

R123

11 22 33 44 55 66 77 88 99 00 AA BB CC DD EE FF

Each channel is returned as a 2-digit hex number with a space between channels.

To read back the first 16 channels of NV-RAM Preset 1, you would send “R200”. To read back NV-RAM Preset 2 you would send “R400”. The very last channel in the last NV-RAM Preset is at address “7FFF”.

D DIRECT TO DMX COMMAND (not recommended for new applications, this command is to maintain compatibility with older versions)

This is the only command that you shouldn't type in at a terminal emulator. The DMX specification has a 1-second timeout for loss of data detection, so unless you type VERY fast, your dimmers will time out and shut down.

The simple version of this command is just a D followed by a string of hex numbers separated by white space. The first number is channel 1, the next is 2, and so on. The converter will hold the data as it's received until you send a return, then it will all be sent out as one packet. If you send more than the number of channels set with "N", the extra channels will be ignored.

Sample:

```
D 1 2 3 4 5 6 7 8 9 a b c d e f 10 11 12 13
```

Since the DMX format doesn't let you skip channels before you get to the one you're interested in, this can get tedious (and slow) if you have a device on channel 500. To use the simple format you'd have to insert 499 zeros plus white space. Enter the Skip subcommand (a form of RLL coding). DMX data is only 8 bits, so each value on the line must be between 00 and FF. If a number is between F00 and FFF it is interpreted as a skip command and will insert a number of zeros in the DMX data stream at high speed relieving both the host computer and the programmer of the task. The number of channels "skipped" is determined by the lower 8 bits (the right two digits) of the 3-digit number. F01 means "insert one channel of 0", F02 means insert 2 zeros, etc. with F00 a special case that inserts 256 zeros. So to send the data 55 to channel 500, you could enter:

```
D F00 FF3 55
```

Note that we had to use two skip commands in a row, as long as they're separated by white space, this works fine. Simply repeat the skip command as many times as you need.

You can also intersperse skip commands with data as in:

```
D F12 55 F22 AA
```

Please note that skipped channels are set to a level of 0. If you have any lights on those channels, they'll be dark. If you have 500 channels of lights and want to change any one of them, you MUST send out all 500 channels of data every time. If this is too much trouble for you, then see the "W" command.

With the above commands the host has to send a minimum of 2 characters and normally 3 characters per channel (white space + one or two digits). This isn't slow but it isn't fast, either. For those of you who want the fastest updates possible, there's the binary subcommand. It works something like the skip command, only the first digit is 'B' and the lower 8 bits are the number of binary channels to follow ("B12 " means the next 12 hex bytes that come from the host are 8-bit binary levels). Since binary data doesn't print well, we'll use 'x' in the example to represent the 8-bit binary data.

```
D B04 xxxx F05 AA 55
```

In the above example the "B04 " means that the next 4 characters, whatever they are, are treated as a level with no checks performed. Note there MUST be a single space (or other white space) after the "B04" or the command won't work. The rest of the command string just shows that you can mix these subcommands with normal simple data. BE CAREFUL, if one of the 'x's was missing, the white space in front of the "F05" would be sent as a level and then the "F05" would generate an error. So, if you had 8 channels in binary mode starting at channel 1, the command would look like:

```
D B08 xxxxxxxx
```

A note on the "D" command. As soon as the "D" is received, the Playback Mode is switched to 0. As each channel is received it is stored in active memory. When the Carriage Return at the end of the line is received, one (and only one) DMX packet is sent out and then the DMX output is halted again. The last data sent out will remain in active memory until changed by the "D" or some other command (including a "P" command). You can use the Copy command to place this data in some other Preset, if you wish.

Also, due to buffering delays in the host computer and the converter, if you are sending out "D" commands rapidly and "N" is set to a large number, you will notice a delay between issuing a "D" command with new levels and the lights actually changing. There are several ways around this, send the "D" commands slower than the packet rate (depends on the setting of "N"), wait for the prompt before sending the next "D" command, use a smaller value of "N", or use the "W" command.

W WRITE TO MEMORY

This command allows you to update locations in the memory (active or NV-RAM). You supply a 16 bit starting address followed by 1 or more bytes of data to be written. To write the data (1, 2, 3, 4) to channels 5-8 of active memory, you would use this command:

```
W 4 1 2 3 4
```

Note that the memory address is always 1 less than the channel number

(assuming channel numbers start at 1). If you are writing to addresses in the active memory and the playback mode is not 0, every change you make will be sent out to the DMX port immediately. This is the preferred way to dynamically control lights from your software; you only need to send the changes and they take effect immediately. If you are writing to NV-RAM (addresses 0200-7FFF), each channel is stored as you enter it but it does not effect the current DMX output.

Be sure to see the note below on the NV-RAM.

This command will allow you to write past the end of one preset and into the next. This could be handy for writing several presets all at once. You will only get an error when you attempt to write to preset 64 (address 8000).

C COPY MEMORY

This will copy any preset to any other. This command ALWAYS copies all 512 channels, no matter what “N” is set to. Simply enter the source preset number and the destination preset number. If the destination is in NV-RAM, this command will be somewhat slow as writing to this memory is not fast.

Be sure to see the note below on the NV-RAM.

The active memory is considered to be Preset 0. If the Playback Mode is not 0 and you copy from NV-RAM to Preset 0, the new levels will be sent out as they are copied (possibly over several DMX packets). This could be a problem if you are controlling moving lights or other items. It is recommended to use the Playback Mode to copy to active memory as it is optimized for speed and is synchronized with the outgoing DMX packets. Using the copy command does NOT interfere with the DMX output as long as you are writing to NV-RAM.

To copy from active memory to NV-RAM Preset 16 (10 hex), enter:

C 0 10

To copy from Preset 1 to Preset 63, enter:

C 1 3F

S START CODE

This command lets you change the DMX Start Code to some other number. The only valid number defined by the DMX Standard is “00”, but other devices may require a different start code. There’s also no reason you can’t change the start code between “D” commands. This way you could have some non-standard item on the same DMX cable as well as normal lighting dimmers. Your computer would have to keep track of what it was sending.

Example:

| | |
|-------------|--|
| S0 | (control regular lighting channels) |
| D 1 2 3 4 5 | (send lighting data) |
| S55 | (special start code for non-standard item) |
| D A B C D | (send commands to non-standard item) |
| S0 | (back to beginning of loop) |

N NUMBER OF CHANNELS

This command lets you set the maximum number of channels. If you are using a small number of channels, this GREATLY increases the number of packets sent out per second. Note that if you change Playback Mode it will ONLY copy the number of channels set by this command. The higher channel numbers in active memory will not be changed.

If you set N to 0, DMX packets of zero length will be sent. Be careful, as this may confuse some devices. If your device doesn't mind this, you could use this as a way to pause the DMX transmission while you update active memory with the "W" command.

The maximum number of channels is 512 (200 hex), factory default is 64 (40 hex).

Z WRITE STARTUP DATA

This command will copy the Playback Mode, Start Code, and Number of Channels to NV-RAM. These settings will be recalled the next time power is cycled. So if you don't want anything to transmit on power up, set Playback Mode to 0 and then issue the "Z" command. Note that you only have to set this up once and it will work every time power is cycled.

Be sure to see the note below on the NV-RAM.

U UPDATE FIRMWARE

This command is ONLY to be used when there is a firmware update available for your model number (firmware updates will NOT work on different models). If you have custom firmware, then the updates will only work for one specific serial number. Use the "I" command to check your model and serial numbers.

Assuming you have a new update that we sent you or you downloaded from our web site, then the first thing you should do is a test run. This will check the update file for errors, but won't actually change the existing firmware.

To do this with a terminal emulator, you should make sure you get a “>” prompt every time you press return. Then, type “UT” but do NOT press return. Open your Send File menu and select ASCII TEXT. Now send the update file. This will take less than a minute at 38400 baud. If everything worked, you’ll receive a message telling you that there were no errors and to re-run it with “UW” instead of “UT”. This will write the new firmware to memory.

DO NOT INTERRUPT THIS PROCESS!

If you get an error message (which you shouldn’t since you ran the “UT” command with no problems), then you may try again (you are still running on the old firmware until you cycle the power). If you can’t get the update to complete properly, you now have a dead unit that must be returned to the factory for repair.

When the update completes normally (it takes more than a minute), you will receive a message that everything went ok. At this point the processor halts and you should cycle the power to load the new firmware. Issue an “I” command and check that the version has changed to the latest.

You will also notice that the setup information saved by the “Z” command is erased. Presets 2-63 are unchanged, but Preset 1 has been cleared to all zeros since the default Playback Mode is 01 (you don’t want to turn on all the lights full by accident), the Start Code is set to 0, and the Number of Channels to 64 (40 hex).

NV-RAM Notes

The original version of this controller used FRAM for all the memory. This had good and bad points. FRAM is non-volatile (maintains the contents with the power off) and is fast enough to keep up with the DMX output speed, but until recently would burn out too soon. This forced us to set the DMX packet rate (number of times per second that one set of “N” channels is sent out) very slow. This made it hard to control certain devices or get smooth fades without sending constant data using the “D” command.

Now, the processor used in this version has lots of internal static RAM (SRAM) and several other nice features. This is something you just couldn’t find a few years ago. The DMX output is now output from this SRAM (Preset “0”), so the number of reads/writes does not matter. This allows a control program to use the “W” command to just send changes to the active memory. Presets 01-63 are now stored in EEPROM. EEPROM isn’t all that fast, but is only accessed when you change presets or edit it (neither of which should mind a few milliseconds delay). Now here’s the warning: an EEPROM allows unlimited reads but is limited to 1 million writes before it is burned out. This limit is per memory location, so if you constantly wrote to Preset 1 it would eventually burn out (return invalid data) but the rest of the memory would still work fine. So, if you have a need to change presets a lot, change which one you use from time to time.

If you were to change Preset 1 every hour non-stop, it would burn out in 114 years. If you changed it every minute, it would burn out in just under 2 years. Still not bad.

If you DO burn out the memory, you can easily replace it. It’s the only chip in a socket on the board (“U8”, Microchip 24LC256IP) and retails for under \$2.

There’s another possible alternative. Ramtron, the maker of the FRAM we used in the past, has announced a part that can replace the EEPROM in the converter. It will still burn out, but it takes 100 billion cycles, and that’s a long time. The new boards support the FRAM chip, but large chips are not yet available, you would only have 15 presets. If you’re interested in swapping out your EEPROM for the new FRAM, contact us for more information.

The first time you turn on the power after replacing the EEPROM, the firmware will not recognize the EEPROM and will clear the information saved with the “Z” command as well as all of Preset 1. This also happens if a new version of firmware is loaded. This prevents strange things from happening if unknown data is read from the EEPROM upon power up. The defaults are P=1, S=0, N=040 (hex), H=1, Preset 01 all zero.

Power Notes

This converter runs on 3.3V internally and has a voltage regulator that is specified to work from about 4V to 26V. To allow a little margin, we have specified the voltage requirements as 5-24VDC at about 100mA (a DMX short will draw more current but not hurt the converter). The voltage regulator has built-in protection for use in automotive environments (reverse battery, 60V spikes). The lower the voltage you run the converter at, the cooler it will run (24VDC at high ambient temperatures is not recommended). The included AC Adaptor is either 6VDC or 9VDC (6VDC ones are sometimes hard to find). If you ordered the DC version, you have received a DC power cord with a cigarette lighter plug on it. If you intend to power the converter from some other DC source, feel free to remove the cigarette lighter plug or use another DC power cord (check for correct polarity). Here are the specifications for the power plug:

- Co-axial Power Plug
- 2.1mm internal diameter
- 5.5mm external diameter
- Center Positive

Warranty

Durand Interstellar, Inc. warrants this product to be free from manufacturing defects in original material, including original parts, and workmanship under normal use and conditions (“manufacturing defect”) for a period of one (1) year from date of original purchase. A charge will be made for repairs not covered by the warranty.

Should service become necessary, contact Durand Interstellar, Inc. for return authorization and then:

- Pack the unit in a well-padded corrugated box
- Enclose a copy of your proof of purchase, if you are not the original purchaser
- Ship the unit prepaid via an insured carrier

NOTE: This warranty is void if the product is:

1. Damaged through negligence, misuse, abuse, or accident
2. Modified or repaired by anyone other than Durand Interstellar, Inc.
3. Damaged because it is improperly connected to other equipment of other manufacturers
4. Connected to any power source other than the adaptor supplied with the product

NOTE: This warranty does not cover:

5. Damage to equipment not properly connected to the product
6. Cost incurred in the shipping of the product to Durand Interstellar, Inc. to perform warranty repairs
7. Damage or improper operation of unit caused by customer abuse, misuse, negligence, or failure to follow operating instructions provided with the product
8. Ordinary adjustments to the product which can be performed by the customer as outlined in the instruction manual
9. EEPROM burnout after the first 60 days
10. Improper operation of the unit caused by software written by any third party

ANY APPLICABLE IMPLIED WARRANTIES, INCLUDING THE WARRANTY OF MERCHANTABILITY, ARE LIMITED IN DURATION TO THE PERIOD OF THE EXPRESSED WARRANTY AS PROVIDED HEREIN BEGINNING WITH THE DATE OF ORIGINAL PURCHASE AT RETAIL, AND NO WARRANTIES, WHETHER EXPRESS OR IMPLIED, SHALL APPLY TO THE PRODUCT THEREAFTER. DURAND INTERSTELLAR, INC. MAKES NO WARRANTY AS TO THE FITNESS OF THE PRODUCT FOR ANY PARTICULAR PURPOSE OR USE.

UNDER NO CIRCUMSTANCES SHALL DURAND INTERSTELLAR, INC. BE LIABLE FOR ANY LOSS, DIRECT, INDIRECT, INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGE ARISING OUT OF OR IN CONNECTION WITH THE USE OF THIS PRODUCT.

THIS WARRANTY IS ONLY VALID IN THE UNITED STATES OF AMERICA. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. HOWEVER, YOU MAY HAVE OTHER RIGHTS WHICH MAY VARY FROM STATE TO STATE. SOME STATES DO NOT ALLOW LIMITATION ON IMPLIED WARRANTIES OR EXCLUSION OF CONSEQUENTIAL DAMAGE, THEREFORE THESE RESTRICTIONS MAY NOT APPLY TO YOU.