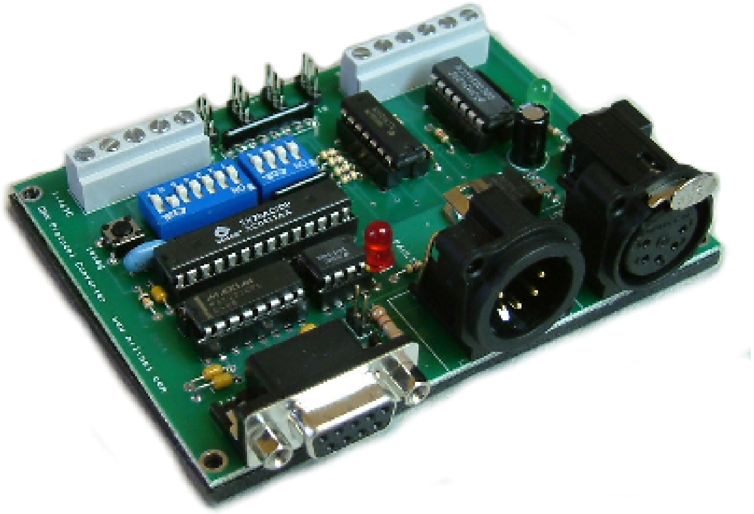
Rev 1.4 22/01/2009

DMX Protocol Converter

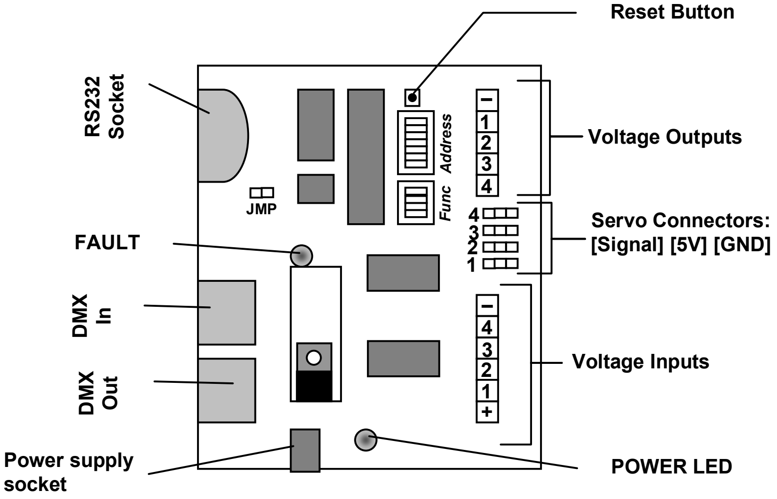


Purpose

This circuit is designed for connection to a DMX loop, specifically for controlling stage lights and special effects. It can receive or transmit DMX code and translate it from and to a variety of I/O channels.

Set-up

The card requires a 9V DC power-supply at 1A, centre positive. For certain modes of operation, the card needs to have an address number. This must be set before power is attached. The function of the card must also be set before power is attached. If either the address or the card function is changed, the reset button must be pressed for the card to fully register the alterations.



Setting the card’s address

The card’s address is used when the card is configured to receive DMX data or if it is transmitting DMX data from a local source. The card can be addressed at any value from 1 to 112. The *Address* DIL block is used for this.
The address is set in binary format. The switches are each numbered, and each switch represents a number:

Switch	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Number	1	2	4	8	16	32	64

To set the address is simple. Turn off all the switches. Take your number and, starting at 64, turn on a switch if it is less than or equal to your address number. When you turn a switch on, subtract that number from your address number. Then continue with your new value.
Example: You want address number 83. All switches are off. You turn on 64. Subtracting 64 from 83 leaves 19 so you leave 32 off but switch on 16. This leaves you with 3. Therefore, you leave 8 and 4 switched off but switch on 2 and 1.

Setting the card’s function

The card can receive or transmit via a number of channels, and the *Function* DIL block is used to tell the card what to listen for and what to send out

R/T	Receive or Transmit DMX512
L/R	Input from Local control via the ADC chip or RS232
An/R	Output on one of the following analogue outputs or RS232
S/D	Output to control a servo or DAC for a 0-5V output

Setting the jumper JMP

The jumper, JMP, must be set if the card is the final unit on the end of a DMX daisy chain. This is to prevent interference on the transmission line. If the card is not the final unit, the jumper must not be set.

Configurations

	OUTPUTS	**RS232**	**Voltage**	**Servo**
Receiving DMX512	*Address*	Required†	Required†	Required†
	R/T	On	On	On
	L/R	N/A	N/A	N/A
	An/R	Off	On	On
	S/D	N/A	Off	On

† The card will search for the relevant frame as set by the address and output four values corresponding to that frame and the next three

	INPUTS	**Local**	**RS232**
Transmitting DMX512	*Address*	Required‡	N/A
	R/T	Off	Off
	L/R	On	Off
	An/R	N/A	N/A
	S/D	N/A	N/A

‡ The card will output the values from the attached voltages in the frames corresponding to the cards address and the next three frames.

DMX Connectors

Twin 5-pin XLR connectors: Pin 1= Ground, Pin 2 = Negative signal, Pin 3 = Positive signal.
Pins 4 and 5 are generally not used.

Serial Connector

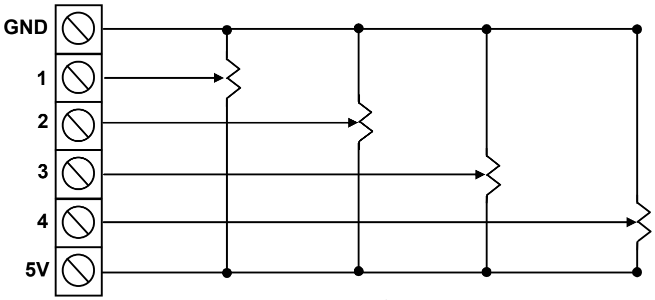
The serial input connector is configured to receive data on pin 3 and transmit on pin 2. Pin 5 is signal ground. If driving from a PC, ensure you use a straight-through modem type cable ie pin2 connected to pin2, 3 to 3, 5 to 5 etc.

Using the card

Transmitting DMX
The jumper, JMP, must not be set. The DMX transmission line must be plugged into the DMX out socket.
The card will read from the selected input and place the data into the appropriate channel in the DMX stream.

RS232: When reading RS232, the card expects data in the form of two numbers. The first number must be the channel number, and the second number must be the value to go into that channel.
E.G.: To set the dimmer at address 34 to half level, you must send [34,127].
The RS232 must be set at 9600 baud, 8-N-1 format.

Local: The card reads the voltage values from the inputs. It then converts from 0-5V to 0-255. This value is then sent to the appropriate channel. The card's address dictates where input 1 goes. Input 2 goes to (address+1); input 3 goes to (address+2) and input 4 goes to (address+3). Ground and 5V are marked on the diagram as '–' and '+' respectively. Use a potentiometer rated at about 10k. There will be three terminals. The outer terminals should be connected so: one to the '+' and one to the '–'. The middle terminal should be attached to the appropriate channel.



Wiring diagram for potentiometers

Receiving DMX
The jumper, JMP, must be set if (and only if) the card is the final unit on the end of a DMX chain.
The card will read from the DMX stream and extract the values in the channel as set by the card address, and the next three channels.

RS232: The card repeatedly sends out the values read from the four DMX channels, first channel followed by the 3 subsequent channels. There is a variable pause (duration depends on the number of DMX channels being sent and data rate) between the 4 byte strings. The RS232 is sent at 9600 baud, 8-N-1 format.

Voltage Out: There are four outputs corresponding to the four selected channels. Each of these will give a voltage from 0 to 5V corresponding to a channel value of 0-255. The voltage is given with respect to the ground terminal, which is marked as '–' on the diagram.

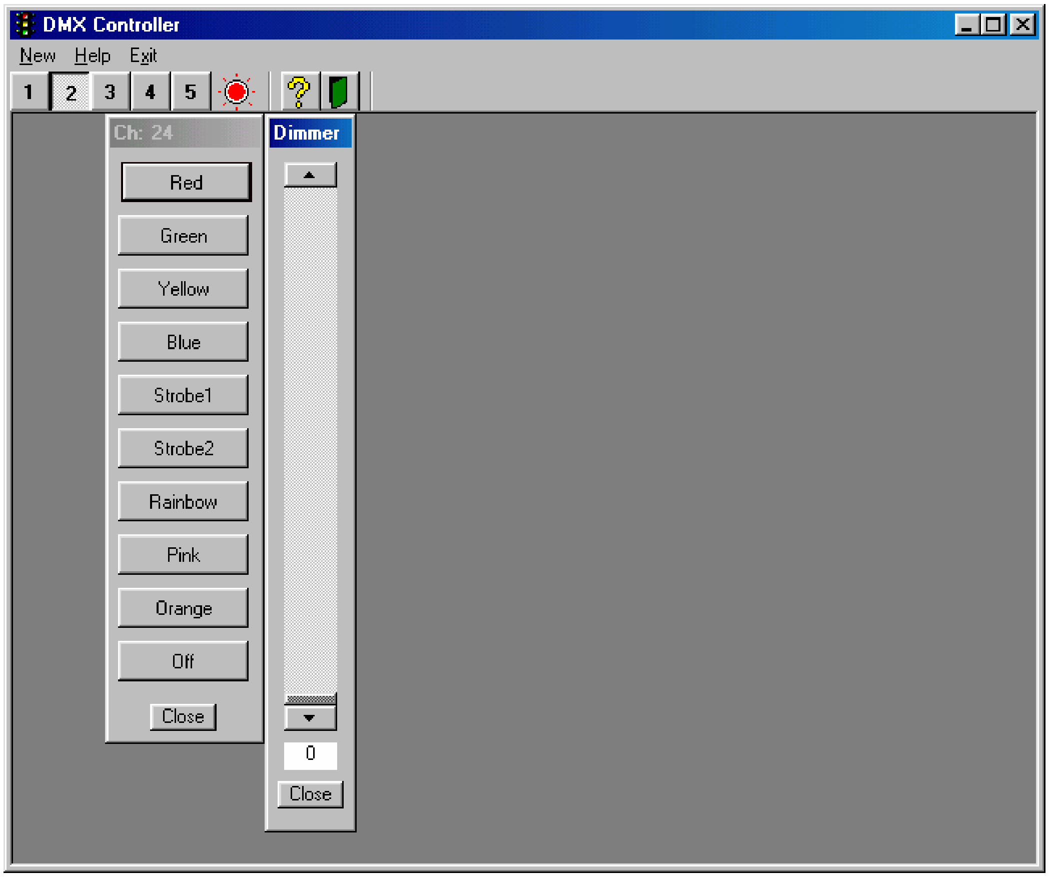
Servo: The four servos' positions are controlled by a value from 0-255. This value is translated to a pulse width for setting the servos' position. A value of 0 gives a pulse width of 1.03ms. A value of 255 gives a pulse width of 2.01ms.

The FAULT LED will light if:

a) The DMX card does not contain a channel that corresponds to the address of the card, e.g. if the card address is 56 and there are less than 56 channels in the incoming stream.
b) The DMX stream is stopped for any reason.

Software

Please check our web site for Windows freeware to accompany the DMX card- **www.milinst.com**.
The current software allows the PC to be used as a simple lighting/effects controller



With this application, the user can control devices that receive a range of values (like a dimmer) or a number of discrete values (like a colour wheel). The Visual Basic source code is included. A programmer may like to add controls for altering a lamp's position on an X or Y-axis, or specific devices for custom made special effects.
The application uses an MDI interface to allow the user to have as many controls of the right kind open at any time.

The application opens COM port 2 by default; by pressing one of the 5 numbered buttons in the toolbar, the user can use a different COM port. Because this is application is designed as a guide for further development only, it contains limited error trapping. As a result, any comms errors result in the application closing. You are advised not to try an open a COM port that is in use (by your mouse, for instance) or one that does not exist.

The red indicator tells you whether the selected port is open or closed. The port's status can be toggle by clicking on this indicator.

Sliders and presets can be generated via the menu or keyboard short-cuts (Ctrl+S for a new slider, Ctrl+P for a new preset). When a new control is generated, you will be asked to assign it a channel number (from 1 to 112) and given the choice to name it. The default name for a control is the Channel number. If you are generating a new preset, you will then be able to assign names to the buttons, and the value that they send out.

The other two buttons on the tool bar access the help, and exit the application.

The web site also includes the VBASIC source code for customer use. Please note that this is not supported by Milford Instruments.