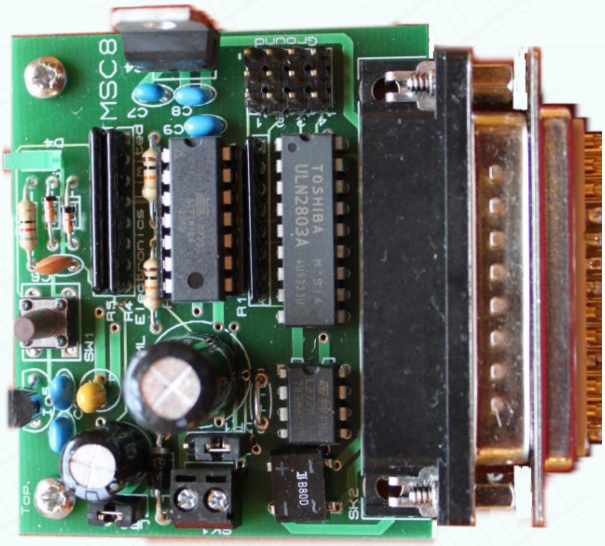
Sig-naTrak®

Model railway electronics by GFB Designs

MSC8

Scenery Animator



User Manual

DCC Compatible
Motor, Servo & Lighting Controller
for 8 accessories.

***Sig-naTrak*®** **by GFB Designs**
www.signatrak.co.uk
67 Boddens Hill Road, Stockport, SK4 2DG, United Kingdom
T: +44(0)161 883 2022 | F: +44(0)161 883 2077 | E: info@signatrak.co.uk

Sig-naTrak®

Model railway electronics by GFB Designs

IMPORTANT NOTICE

CHANGE OF OWNERSHIP AND TRADING NAME

Please note that following the purchase of CML Electronics by GFB Designs, some of the product manuals may still refer to “CML Electronics Limited”.

This should now be read as
“Sig-naTrak® by GFB Designs”.

Contact details listed in this manual are also incorrect and the details at the bottom of this page should take precedence over any found in this manual for CML Electronics.

This includes our website which is now
www.signatrak.co.uk

***Sig-naTrak*® by GFB Designs**
www.signatrak.co.uk
67 Boddens Hill Road, Stockport, SK4 2DG, United Kingdom
T: +44(0)161 883 2022 | F: +44(0)161 883 2077 | E: info@signatrak.co.uk

1 Description

The MSL8 Scenery Animator controls a motor, up to 4 servos and several lighting outputs for animation of static scenery on a model railway. It can be used on DCC or DC controlled railways and allows a variety of manual or automatically controlled effects to be achieved.

The Scenery Animator can generate timed effects to control its outputs. This allows effects such as welding, disco lights and house lights to be animated using LEDs or miniature lamp bulbs. Servos can operate position sequences: for example to move cranes, or to move a windmill to face into the wind.

Output 1: DC or AC synchronous motor (can also drive 2 flashing lights)
Outputs 2-5: servo, or lamp
Outputs 6-8: lamp

2 Connecting it up

The main connections to the unit are via the 25 way connector SK2. This provides the DCC connection and all output connections except for servo outputs. Connections may be made via a solderable mating connector (provided) or alternatively via a ribbon cable and connector supplied as an optional extra.

Connector pin	**signal**	
1, 2	DCC rail A in	DCC rail connection.
14, 15	DCC rail B in	DCC rail connection.
3, 16	pushbutton	Connect to an external pushbutton if needed for on/off control of all outputs.
4, 17	motor out	Connection to a DC motor or AC synchronous motor.
5, 18	Not used	
6, 19, 20, 21, 22, 23, 24, 25	+12v common	Common return wire for all lighting function outputs.
7	Light function 8	Connects the seven lighting function outputs. Each output can drive LEDs or miniature filament bulbs. If driving LEDs, series resistors will be required as shown. Any one output, and the total current load, should not exceed 0.5A.
8	Light function 7	
9	Light function 6	
10	Light function 5	
11	Light function 4	
12	Light function 3	
13	Light function 2	

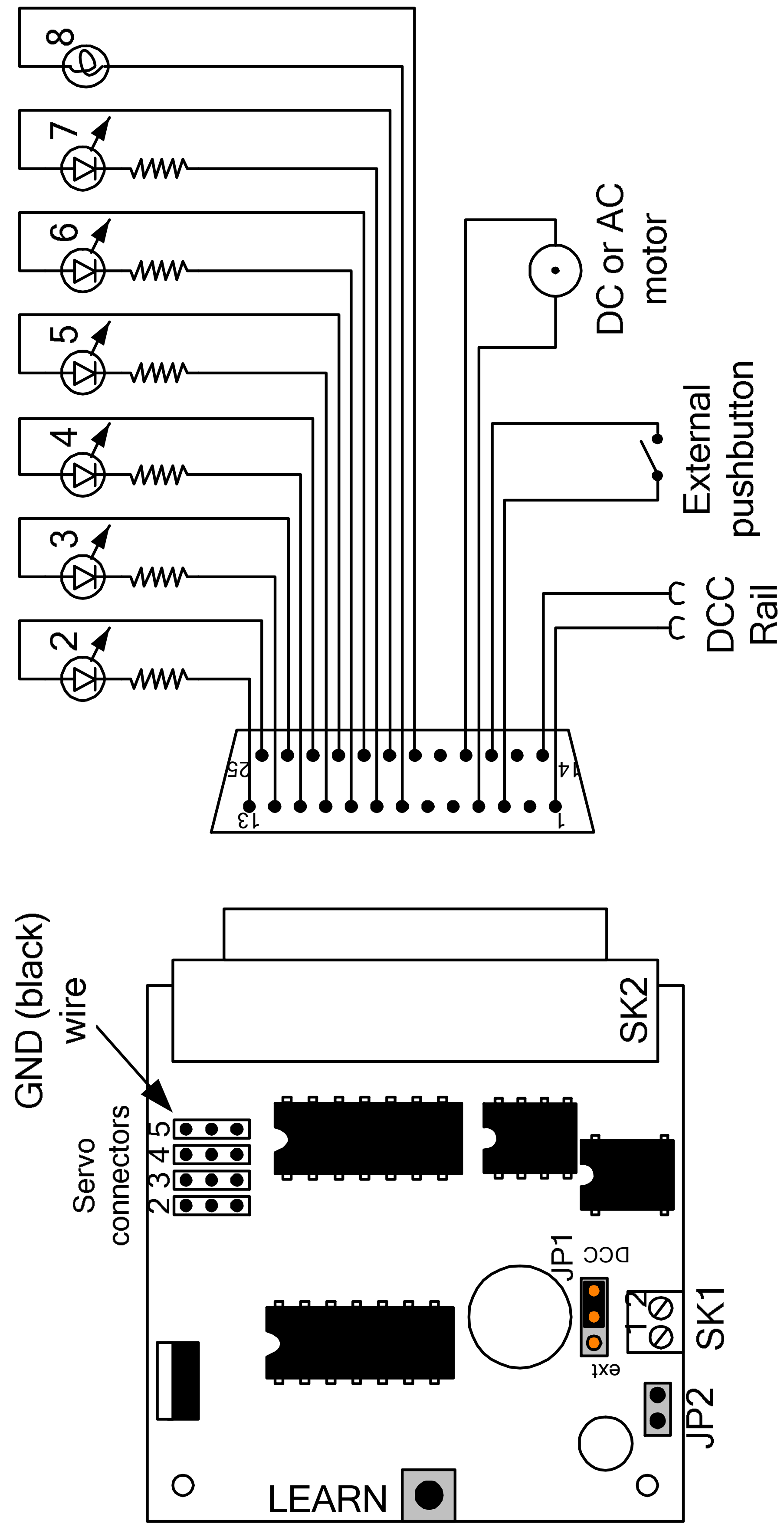


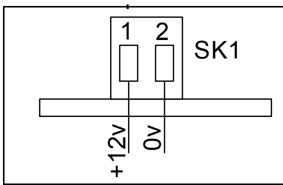
Figure 1: Connecting up the MSL8

If the ribbon cable option is used, the pinouts allow the cable to be split into a series of pairs to connect to the various inputs and outputs. This can make connection very simple!

2.1 Power Input

The MSC8 can operate from the DCC rail input. In this case all power for the outputs is taken from your DCC booster. Position jumper JP1 to position “DCC”.

It is also possible to provide an external power feed to the unit, to minimize power drain from your DCC system. In this case connect a DC supply of 12-15v to SK1 as shown and set jumper JP1 to position “ext”.

<table>
<tr><td rowspan="4">
</td><td colspan="3">SK1: 2 pin screw terminal</td></tr>
<tr><td><b>Pin</b></td><td><b>Function</b></td><td><b>Signal Level</b></td></tr>
<tr><td>SK1 pin 1</td><td>+ve DC power</td><td rowspan="2">Connects to +12 to +15v DC supply, rated at 0.5A to 1A.</td></tr>
<tr><td>SK1 pin 2</td><td>Ground</td></tr>
</table>

2.2 Servo Outputs

Up to 4 servos can be connected to the Scenery Animator, controlled by outputs 2-5. The servos plug onto a connector on the board as shown.

Note: most, but not all, servos have pinouts that the board is already wired for. The board supports connections using 0.1” pitch 3 pin connectors with pins in the order “Signal, +ve supply, ground”. These often have colours “white, red, black” or “orange, red, black”. **It is your responsibility to check** that your servo has these connections. It is usually possible to swap the pins within the servo plug if needed.

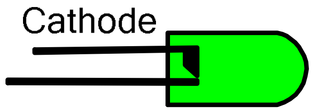
2.3 Motor Output

The motor output can drive small, lightly loaded motors connected to scenery objects e.g windmills. The unit can drive a DC motor at up to 0.5A maximum, or a 50Hz synchronous AC motor for example as is used in the “Faller” range of kits.
This output may also be used to drive a pair of blinking lights for a level crossing type application. In this case the motor output each connect to a lamp; the other ends of the lamps should be connected to the +12v return.

2.4 Lighting Outputs

Lighting outputs may be used to drive LED lamps, or miniature incandescent lamps. LEDs will need a series resistor – choose a higher value to make the LED dimmer. As a suggestion begin with a value of 1k and work from there.

The LED "Anode" is the terminal which is positive when the LED is lit. The "Cathode" is the pin that is connected to the negative supply. The anode often has a longer lead than the cathode.



Be aware the total load should not exceed 0.5A. This will not normally be reached with LEDs and "grain of wheat" type lamps.

3 Programming Board Settings

The board has many Configuration Variables (CVs) to control its operation. This section describes how to program them. CVs may be programmed on a programming track or "on the main". A factory reset to initial settings is also possible: this operation also sets the base address for DCC operation.

The CVs themselves are described later in this document, and are listed in an appendix.

3.1 Programming Track

To program the unit on a programming track take the following steps:

- Connect the unit's "rail" inputs to a programming track
- Disconnect all servos and motors from the unit.
- Select jumper position JP1 = "DCC" and remove JP2
- Use your command station in "paged" or "direct" mode
- Access the CVs following your command station instructions. It is possible to read and write CV values, if your command station permits those operations.

3.2 "On the Main"

The unit can be programmed while on a normal live DCC track. This permits changes to be made but doesn't allow CV values to be read back. To use this mode:

- Ensure the unit is connected to the track and the command station is operating; the green LED should be lit.
- Press and hold the pushbutton for more than one second. The green LED will begin to flash.
- Select an unused locomotive number on your DCC system.
- Enter "programming on the main" mode (sometimes called "ops mode")
- Select a CV number and new data and follow the command station instructions to write the CV
- The green LED will flicker momentarily to indicate success.
- After all programming is complete, press the pushbutton momentarily. The unit will restart with the new settings.

3.3 Factory reset/ Setting base address

The board has a quick way to set its DCC base address AND reset all CVs to factory default values. This is useful both for initial address setting and to recover if the settings have become confused. To use this mode:

- Ensure the unit is connected to the track and the command station is operating; the green LED should be lit.
- Press and hold the pushbutton for more than one second. The green LED will begin to flash.
- Use the command station to set a point (sometimes called "switch") address to THROWN.
- This address will be memorized by the board, and all other settings will be reset.

3.4 Manually Setting the base address

The unit's main base address can be set by programming CVs 1 and 2 either on a programming track or "on the main" as described above. A table of CV values to set any base address is in an appendix.

The board has a second base address set by CVs 3 and 4. This is used to set servo positions depending on the setting of two accessory outputs allowing a servo to be set to 4 possible positions. See section 6.3.

4 Configuring the Outputs

4.1 Output types

Each output can be separately programmed. The output typers are set by CVs 9-16. Allowed values for the cell types are shown in this table:

CV value	**Cell type**	**Can be used on outputs:**		
		1	**2-5**	**6-8**
0	Lighting, instant on/off	-	Y	Y
1	Lighting, fast on & off time	-	Y	Y
2	Lighting, medium on & off time	-	Y	Y
3	Lighting, slow on & off time	-	Y	Y
4	Servo, moves between P1 & P2 positions	-	Y	-
5	Servo, moves to 4 P1-P4 positions	-	Y	-
6	Servo, position set by animating sequence	-	Y	-
7	DC motor, speed set by CV 6	Y	-	-
8	50 Hz AC motor synchronous	Y	-	-
9	Two "crossing" lights, blink rate set by CV6	Y	-	-

- "Lighting" type outputs are controlled on/off by the DCC controls or by a lighting effect. The "brightness" can be ramped up and down at fast, medium or slow rates to look like particular kinds of light.
- Servo outputs are available on outputs 2-5. They allow physical movement to be controlled in several ways. See section 6 for more details.
- Output 1 can drive a motor. It is intended to drive a simple, lightly loaded motor that drives a scenery object – for example the sails of a windmill.
    - If a DC motor is used, its speed is set by CV6. The speed value is given by a value 0-127, with 127 being fastest. To make trhe motor run in the reverse direction, add 128 to the value.
    - A 50Hz AC synchronous motor can be used. Faller models often use these motors for example.
    - The output will also drive two alternately lit lights that can be used for a level crossing. In that case, connect the lights or LEDs between these two motor output and the +12v common" pins on SK2.

4.2 Effect Sequences

Effect sequences can be used to add realism to the outputs. They turn an output on or off following a timed sequence. There are 8 factory defined sequences programmed into the board, and 4 user defined sequences can be created by programming CVs.

A sequence is a set of 256 on / off settings. The Scenery Animator steps through these 256 on/off settings at a defined rate. By setting the on/off sequences and choosing a suitable rate, the Scenery Animator can create realistic effects.

<table>
<tr><th>Output</th><th>Sequence number CV</th><th>Sequence start position CV</th><td rowspan="9">Program a sequence for an output by putting the sequence number (1-12) into the output "Sequence Number" CV. Sequences 1-8 are factory defined; sequences 9-12 are user defined.

Sequences also have a "Start" CV which sets where the sequence starts in the set of 256 on/off states. This can be used to extend the range of effects available.</td></tr>
<tr><td>1</td><td>CV17</td><td>CV32</td></tr>
<tr><td>2</td><td>CV18</td><td>CV35</td></tr>
<tr><td>3</td><td>CV19</td><td>CV38</td></tr>
<tr><td>4</td><td>CV20</td><td>CV41</td></tr>
<tr><td>5</td><td>CV21</td><td>CV44</td></tr>
<tr><td>6</td><td>CV22</td><td>CV47</td></tr>
<tr><td>7</td><td>CV23</td><td>CV50</td></tr>
<tr><td>8</td><td>CV24</td><td>CV53</td></tr>
</table>

The 8 factory defined sequences are:

1. **Arc Welding light**. This simulates the blue flickering light associated with arc welding. The sequences pauses periodically while the welding rod is changed. Use with a blue LED and the output programmed to "Lighting, instant on/off"
2. **House light 1**. This simulates a light in a house, turning on and off periodically as the occupier moves around. Use with a while LED and the output programmed to "Lighting, medium on & off time". If the house has several rooms: use the same sequence for two or more outputs, and set the start

position to values such as 50/100/150/200 so the lights don't turn on and off at the same time

3. **House light 2**. As sequence 2, but the room is occupied less often.
4. **Emergency vehicle light 1**. Simulates flashing warning lights. Use with a blue or red LED and the output and the output programmed to "Lighting, instant on/off"
5. **Emergency vehicle light 2**. Simulates flashing warning lights. Use with a blue or red LED and the output and the output programmed to "Lighting, instant on/off".
6. **Flickering light**. Simulates a candle or oil lamp flickering in a draught. Use with a white or yellow LED programmed to "Lighting, fast on & off time"
7. **Smoke generator**. The output is active for approximately one minute every 5 minutes. Use on the "motor" output to operate a smoke generator.
8. **Disco light**. Simulates disco lights flashing as music plays. There is an "off" period for a few seconds while the record is changed. Use with a coloured LED. Consider a second output with a different LED set to the same sequence, with a small "start position" value of 5-10.

5 Controlling The Outputs

The unit's 8 outputs can be controlled in several ways. If an output doesn't have a sequence assigned to it, then these controls turn the output on and off. If a sequence is assigned, these controls turn the sequence on and off and the sequence then controls the output.

5.1 DCC Commands

DCC point (accessory) commands can be used to set outputs on or off: this is the most common method to control the unit. The unit responds to 8 DCC point commands beginning at the base address set by CVs 1&2 (see section 3.4 above). The first point number controls output 1; the last controls output 8. Depending on how the unit is programmed this will either turn the output on or off, or turn an effect controlling the output on or off.

Outputs 2-5 can control radio control servos: these are often useful to move parts of scenery around. The "on/off" command simply move the servo between two positions. The positions can be programmed with CVs. The unit can also create a simple "bounce" effect at the end of the movement. This is useful to create a realistic effect for a signal arm moving, for example.

5.2 Pushbutton Control

The board has a pushbutton. When pressed this will turn all outputs on or off. This can be used with a simple DC power feed and no DCC system.

If you wish to have another pushbutton to control all the outputs – for example mounted on a fascia – then an external pushbutton may be connected to the main connector: see section 2.

6 Using servos

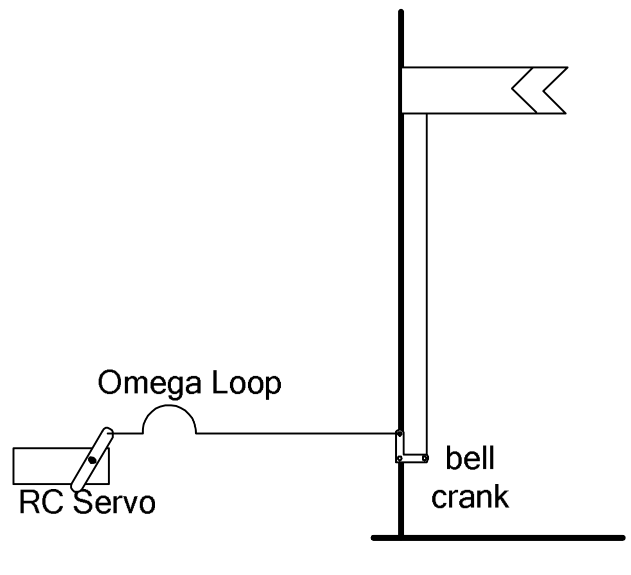
6.1 Choosing & using servos

Radio Control servos are becoming popular for animating accessories on model railways. They are small, low cost and powerful. The Scenery Animator can control 4 servos in several ways. Even the smallest "sugar cube" servos will move a substantial mechanical load. They can be used to operate point motors, semaphore signals, cranes etc.

There are hundreds of RC servos available. The MSL8 will drive most types, and we've tested it will a range of common ones; we can't guarantee it will drive all of them.

Servos will usually take a significant current when operating: 100-200mA is common. The total load from the Scenery Animator should not exceed 500mA, so avoid having several servos all moving at once. After their movement is complete, the servo control signal can be turned off and the power reduces to a tiny amount. As long as the servo isn't pulling against a return spring, it will stay in its current position.

The board will normally "know" the servo position when it is first switched on, so unless the servo has been moved while power was off it will stay in its previous position.



Make sure just in case that your model will not be damaged by any sudden movement of the servo output.

Putting a bent spring wire into the servo operating output – known as an "omega loop" - is a wise precaution.

6.2 Servo Programming

The servo behavior is set by programming the output type (see section 4.1) and a set of 7 CVs for each servo output. This section describes the settings.

The servo output is usually a shaft which turns through approximately 180º. The Scenery Animator sets the output position between 0 and 255, corresponding to opposite ends of the servo's output range.

The simplest servo operation is to move between positions 1 and 2. A DCC "thrown" command moves it to position 2; a DCC "closed" command moves it to position 1.

A "bounce" can be programmed. The servo overshoots its target position by a programmed amount at each end, then returns to the final programmed position. This can be disabled by programming the "bounce" value to 0.

The servo movement rate is set by the "Speed" CV. This allows the movement speed to be set from a very slow movement through to approx half a second full scale move. Experiment with settings to see what value is appropriate for your model.

Servo power can be removed after it has finished moving: this should normally be enabled to save power. Set a value between 1 and 255, which sets a delay of up to 4 seconds. A setting of 64 will be a delay of approx 1 second which should be sufficient.

Feature	**o/p 2**	**o/p 3**	**o/p 4**	**o/p 5**	**Meaning**
Position 1	CV54	CV61	CV68	CV75	"closed" position
Position 2	CV55	CV62	CV69	CV76	"thrown" position
Position 3	CV56	CV63	CV70	CV77	See section 6.3
Position 4	CV57	CV64	CV71	CV78	
Bounce	CV58	CV65	CV72	CV79	Servo "overshoots" by this amount
Speed	CV59	CV66	CV73	CV80	1=slow; 255=very fast
Off Delay	CV60	CV67	CV74	CV81	0=always powered; 64 = 1s delay before power removed

6.3 Setting Servo to 4 Positions

Sometimes it is appropriate for a servo to be set to more than two positions. For example, some semaphore type signals have a third position. To allow a servo to be set to 4 possible positions, the position can be set by pairs of DCC accessory commands; the board also responds to 8 DCC point commands at a second base address set by CVs 3&4 for this. Two point settings per servo are used to set up to 4 servo positions.

The combinations of settings are shown below. Each output needs to be specifically programmed to enable this mode: see section 4.1.

Point Address	**Servo Selected**	**Two point settings:**	**Servo Position**
Base address, base address +1	1	Closed Closed	1
Base address+2, base address +3	2	Closed Thrown	2
Base address+4, base address +5	3	Thrown Closed	3
Base address+6, base address +7	4	Thrown Thrown	4

6.4 Servo Movement CV

When a servo moves, it usually takes a fair current from the power supply: often 100-200mA. CV26 can be used to force servos to move one at a time, minimizing the current drawn. If this CV is 0 (default) then only one servo will be allowed to move at a time. The others will wait until movement has finished before moving themselves.

6.5 Animated Sequences

A servo can also be set to move according to a user defined sequence of 32 positions. This is described on our website.

7 Advanced operations

7.1 User Defined Sequences

You can define 4 sequences in addition to the 8 factory defined sequences. Instructions, and example sequences, will be provided on our website.

7.2 Controlling More than One Output

Sometimes it may be appropriate for two outputs to be controlled together. For example: turning on or off all of the outputs that control lights in each room of a house; turning on or off two outputs used as disco lights; and turning on or off a smoke generator, flickering lights simulating the glow from a fire, and the lights on a fire engine.

There are two ways this can be done:

1. The on/off controls for several outputs can be tied together, using the "**DCC Gate**" CVs. This means that all the sequences are turned on or off by a single

DCC command, e.g. for the house lights or disco lights. This mean a DCC command to turn on output 2 can also turn on outputs 1&7, for example.

2. One sequence output can turn on or off another sequence output, by using the "**Output Gate**" CVs. This allows the "smoke generator" sequence (which is active for 1 minute every 5 minutes) to turn on the flickering lamp sequence which simulates the fire using a red LED.

Look on our web page for examples of how to use this!

<table>
<tr><td>Output</td><td>DCC gate CV</td><td>Output Gate CV</td><td rowspan="9">The CV value is formed by adding together numbers for each additional output to be controlled by as follows:
o/p 1: add 1; o/p 2: add 2; o/p 3: add 4; o/p4: add 8; o/p 5: add 16; o/p 6: add 32; o/p 7: add 64; o/p 8: add 128.

EXAMPLE: the DCC command for output 5 should also turn on outputs 2 and 6.
Program CV42 with (2+32)=34.</td></tr>
<tr><td>1</td><td>30</td><td>31</td></tr>
<tr><td>2</td><td>33</td><td>34</td></tr>
<tr><td>3</td><td>36</td><td>37</td></tr>
<tr><td>4</td><td>39</td><td>40</td></tr>
<tr><td>5</td><td>42</td><td>43</td></tr>
<tr><td>6</td><td>45</td><td>46</td></tr>
<tr><td>7</td><td>48</td><td>49</td></tr>
<tr><td>8</td><td>51</td><td>52</td></tr>
</table>

8 Appendix: Table of CVs

CV	Name	Meaning	Default
1,2	Base Address	Decoder base address (low byte 1st)	201
3,4	Servo Base address	Extra base address for servo 4 position setting (if 0: not used)	0
5	Stored position	Memorised position of each output	0
6	Motor speed & direction	Motor speed, 0=min 127=max. add 128 to reverse direction. Also flash rate for blinking lights; value 127 is ~1s on, 1s off; smaller number=faster flash)	127
7	Mfr version	NMRA manufacturer version number	1
8	Mfr ID	NMRA manufacturer ID	1
9-16	Output Type	• 0=lighting, no PWM • 1-3=lighting, fast/medium/slow PWM • 4=servo, moves between 2 positions • 5=servo, moves between 4 positions • 6=servo, position set by sequence • 7=DC motor; 8=AC motor; 9=blinking light pair (output 1 only)	7, then all 1
17-24	Sequence assign	Assign output to sequence 0=none; 1-8=factory sequence; 9-12 = user sequence	0,1,2,3, 4,5,6,7
25	Servo position	Reserved	0
26	Servo movement	If 0, only one servo allowed to move at a time	0
29	Config register	NMRA defined. Always set to 128	128
30-53	Output data. 3CV: output gate, DCC gate, seq start pos	1 bit per o/p; if 1, that output gates this one on 1 bit per o/p; if 1, that DCC cmd gates this on 0-255 sequence start position (to randomise effect)	all 0
54-81	Servo data. 7 CVs: Position 1-4 Bounce value Rate value Off Delay	1-4: desired position for this servo (0-255) 5: Bounce: under or overshoot (0-31) 6: Speed (1=slow; 255=fastest) 7: Delay before power off. 0=always on; 255=~4s.	50,200,100, 150, 10, 1,128 4 times

82-93	User defined seq:
	Sample rate;
	Repeat count;
	Delay after seq
		Settings for the 4 user defined sequences.
		Rate: sample rate in 20ms steps* (1=50Hz)
		Repeat: no. repeats before delay (1=run once)
		Delay: delay before restart (0 to 255 sec)
			0

94-125	User sequence 1	Arrays of 32 1/0 sample words	0
126-157	User sequence 2	Arrays of 32 1/0 sample words	0
158-189	User sequence 3	Arrays of 32 1/0 sample words	0
190-221	User sequence 4	Arrays of 32 1/0 sample words	0
222	Page register	Reserved	

9 Appendix: Base Address values

This chart tabulates the decoder base addresses obtained from different settings of CV1&CV2 **for non ZTC systems**. Although this table shows steps of 10, note that all intermediate addresses can be used. Examples of addresses:

- Base address = 71: set CV2=0, CV1=71
- Base address = 645: set CV2=2, CV1=133

The same table is also used to set the base address for 4 position servo setting using the base address stored in CV3&4.

CV1	**CV2 (Y value)**							
(X value)	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	1	257	513	769	1025	1281	1537	1793
11	11	267	523	779	1035	1291	1547	1803
21	21	**277**	533	789	1045	1301	1557	1813
31	31	287	543	799	1055	1311	1567	1823
41	41	297	553	809	1065	1321	1577	1833
51	51	307	563	819	1075	1331	1587	1843
61	61	317	573	829	1085	1341	1597	1853
71	71	327	583	839	1095	1351	1607	1863
81	81	337	593	849	1105	1361	1617	1873
91	91	347	603	859	1115	1371	1627	1883
101	101	357	613	869	1125	1381	1637	1893
111	111	367	623	879	1135	1391	1647	1903
121	121	377	633	889	1145	1401	1657	1913
131	131	387	643	899	1155	1411	1667	1923
141	141	397	653	909	1165	1421	1677	1933
151	151	407	663	919	1175	1431	1687	1943
161	161	417	673	929	1185	1441	1697	1953
171	171	427	683	939	1195	1451	1707	1963
181	181	437	693	949	1205	1461	1717	1973
191	191	447	703	959	1215	1471	1727	1983
201	201	457	713	969	1225	1481	1737	1993
211	211	467	723	979	1235	1491	1747	2003
221	221	477	733	989	1245	1501	1757	2013
231	231	487	743	999	1255	1511	1767	2023
241	241	497	753	1009	1265	1521	1777	2033
251	251	507	763	1019	1275	1531	1787	2043

Avoid address ranges 1017-1020 (these are used by Digitrax for a special purpose)
For ZTC systems, use the "LEARN" method to set the base address