

# Getting Started

This guide should serve as a road map for initial installation of the DecaBox DMX to RS-232 Field Programmable system. Unfortunately, since every installation is unique, it's impossible to provide a one-size-fits-all solution. However, when we help customers via telephone or remote desktop, here are the general steps we follow.

For purposes of this document, we'll walk step by step through controlling a Digital Projection M-Vision Laser 18k Video Projector via several channels of DMX data.

The general routine looks like this::

1. Obtain the RS-232 command set for the device in question and test these commands using a PC and serial interface.
2. Add the commands to the DecaBox patch file.
3. Confirm that the DecaBox output strings are correct
4. Connect the DecaBox to the projector and verify proper operation.

Each step will be addressed below.

## Step 1: Obtain a Command Set and Test Using a PC

A detailed command set for this projector can be found at this link. For demonstration purposes, we will implement 'Power On' and 'Power Off' commands.

On page 33 of the document we learn the required baud rate and serial settings:

### Serial Port setup

- Baud rate 9,600 bps
- Data length 8 bits
- Stop bits one

The detailed generic command structure is on page 34:

### Protocol commands

Thus, all commands begin with the `*` character and end with the carriage return. There are spaces between characters for easy readability.

`*command operator <value>↵`

- The `<command>` string determines which setting the command will affect.

#### Examples

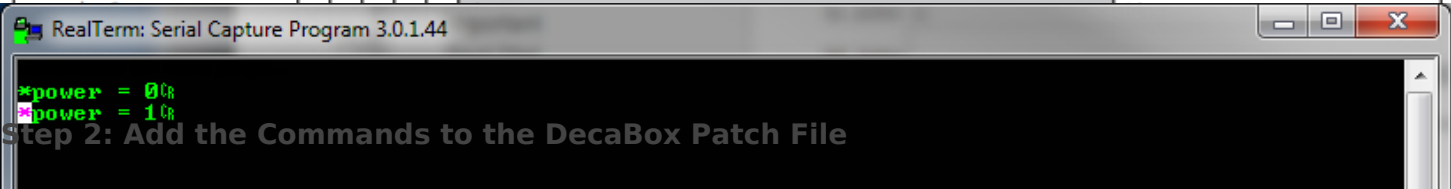
- `*orientation = 3↵` sets the orientation to Rear Ceiling (for a ceiling mounted projector positioned behind the screen)
- `*aspect.ratio ?↵` asks what the current aspect ratio is
- `*zoom.in↵` commands the projector to zoom in
- `*orientation=3↵` is an invalid instruction because of the missing spaces before the operator and the value

Execute		Performs an action. No operator is entered for this type of command.
---------	--	--

On page 58 we see the power command:

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Ext	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
Miscellaneous								
power	●	●	×	×	×	0 = Off 1 = On		
shutter	●	●	×	×	×	0 = Open 1 = Close		n/a
*power = 0[cr] ← Off *power = 1[cr] ← On								
status	×	●	×	×	×	0 = Standby 1 = Warm Up 2 = Imaging 3 = Cooling 4 = Error		
errcode	×	●	×	×	×	string		

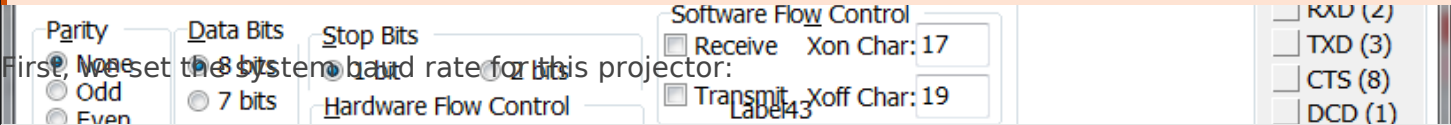
Connect the projector to a PC running a terminal program (we like RealTerm) and send the data to the projector, confirming that the command syntax is correct. The projector should respond to commands.



Step 2: Add the Commands to the DecaBox Patch File

The DecaBox uses a file called ‘patch3.txt’ to bind serial strings to specific DMX channel:value combinations. A sample file is included at the end of this document. The file is also shipped on the SD card of each DecaBox. Remove the four screws holding the chassis together to access the SD socket.

This file **must be edited** in NotePad or TextEdit. Don’t use Word or other complex programs - they introduce garbage characters which will cause the system to fail.



First, we set the system baud rate for this projector:

```
; Set the global baud rate. Default here is 8N1, no handshaking; Regular options are 9600 19200
38400 57600 115200
; no commas are needed
;
baud 9600
;
```

```
; This document is processed at startup.
```

```
;
```

Then, let's assign the power command two ways:

1. Assign power on and power off to a pair of DMX channels at 95%+
2. Assign power on and power off to a single DMX channel in two sections of the [0 255] range

We'll add these commands to the end of the file. Note the use of \$0D to designate the carriage return:

```
;Assign DMX channels 10 and 11 at 95%+ to trigger on and off
10 250 255|*power = 0$0D
11 250 255|*power = 1$0D
; Assign DMX channel 12 to trigger on and off throughout the range:
12 10 127|*power = 0$0D
12 128 255|*power = 1$0D
```

When the dust settles, 'bumping' channel 10 or 11 will trigger a single transmission of either command. Running channel 12 between 0 and 100% will generate many, many copies of the command. When programming cues in a lighting console, an example cue might look like this:

**CH12 @ 40% Time 0**

**CH12 @ 100% Time 0**

### Step 3: Connect the DecaBox to a PC and Test Using a Terminal Program

Now that the commands have been created, save the patch3.txt file back to the SD card, insert it in the DecaBox, and re-add power. After several seconds, the system will be stable and ready to process DMX data.

When connecting the DecaBox to a PC, a null modem / crossover cable *is required*. The reason for this is that the DecaBox pretends to be a PC as far as the serial port wiring is concerned. When connecting two PCs together, a cable which swaps pins 2 and 3 from end-to-end will allow them to talk with each other. As printed on its panel, the DecaBox transmits data on pin 3. In this firmware build, it does not receive serial data, but that data would be accepted on pin 2. Ground is pin 5.

Here we connect a DMX source to the DecaBox and then bump channels 10 and 11 to 100%:

RealTerm: Serial Capture Program 3.0.1.44

\*power = 0

\*power = 1

On the DecaBox, the left LED will illuminate and pulse when valid DMX data is present. When an RS-232 command is triggered, the right LED will flicker briefly as the string is transmitted.

Comparing the received commands to those generated earlier indicates that they are correct.

Step 4: Connect the DecaBox to the RS-232 Equipment and Test

This can be the trickiest step. Depending on the receiving equipment's serial pinout, a crossover cable may or may not be required to make the final connection. Try both to be sure.

DisplayPortCapturePinsSendEcho PortI2CI2C-2I2C MemI2C MiscMisc

Baud9600Port2OpenSpyChange

ParityNoneData Bits8 bitsStop Bits1 bit2 bits

OddEvenMarkSpace

Revision #5Created Mon, Nov 12, 2019 9:00 AM by ESINCUpdated Wed, Jan 29, 2020 3:57 PM by ESINC

Hardware Flow ControlNoneRTS/CTS

DTR/DSRRS485-rtts

Software Flow ControlReceiveXon Char: 17TransmitXoff Char: 19

Winsock is:RawTelnet

Status

Disconnect

RXD (2)

TXD (3)

CTS (8)

DCD (1)

DSR (6)

Ring (9)