1. Introduction

The DAC22 is an advanced accessory decoder and controller for model railways operated using Digital Command Control (DCC) systems. It allows DCC command stations to control not only locomotives, but also other devices such as point motors, turntables, signals, lighting and much more. These facilities allow control of an entire layout and accessories through a single digital control system.

The DAC22 can operate as a conventional accessory decoder. Its outputs can be controlled by the digital control system, allowing points to be operated from throttles. Local input switches can be connected, allowing pushbuttons to control the points.

The DAC22 has a LocoNet interface. It can send feedback messages, allowing software such as "Railroad & Co" to know the positions of points when the layout power is turned on. It can be connected to occupancy detectors to send train position information.

The DAC22 has 8 local routes. These allow up to 10 points to be controlled in a single operation; the points can be driven by the DAC22 or by other accessory decoders. The routes can be triggered by a local pushbutton, or by a DCC accessory operation from a throttle.

The DAC22 can also be programmed to carry out advanced operations. Its outputs can be controlled by other point positions, and by the settings of signals and sensors connected to LocoNet. This allows complex automatic control. It allows "illegal" states on 3 way points and slips to be prevented from occurring.

The DAC22 can control points, signals, relays, lamps and light emitting diodes. It allows operations that are as simple, or as complex, as you need them.

Contents

Section 2 provides a "quick start" guide. This allows the unit to be put into operation quickly, without having to understand the full intricacies of the unit.

Section 3 describes all of the connections to the unit.

Section 4 describes the functions of the board, and how to set them up using configuration variables (CVs).

Section 5 describes the advanced programming capabilities.

Section 6 describes the different methods for programming the unit.

Appendix A lists all the programmable Configuration Variables in the unit.

Appendix B lists the address CV settings to set the output addresses if needed.

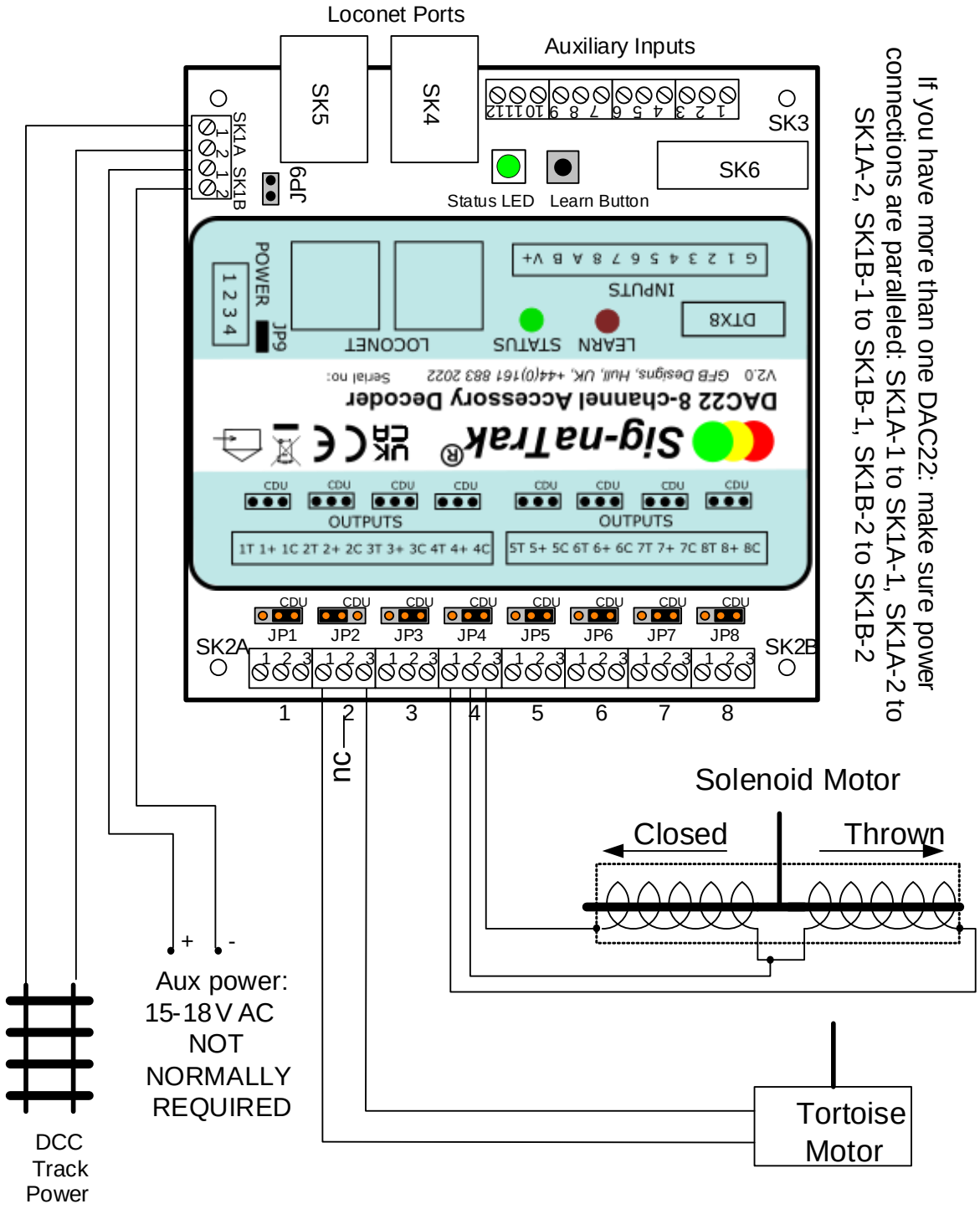
Appendix C provides a decimal to hexadecimal conversion chart if required.

Appendix D describes the LocoNet start-up "Interrogation Sequence".

Appendix E describes DAC22 operation without a track power connection.

2. Quick Start Guide

This section provides a "quick start" introduction to the DAC22. This covers connection of point motors & power, CV programming and simple operations. The diagram below shows the connections needed to provide DCC track power, and to connect output 2 to a tortoise motor and output 4 to a solenoid type motor. To connect other outputs to these motor types, see the table in section 3.2 for the relevant connections.



Aux supply must be a separate transformer winding from DCC booster!

2.1 Safety First!

Before beginning to use the DAC22, there are a few safety points to remember:

Don't rest the board when operating on its bag: it is conductive!
Hold the board by its edges. Some of the pins on the reverse side of the board are sharp and could cause skin abrasions etc if handled incorrectly. Also, electronic circuitry can be susceptible to static electricity from the human body.
Allow airflow around the board. Some of its components may run warm in use. Do not obstruct free circulation of air, or allow cloths etc to cover the board.
Do not exceed rated operating voltage. The board could be damaged if an excessive input voltage is applied. The DCC operating voltage should not exceed 20v; the auxiliary input voltage if used must not exceed 18v AC or 25v DC.
Do not handle the board when in use. The voltages present on the board (<25v DC) are not considered hazardous to health. However if they should come into contact with sensitive parts of the body (e.g. the mouth) a nasty shock might result. The same is true of the voltage on the rails of a DCC (or other model railway) system, so take care!
If you have more than one DAC22 connect the SK1 connections in parallel, i.e. SK1A pin 1 to SK1A pin 1, SK1A pin 2 to SK1A pin 2, SK1B pin 1 to SK1B pin 1 and SK1B pin 2 to SK1B pin 2.`
The auxiliary power feed (if used) must come from a separate power supply from that used by the DCC booster that provides the track power.

2.2 Elementary Programming

The DAC22 is factory programmed to select all outputs as solenoid types, and with an address range of 45-52. Some CVs will need to be reprogrammed for most users' purposes. Details of the CV values and effects are described in section 4.

The board address can be set simply as follows:-

1. Turn on power to the board
2. Press and hold the LEARN button for at least 1 second, then release it. The green STATUS LED flashes.
3. Using a throttle, set the point address required for output 1:
4. If it is set "Thrown": the board sets its base address to the address used, clears all CVs to factory values and sets all outputs to "solenoid" type motors
5. If it is set "Closed": the board sets its base address to the address used, clears all CVs to factory values and sets all outputs to "tortoise" type motors

The elementary CVs which need to be programmed initially are:-

CV	Effect	Values to be programmed to
CV5, CV6	Board base address	Can be set automatically as above. To change the address manually, reprogram according to the settings listed in Appendix B.
CV11	Output 1	To select an output as solenoid type (e.g. Peco, Seep), program the output CV to 3 (the board is shipped with this setting). To select an output as a continuously powered stall motor type (e.g Tortoise), program the CV to 10.
CV12	Output 2	
CV13	Output 3	
CV14	Output 4	
CV15	Output 5	
CV16	Output 6	
CV17	Output 7	
CV18	Output 8	

There are two ways to program CVs: using a programming track, and "ops" mode

1. To use "ops" mode:
   - Press and hold the LEARN button for at least 1 second, then release it. The green STATUS LED flashes.
   - Use the "ops mode" programming mode of your DCC system to change CVs.
   - When finished, press and hold the LEARN button for at least 1 second, then release it. The board will restart with the new settings.
2. To use a programming track:
   - Connect the programming track to SK1A, terminals 1&2;
   - Remove JP9;
   - Follow the **paged mode programming** instructions for your command station.

2.3 Operating the Outputs

Connect the board as shown to the command station, re-insert JP9 and apply power. At this point, any fully powered point motors (e.g. Tortoise) will be driven to one end of their travel or the other. Solenoid type point motors are only powered momentarily when commands to drive them are received so no activity will be noticed. The green STATUS LED on the board should be lit steadily.

Use the command station to move each point motor between THROWN and CLOSED, waiting about 3 seconds in between each output change. "Tortoise" type point motors should slowly traverse to the new state; solenoid motors should move with a sharp "snap". The board allows around a second between output change when using solenoid motors to allow the capacitor discharge unit to recharge.

Each time a command is sent to the board, the green STATUS LED will flicker off momentarily. If this is not observed, the board is not responding to the point address

correctly. Check the point address being driven by the command station and also the programmed base address in the DAC22 (in CVs5&6 – factory programmed to 45).

Unlike the original DAC20, the DAC22 does not usually need an auxiliary power supply to operate solenoid motors reliably as it has a built-in voltage booster circuit. This guarantees that the Capacitor Discharge Unit (CDU) is always charged to between 19.5 and 20 volts when the board is setup for at least one solenoid output.

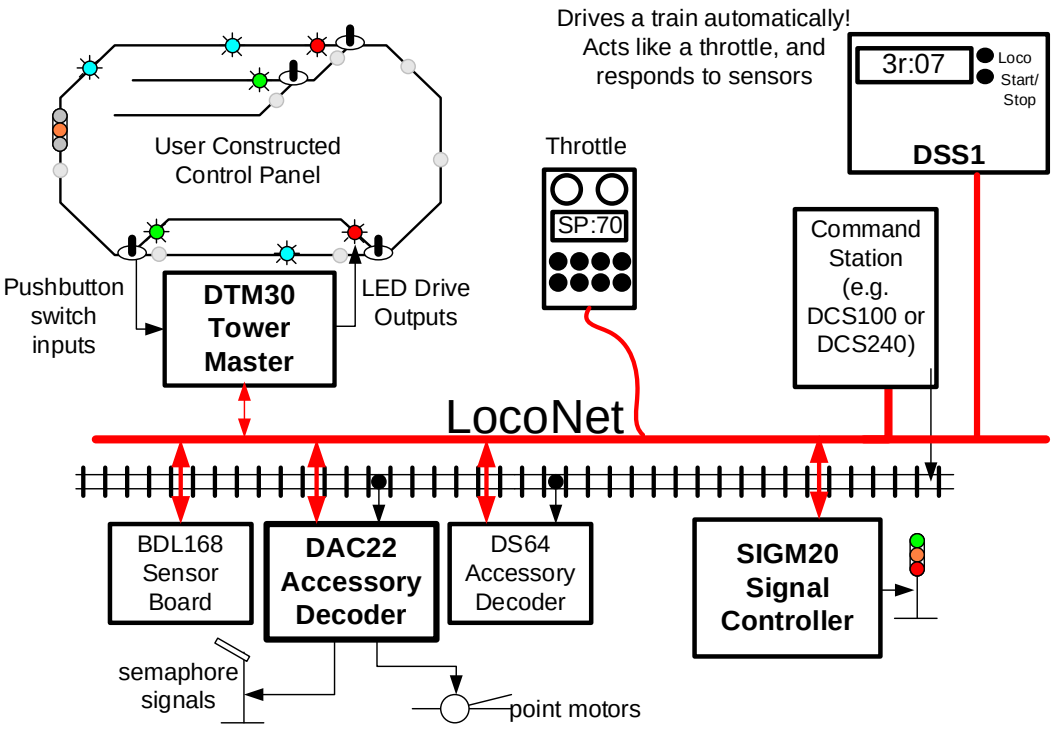
If the solenoids do not "snap" across with a point attached, then the DAC22 is probably not configured correctly or the connections are faulty – see Sec. 3.2 & 4.2.1.

If an output voltage greater than 20 volts is desired (very unlikely!), then an auxiliary supply will be needed. 15-18V AC or a maximum of 25V DC is recommended: this **must** be from a power supply separate from the DCC booster (track) supply.

2.4 Building up to a System

That's just the start; there's a lot more that the DAC22 can do. Read the other sections to find out more.

LocoNet can be used to join the DAC22 to other accessory modules, to create a very powerful system. Signals can be set automatically as trains move; points can be controlled and displayed from panels, rather than throttles; trains can be driven automatically between stations. Automatic control from PC programs is even possible. LocoNet is the backbone that allows all of this to happen.



3. Installation & Connections

The DAC22 unit comprises a single circuit board with dimensions 102mm x 102mm as shown in Figure 3.1. It may be mounted onto spacers or pillars using four screws into the four corner holes: 6BA or M2.5mm screws will be ideal. Alternatively small self tapping screws can be used.

All connections to the DAC22 are made through the following connectors:

Power Connections	SK1
Point Motor Outputs	SK2
Auxiliary Input Connections	SK3
LocoNet Ports	SK4, SK5
Optional DTX8A board Connections	SK6

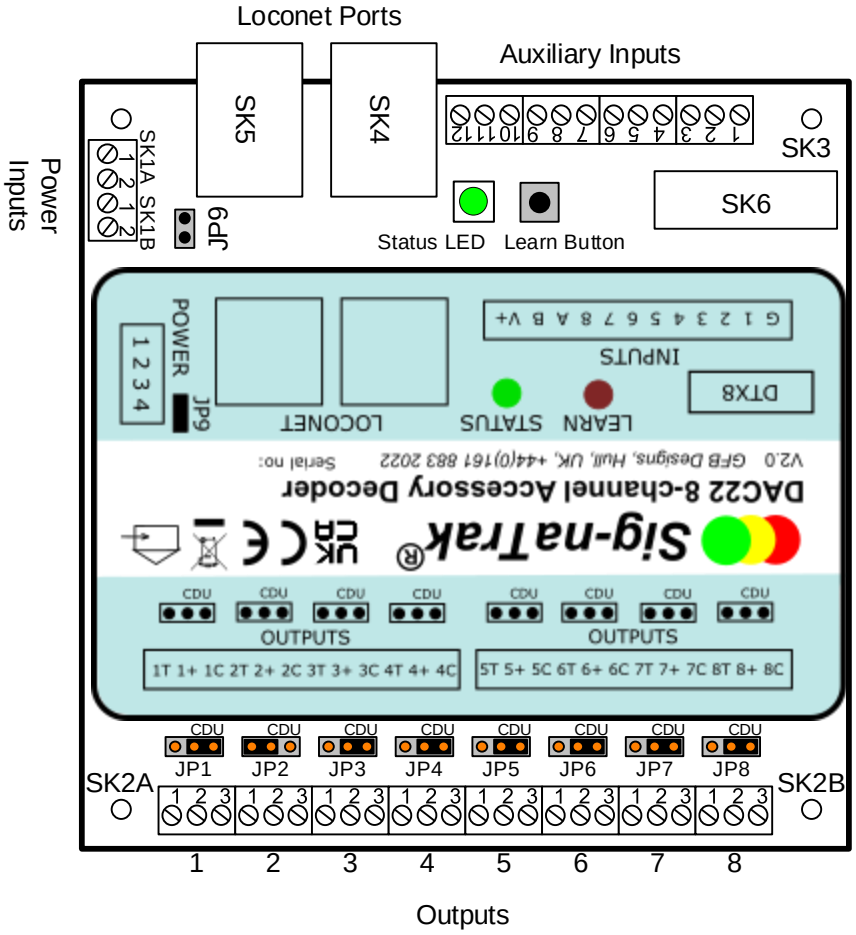
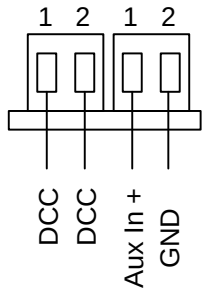


Figure 3.1: DAC22 Interconnections

3.1 SK1: Power Connections

<table>
<tr><td rowspan="5"> 

View looking into SK1A&B from board edge</td><td colspan="3">SK1A&B: Power connections. Two 2-pin screw terminals</td></tr>
<tr><td>Pin</td><td>Function</td><td>Signal Level</td></tr>
<tr><td>pin A1</td><td>DCC track power</td><td rowspan="2">Connects to DCC booster output. 12-20V max. It doesn’t matter which rail signal goes to pin 1 or 2.</td></tr>
<tr><td>pin A2</td><td>DCC track power</td></tr>
<tr><td>pin B1</td><td>Aux_in_+</td><td>Auxiliary power to output stages; NOT USUALLY NEEDED.
• DC: connect +15-25V to pinB1; negative to pin B2. Do not exceed 25V DC.
• AC: connect 15-18v AC to pin B1 & pin B2. Do not exceed 18V AC.</td></tr>
<tr><td></td><td>pin B2</td><td>GND</td><td>Board ground.</td></tr>
</table>

The DAC22 derives its operating power from the DCC track input. In normal operation it consumes up to approximately 100mA from the track feed. The DAC22 contains an in-built capacitor discharge unit (CDU) to provide the high current needed to operate solenoid type point motors (eg Seep, Peco). In addition, an internal voltage booster and regulator maintains approximately 19.5V on the CDU capacitors.

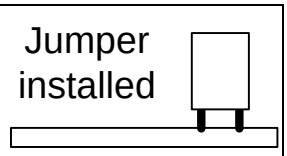
If the DAC22 is driving continuously powered point motors (e.g. Tortoise) then track power is usually sufficient. The DAC22 has significantly improved output drive capability (much lower voltage drops), compared with the original DAC20, and each output can source up to around 100mA (0.1amp) without significant reduction in the output voltage.

An auxiliary AC or DC supply may be connected between Aux in + and GND. This supply feeds the output stages & capacitor discharge unit only. **This power must be derived from a source which is separate from the main DCC power feed to the booster.** Several DAC22s can be powered from one auxiliary supply as long as they are paralleled: **i.e. all SK1Bpin1s connected together, and all SK1Bpin2s connected together.**

The jumper JP9 is provided to connect a power feed from the DCC track input to the output circuits and to the capacitor discharge unit. If the DCC track feed is being used to drive the outputs, then this jumper needs to be connected in normal use and removed when programming CVs via a programming track feed.

For Digitrax command station users, it is also possible to use commands through LocoNet, rather than from the DCC track feed. This means that track shorts will not

affect DAC22 operation. A DC supply of between 12V to18V needs to be connected to SK1A pins 1&2: see Appendix E for details. A "ground" wire should also be added from SK1B pin 2 to the command station ground. JP9 MUST be plugged in.

JP9	2 position jumper	
Position	Function	
installed	normal operation.	
not installed	Remove when using a programming track output	

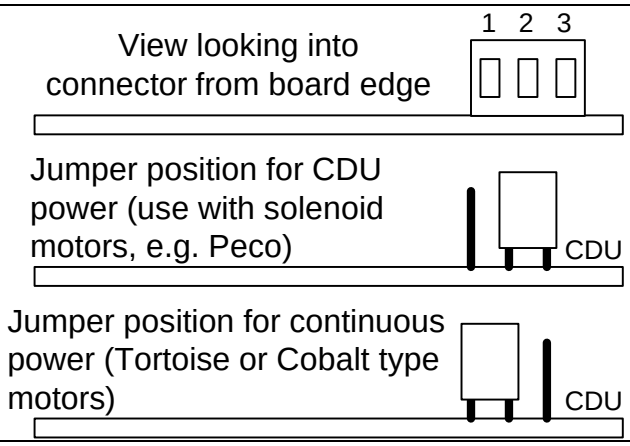
3.2 SK2A&B: Output Connections

These connectors are provided to supply the point motors or other devices themselves and may be connected to drive several different kinds of motor as indicated below. The power feed to each output is individually selectable using jumpers which are described below. Each output has 3 screw terminals, as marked on the coloured label.

3.2.1 Driving Point Motors

The jumpers, JP1 to JP8 select how the outputs are powered. This is either continuous DC (from SK1), or pulses from the capacitor discharge unit (CDU). The CDU should be selected for solenoid type point motors (e.g. Peco); most other devices will need continuous DC power.

Output	Jumper
1	JP1
2	JP2
3	JP3
4	JP4
5	JP5
6	JP6
7	JP7
8	JP8



SK2A&B: Point Motor or other device Output Connectors. Screw terminal		
Pin number	**Function**	**Signal Level**
1	Control signal to drive point motor to THROWN state.	Clamps to Ground to energise motor.
2	+ve Output Power Feed	Selected by JP1-8 (see above)
3	Control signal to drive point motor to CLOSED state.	Clamps to Ground to energise motor.

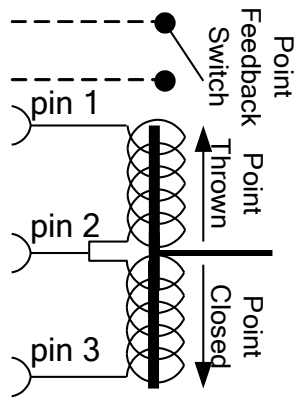
Each output is individually programmable to select the kind of point motor it controls. The common solenoid and stall motor types should be connected as follows:-

Connections for Solenoid type point motors

Solenoid point motors should be connected as shown (right). The Capacitor Discharge Output connects to both solenoid coils; the two other motor control signals connect to the other ends of the solenoid coils.

The appropriate jumper should be in position 2-3 (i.e. the position marked “CDU”).

If feedback of actual point position is required, a switch may be connected as shown. By convention the switch contacts should be closed if the point is closed.

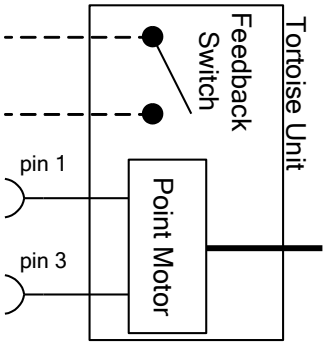


Connections for Slow motion point motors

Slow motion point motors (e.g. Tortoise or Cobalt) should be connected as shown (right). The two motor control signals connect to the motor drive terminals.

The appropriate jumper should be in position 1-2 (i.e. to the left, when viewing the board from the SK2A/B edge).

If feedback of actual point position is required, a switch may be connected as shown. By convention the switch contacts should be closed if the point is closed.



3.2.2 Driving Other Devices

IMPORTANT - because the output driver circuitry has been improved over the original DAC20, the output connections are different if the board is being programmed to control 16 individual outputs.

In this case, all 8 jumpers – JP1 to JP8 – MUST be removed and replaced with a DACA2208 Power Break-out Board (available from Sig-naTrak retailers).

The power feeds to the output devices (up to 16) are taken from the screw terminals on the DACA2208 (labelled “1+ 2+ 3+ 4+ 5+ 6+ 7+ 8+”).

This does not apply if the board is used in its default 8-output mode.

The DACA2208 Power Break-out Board is shown below:-

Connections for Relays and lamps

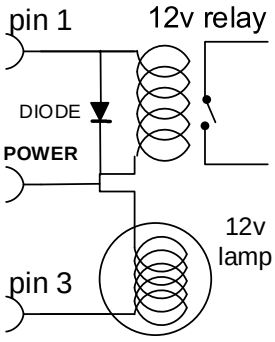
Relays and lamp bulbs can be driven. The two output connections per output cell can be controlled separately to allow up to 16 individually controlled outputs.

Note: "POWER" is the +12v supply to the relay or lamp and this is taken from any of the 8 "+" terminals on the DACA2208 Power Break-out Board.

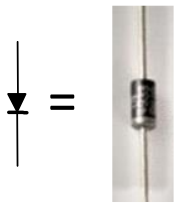
All relays differ in their current drawn. The total current drawn from all DAC22 outputs must not exceed 0.5A. A 400Ω relay will require approx 30mA, so a DAC22 could drive 16 of them; but test the current first!

Note also: Relays require a diode connected across them to supress voltage "spikes" (which occur when the relay is turned off). We recommend type 1N4933 – available from *Sig-naTrak* retailers – as shown opposite. NOTE THE POLARITY!

Filament lamps may be connected in the same way but they do not need a diode across them.



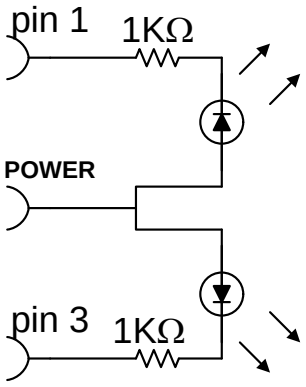
1N4933 Diode



Connections for Light Emitting Diodes (LEDs)

Outputs can also drive light emitting diodes for lighting purposes. Again, each output can be individually controlled. Use LEDs with integral resistors, or add a resistor in series with each LED to limit the current. **Your LEDs will be destroyed if you don't do this!** A 1KΩ resistor is a good starting point.

Note the polarity: "POWER" is the +12v supply to the LED circuit and this is taken from any of the 8 "+" terminals on the DACA2208 Power Break-out Board.



See section 4.2.1 for details of how to program the outputs.

3.3 SK3: Switch / Sensor Feedback Inputs

These pins are provided as external inputs to the board for at least three purposes:

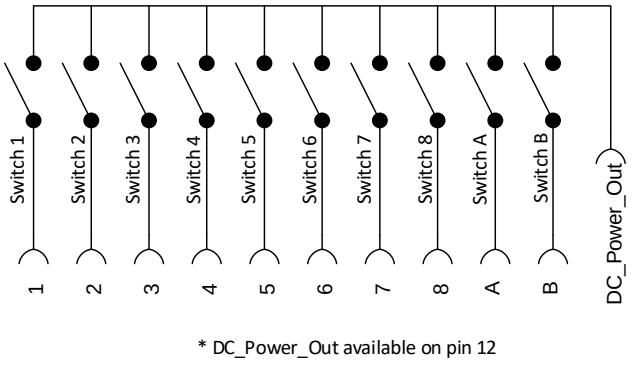
- As external sensor inputs;
- As point position feedback inputs;
- As external command inputs to control the point motors directly.

Signal	**Function**	**Signal Level**
Feedback_In_N	Feedback, switch or sensor input. Connect as shown below.	<6v in: input INACTIVE >6v in: input ACTIVE
(N=1 to 8 for input 1 to 8; N = A or B for auxiliary inputs A, B)		

Pin	**Function**	
G	GROUND	Board ground.
1	Feedback_In_1	Can control or feedback point 1
2	Feedback_In_2	Can control or feedback point 2
3	Feedback_In_3	Can control or feedback point 3
4	Feedback_In_4	Can control or feedback point 4
5	Feedback_In_5	Can control or feedback point 5
6	Feedback_In_6	Can control or feedback point 6
7	Feedback_In_7	Can control or feedback point 7
8	Feedback_In_8	Can control or feedback point 8
A	Feedback_In_A	Input A; can be used as sensor input
B	Feedback_In_B	Input B; can be used as sensor input
V+	DC_Power_Out	+12v Power output. Provides a common feed for switches driving these inputs

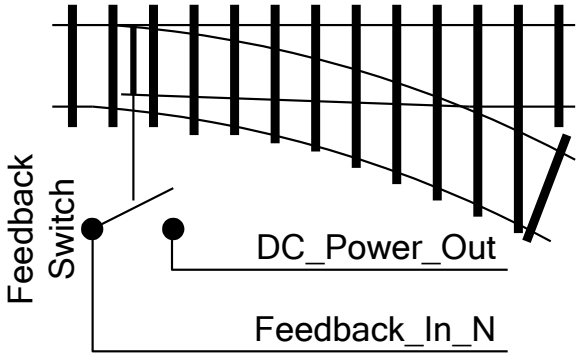
Each input may be individually connected and programmed for a variety of functions. See section 4.3 for programming details. Three possible applications:

Up to 10 manual switches can be connected. These may be used on local panels to control the state of the attached point motors, or may be used to generate system messages.

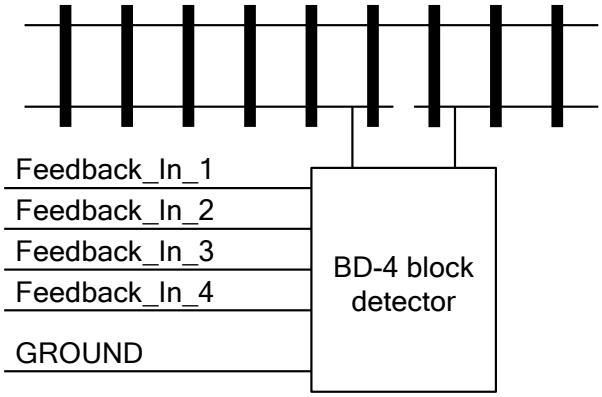


The inputs 1-8 may be used as point position feedback sensors. To achieve this, a switch connected to each point for which feedback is required must be connected as shown.

By convention the switch contacts should be closed if the point is closed.



Block Detectors (such as the Digitrax BD-4) may be connected to feedback inputs as shown.



3.4 SK6: LED Drive Outputs

The DAC22 can connect to an optional add-on DTX8A board to drive LEDs which indicate the position of the 8 outputs. These may be used in constructing local control panels. The DTX8A drives two LEDs for each output: one lit if CLOSED, the other lit if THROWN.

3.5 SK4, SK5: LocoNet Connections

These two identical connectors allow for connection to a LocoNet network using conventional 6 pin RJ12 (US style telephone) connectors. The two connectors are wired in parallel: the LocoNet wiring may be connected to either port, or may be daisy-chained through the DAC22.

These signals are defined in the LocoNet Specification which is available from Digitrax.

3.6 Status LED

A green status LED is provided on the main board itself. This provides indications of the DAC22's function, as follows:

- The LED is lit continuously while the board receives proper DCC signals.
- If the LED is off, the DAC22 is not receiving track power.
- The LED flickers off momentarily when a command for this DAC22 is received. This is useful for checking the address is correctly set.
- If the unit is in LEARN mode, the LED will blink on and off equally once per second. Programming operations can be carried out in LEARN mode.
- If the unit does not have a serial number set, the led will flash briefly on once per second. This will only happen if the serial number is accidentally erased by programming. When the LED blinks in this way, read the serial number from the label on the board and set that switch address to THROWN; the board will then resume normal operation.

4. Configuration

This section describes the functions available from "common" programming of the DAC22. The details of how to achieve the programming itself are in section 6.

4.1 Address Settings

The DAC22 provides the option of two modes for setting the DCC addresses of the board outputs – "Block Addressing" and "Individual Addressing".

4.1.1 Block Addressing

CV5 & CV6 set the base address for the board. This is the DCC accessory (point) number of the first output; the board will respond to 8 or 16 accessory numbers from that number upwards. Almost all users will need to change this setting!

The base address can be set in two ways:

1. Using "LEARN" mode: the address is picked up from a normal point setting command from the throttle after pressing the "LEARN" button. See section 6.1. "LEARN" mode also clears all other CVs and is useful if an error in programming has happened.
2. Using normal programming methods as described in section 6. The values for CV5 and CV6 can be looked up from a table in Appendix B.

4.1.2 Individual Addressing

This mode gives each output a unique address which can be any value from 1 to 2044. In other words, the address of an output is not related to that of any other output and the user is not restricted to a sequential block of 8 (or 16) addresses.

To use Individual Addressing, CVs 5 and 6 must BOTH be programmed to 0 (zero) and the address of each output must then be programmed individually.

LEARN mode can not be used for setting the individual addresses – the normal programming methods, described in section 6, must be used.

Each output address is stored in two CVs from CV431 to CV462. The odd-numbered CV holds the "low-order address" and the even-numbered CV holds the "high-order address". The correct values can be calculated from the table in Appendix B, but the use of our "Locoanalyse" PC application is recommended as a simpler method.

The table below shows the allocation of the address-setting CVs:-

CV	PURPOSE	NOTES
431	Output 1 low-order address	Output 1T, when the DAC22 is programmed to control 16 outputs.
432	Output 1 high-order address	
433	Output 2 low-order address	Output 2T, when the DAC22 is programmed to control 16 outputs.
434	Output 2 high-order address	
435	Output 3 low-order address	Output 3T, when the DAC22 is programmed to control 16 outputs.
436	Output 3 high-order address	
437	Output 4 low-order address	Output 4T, when the DAC22 is programmed to control 16 outputs.
438	Output 4 high-order address	
439	Output 5 low-order address	Output 5T, when the DAC22 is programmed to control 16 outputs.
440	Output 5 high-order address	
441	Output 6 low-order address	Output 6T, when the DAC22 is programmed to control 16 outputs.
442	Output 6 high-order address	
443	Output 7 low-order address	Output 7T, when the DAC22 is programmed to control 16 outputs.
444	Output 7 high-order address	
445	Output 8 low-order address	Output 8T, when the DAC22 is programmed to control 16 outputs.
446	Output 8 high-order address	
447	Output 1C low-order address	These are only used when the DAC22 is programmed to control 16 outputs (e.g. when lamps, relays or LEDs are being driven). (When CV9 = 1)
448	Output 1C high-order address	
449	Output 2C low-order address	
450	Output 2C high-order address	
451	Output 3C low-order address	
452	Output 3C high-order address	
453	Output 4C low-order address	
454	Output 4C high-order address	
455	Output 5C low-order address	
456	Output 5C high-order address	
457	Output 6C low-order address	
458	Output 6C high-order address	
459	Output 7C low-order address	
460	Output 7C high-order address	
461	Output 8C low-order address	
462	Output 8C high-order address	

4.2 Motor/Output Control Settings

4.2.1 Output Types

The DAC22 supports eight output cells, as described in section 3.2. Each cell controls two output signals to an output device. Each cell can be separately configured to operate as one of three basic types:

1. Point type outputs. These control two wires to a point motor; they change between CLOSED and THROWN in response to throttle commands, routes and other programming. They can be programmed to drive continuously power point motors (e.g. Tortoise, Cobalt) or solenoid type motors (e.g. Peco, Seep). For solenoid motors, the "On" time is programmable.
2. "On/Off" type outputs, which control two wires. "On/Off" outputs are put into an "On" state when set to THROWN and an "Off" state when set to "CLOSED". These cells are intended for driving devices such as signals which require a point-type motor to operate them. There is also an option to control pairs of lamps which flash alternately, e.g. for crossing gates.
3. Two separately controlled "On/Off" type outputs, each controlling one wire. These outputs are intended to control lamps, which can be lit constantly or can flash at programmed rates.

Output types 1&2 are controlled by a DCC point (switch) command. When the board is programmed for Block Addressing, the first output's DCC number is the board base address; the next 7 numbers control the next 7 cells. If, however, the board is programmed for Individual Addressing, then each output can have any address within the range 1 to 2044.

For output type 3, the board needs to be set to accept 16 addresses (see section 4.4). In Block Addressing mode, the "1T" output wire is controlled by the board base address and the "1C" output wire is controlled by the base address +8. The addresses of the remaining 7 output pairs follow in sequence – e.g. "2T" has base address +1, "2C" has base address +9 ……. up to "8T" with base address +7 and "8C" with base address +15.

If, however, the board is programmed for Individual Addressing, then each of the 16 output wires can have any address within the range 1 to 2044.

(See sections 4.1.1 and 4.1.2 above.)

Each output type is programmed by the setting of CV11-18. Several settings are available:-

CV	**Setting**	**CV**	**Setting**
11	Output Cell 1 type	15	Output Cell 5 type
12	Output Cell 2 type	16	Output Cell 6 type
13	Output Cell 3 type	17	Output Cell 7 type
14	Output Cell 4 type	18	Output Cell 8 type

CV Value	**O/P Type**	**Meaning**
0	n/a	output not used
1-9	1	Point cell, solenoid motor; operating time 0.125s -> 1.125s
10	1	Point cell, constantly powered motor (e.g. Tortoise)
11-19	2	On/Off cell, solenoid motor; operating time 0.125s -> 1.125s
20	2	On/Off cell, constantly powered motor (e.g. Tortoise)
21-29	2	On/Off cell, two blinking lamps, 9 rates
30	2	On/Off cell, one lamp constantly powered
CV Value	**O/P Type**	**Meaning**
31-39	3	On/Off cell, individually controlled blinking outputs, 9 rates
40	3	On/Off cell, individually controlled outputs continuously lit
41-49	3	On/Off cell, individually controlled blinking outputs, 9 rates, inverted output
50	3	On/Off cell, individually controlled outputs continuously lit, inverted output

- CV value of 3 is recommended for solenoid point motors (e.g. Peco, Seep type)
- CV values 41-50 behave like values 31-40, but the output is inverted. When the output is “on” it is at +12v, and when “off” it is at 0v

4.2.2 Output Message Programming

When an output changes, or during the Digitrax “Interrogation” sequence after track power is applied, each output cell can be programmed to send a LocoNet message. This will inform other devices on LocoNet as to the output’s state. This is useful for software programs such as “Railroad & Co” and JMRI. The messages are controlled by programming CV31-38.

CV	**Setting**	**CV**	**Setting**
31	Output Cell 1 message	35	Output Cell 5 message
32	Output Cell 2 message	36	Output Cell 6 message
33	Output Cell 3 message	37	Output Cell 7 message
34	Output Cell 4 message	38	Output Cell 8 message
Value	**Meaning**		
0	No message for this output cell		
1	“Output State” message generated for this cell		
2	“Turnout Feedback” message generated for this cell		
3	Signal or “SE” message generated. This allows the board to interact with SIGM10/SIGM20 signals.		
4	“Output State” message generated, when interrogated only		
5	“Turnout Feedback” message generated, when interrogated only		

- All messages are defined by the LocoNet specification owned by Digitrax Inc.
- CV=2: the “Turnout Feedback” message is the same as is commonly associated with position feedback microswitches. The DAC22 stores the output state and generates this message with no microswitch needed. Programs like “Railroad &CO” will detect the point position correct when they start up.
- The Signal “SE” message allows a SIGM20 colour light signal to be set according to the position of a semaphore signal driven by the DAC22, or vice versa.

4.2.3 Other Output Functions

The DAC22 includes a "Capacitor Discharge Unit" (CDU) to power solenoid type point motors. When the output control CVs (CV11-18) are set to a solenoid type (CV=1-9, or CV=11-19) a delay time is included by the unit between output changes to allow the CDU to recharge itself from the board's internal voltage booster. Typically this takes between 0.8 and 1 second.

The delay time is set by CV19. The value programmed into this CV sets a delay, in units of 0.1 seconds. It can be programmed with values 1-32, implying delays between 0.1 and 3.2 seconds.

The DAC22 can be used to control lighting circuits. The lamps can be filament bulbs, or can be light emitting diodes (LEDs). LEDs, in particular, turn on or off instantly. The DAC22 has a feature that will simulate the gradual brightening and darkening of "real" lights. This is turned on or off for each output using CV20.

To enable brightness simulation, the values from the following table need to be added together then programmed into CV20 to turn "on" the effect for selected outputs.

CV20 (Brightness Simulation)			
Output	**Add value**	**Output**	**Add value**
1	1	5	16
2	2	6	32
3	4	7	64
4	8	8	128

Example: to enable brightness simulation for output 3,5 & 6: program CV20 with (4 + 16 + 32) = 52

4.3 Input Control Settings

The DAC22 has 10 inputs that can be programmed to achieve several purposes. The inputs can control output cells, or trigger routes. They can generate LocoNet messages. Inputs are wired as shown in section 3.3.

4.3.1 Input types

Each input cell is separately programmed to control its operation. This covers when it takes an action, and what action is triggered. Inputs can generate an action:

- When changing from 0v to +12v: this is appropriate if a pushbutton is used;
- When changing from 0v to +12v, and +12v to 0v: this setting is appropriate if a toggle switch or feedback device (microswitch, occupancy detector) is used.

Input operations are controlled by CV41-50, as shown in the table below:-

CV	Setting	CV	Setting
41	Input Cell 1 setting	46	Input Cell 6 setting
42	Input Cell 2 setting	47	Input Cell 7 setting
43	Input Cell 3 setting	48	Input Cell 8 setting
44	Input Cell 4 setting	49	Input Cell A setting
45	Input Cell 5 setting	50	Input Cell B setting
Value	**Meaning**		
0	Cell not used.		
1	Pushbutton type cell, doesn't affect output cell		
2	Toggle switch type cell, doesn't affect output cell		
3	Pushbutton type cell, change output cell state		
4	Toggle switch type cell, output cell follows input setting		
5	Pushbutton type cell, trigger a local route		

4.3.2 Input Message Settings

When an input changes, or during the Digitrax "Interrogation" sequence after track power is applied, each input cell can be programmed to send a LocoNet message. This will inform other devices on LocoNet as to the input's state. This is useful for software programs such as "Railroad & Co" and JMRI. The messages are controlled by programming CV51-60.

CV	Setting	CV	Setting
51	Input Cell 1 message	56	Input Cell 6 message
52	Input Cell 2 message	57	Input Cell 7 message
53	Input Cell 3 message	58	Input Cell 8 message
54	Input Cell 4 message	59	Input Cell A message
55	Input Cell 5 message	60	Input Cell B message
Value	**Meaning**		
0	No message for this input cell		
1	"sensor" message generated for this cell. These are used to indicate whether track is occupied. May be used with occupancy detectors, e.g. IRDOT.		
2	"Turnout Feedback" message generated for this cell. These are used to indicate whether a point is thrown or closed, using a feedback microswitch on the tiebar. The convention is the switch contact is closed when the point is CLOSED.		
3	Generate an "emergency stop" message to halt all trains.		
4	Generate a "track power off" message.		

When Sensor messages are used, the unit generates messages with sensor numbers 1-10 corresponding to inputs 1-8, A and B respectively. The sensor message also needs a board number (1-256) that the sensor message originated from. The value programmed into CV39 provides this value.

4.4 General Unit Settings

CV9 controls whether the board occupies 8 or 16 address locations. This needs to be set depending whether the ability to separate each output cell into two independently controlled halves is needed. If the board is purely to control point motors, set for 8 address locations; if some outputs are being used to control relays or lamps individually, set to 16 locations.

- For 8 locations: set CV9 to 0
- For 16 locations: set CV9 to 1

CV10 controls whether the unit accepts its point commands from the DCC rail signal or from LocoNet. This may be useful in certain specialised applications where a non-Digitrax command station is being used; it may allow a DTM30 to control points while a non-LocoNet command station controls the trains, for example:-

- For DCC control (normal): set CV10 to 0
- For LocoNet control (specialised applications only): set CV10 to 1 (see Appendix E).

4.5 Output State Memory CV

(Access to this CV is not normally required)

The DAC22 "remembers" the state of its outputs in a non-volatile memory, which remembers its state even while power is turned off Every time an output is changed, the new output state is recorded to this memory.

CV30 sets the address used within the non-volatile memory to record this output state information. It specifies an address in the range 0 – 19 and may be set to any value in that range.

In normal use this CV need never be touched. Because of a fundamental "lifetime" restriction with non-volatile memories, each individual address in the memory may eventually fail to store data correctly. In a DAC22 used daily this may occur once every few years. If this happens, by reprogramming this CV a different address may be selected to allow output state to be recorded.

4.6 Output Following Control CVs

Output following is a facility which allows one output to "follow" the state of another. This is useful where points need to be changed together. An example of this is at a crossing between two lines: the points on each line would normally be operated together. With point following, one point can be controlled by the user and the second will automatically follow its state.

Output following is controlled using CVs 21-28. Each CV controls the feature for one individual output.

Output Following Configuration Variables	
CV21: Output 1 following	CV25: Output 5 following
CV22: Output 2 following	CV26: Output 6 following
CV23: Output 3 following	CV27: Output 7 following
CV24: Output 4 following	CV28: Output 8 following
Value	**Effect**
0	No output following (normal operation)
1-8	This output follows the output number entered
Other values	reserved (do not use)

Table 4.1: Output Following CV Values

Example: Output 3 is to follow the state of output 4. To achieve this CV23 is programmed with a value 4. If output 4 is activated either from a local switch input or from a DCC command, output 3 will also be driven to the same state.

4.7 Local Route CVs

Local routes provide a facility for users to control more than one point motor at a time from a single action. This is useful, for example, where a number of points need to be set to define the route to be taken by a train. This may be through a complex junction (e.g. in the throat to a large station) or a storage area (e.g. a set of "hidden" staging yards).

8 Local routes are provided. Each allows up to 10 points to be set to a predefined state. Note that a local route *sets* the points to a defined state. There is no equivalent mechanism (other than using a different local route) to "*unset*" them. For operations requiring both actions, consider the "output following" mechanism.

Route	**CV Range**	**Route**	**CV Range**
1	63-74	5	111-122
2	75-86	6	123-134
3	87-98	7	135-146
4	99-110	8	147-158

Table 4.2: Local Route CVs

The route definition consists of a list of values that describe the settings required. Any unused CVs, at the end of the list, must be set to 0. The values used may be as follows:

- Value=0: end of route definition
- Value =1-8: set cell 1-8 (on this board) to CLOSED
- Value =11-18: set cell 1-8 (on this board) to THROWN
- It is also possible to control points that are driven by another accessory decoder. To do so requires two entries in the list:
    - Look up X and Y values for the point address from the table in Appendix B
    - If the point is to be CLOSED: program the first entry to be (Y+100)
    - If the point is to be THROWN: program the first entry to be (Y+200)
    - In either case, program the X value as the second entry.

Routes may be entered very easily into the PC program “Locoanalyse” and these values will be calculated automatically!

Local routes are normally triggered using a pushbutton input. They may also be triggered by a DCC point setting from a throttle. To use this feature, CV61 & CV62 need to be set to correspond to the first location to be used to control routes; routes 2-8 are then triggered by the subsequent 7 addresses. It may be a good idea to choose location set apart from normal point numbers (e.g. 501-508).

5. Advanced Programming

This section describes the "advanced" programming that is possible. This may be skipped if standard accessory operations are sufficient. There are a range of uses for which more complicated programming can be defined. These functions could be used, for example, to carry out operations such as the following:

- Operate a relay when a track sensor is occupied
- Operate a relay when a track sensor is occupied, but only when a point is in a particular position
- Control a signal so that the signal aspect is "danger" when a track sensor reports that the track is occupied, or when a point is set against the signal.

These functions go beyond what a simple accessory can do; they require that some logic is provided to take the settings of the various devices on the railway, and work out how to control the output. This manual describes how to use these functions.

5.1 What is a Condition?

These functions are controlled by ***conditions***. Conditions are simply lists of settings for points, sensors or signals that will cause something to happen. Users familiar with the SIGM10/SIGM20 signal controllers will already have used similar groups of conditions to control signal aspect.

We could control the output cell with a statement such as:

"if X happens, make the point go to CLOSED"

We want to be able to control the output cells according to a list of several things so we might have:

"make the point go to CLOSED if A happens, or if B happens, or if C happens"

This is what conditions are. They are simply a list of events where if any of them happen, the specified action for the signal will occur.

Conditions can be programmed for the following events:

- A point being CLOSED or THROWN;
- A signal being RED or NOT RED;
- A sensor being OCCUPIED or NOT OCCUPIED;

By creating combinations of these conditions, very complex point control; logic can be constructed; for example:

"make this point go to THROWN if point 45 is thrown,
or if sensor 5 on board 3 is occupied,
or if signal 29 is red"

It is worth noting that accessory numbers (e.g. point numbers) are not restricted to those assigned to real points and accessories. If a user wants to make something happen under user control, it can be achieved by setting unused accessory numbers to thrown or closed. For example, if point 503 does not exist on the layout, then effects in a DAC22 can be invoked by setting 503 to closed or thrown using a throttle: this does not affect the operating track because there is no point numbered 503.

So far, conditions groups have been described that are dependent on one or more individual events happening, e.g. “go to CLOSED if: point 75 is closed, or point 67 is thrown”. It is also possible to define conditions where two or more events have to happen at the same time, e.g. “go to CLOSED if “point 75 is closed AND point 67 is thrown”. In this case the point will only go change to closed when **both** points are set as indicated.

Conditions such as this can be incorporated into the condition groups freely and can be intermingled with other conditions to create complex logic.

5.2 Advanced Output Control Operation

The three basic types of output cell in the DAC22 are described in section 4.2.1. For simple output control functions, the cells appear very similar. However when more complicated programming is used, there are some distinctions between the two types. These are described in the next section, together with the programming that can be used.

5.2.1 Advanced Control for Point Cells

What a point cell is typically used to control a point motor. It is driven to CLOSED or THROWN, and there is no “bias” towards either side. A point cell can be set:

- By an external command (e.g. by a throttle);
- By an input pushbutton;
- By a condition group. This allows other actions on the railway to control it.

Three groups of conditions are available for each output cell:

“Closed” conditions	7 conditions per cell can cause a point cell to change to “Closed”. If any of these conditions is active, then the point will change as long as there are no “Thrown” or “Freeze” conditions active.
“Thrown” conditions	7 conditions per cell can cause a point cell to change to “Thrown”. If any of these conditions is active, then the point will change as long as there are no “Closed” or “Freeze” conditions active.
“Freeze” conditions	2 conditions per cell can disable other conditions from changing the cell. These can be used to stop a point changing if a train is on its track, or prevent “illegal” states in 3 way points for example.

Note that conditions must be unambiguous. If both “Closed” and “Thrown” conditions are active, the cell will not change.

5.2.2 Advanced Control for On/Off Cells

An “on/off” cell differs from a point state in that it has two states, but the two states aren’t equally biased. The cell will go to its “on” state if an external command is set to “THROWN”, or if its condition groups are active. That’s like having two switches for a light: either one can turn the light on, but both must be at “off” before the cell will go to “off”.

Type 2 “On/Off” Cells with two output wires

For cells that have two output wires (e.g. still controlling a point motor) there are 3 conditions groups that control the cell.

“On” conditions A	7 conditions per cell cause an “on/off” cell to become “on”. The cell will change to “on” as long as there are no “Freeze” conditions active.
“On” conditions B	7 conditions per cell cause an “on/off” cell to become “on”. The cell will change to “on” as long as there are no “Freeze” conditions active.
“Freeze” conditions	2 conditions per cell can disable other conditions from changing the cell.

The “A” and “B” conditions groups both control the cell, and there is no difference between them.

Type 3, Individually controlled “on/off” cells

Main “On” conditions	7 conditions per cell cause the “main” output of an individually controlled “on/off” cell to become “on”. The cell will change to “on” as long as there are no “Freeze” conditions active.
Aux “On” conditions	7 conditions per cell cause the “aux” output of an individually controlled “on/off” cell to become “on”. The cell will change to “on” as long as there are no “Freeze” conditions active.
“Freeze” conditions	2 conditions per cell can disable other conditions from changing the cell.

5.2.3 Condition Delay for On/Off cells

Sometimes, it is appropriate that when a condition happens, the state should stay in that position for a period afterwards. This might be used to make a signal go to “danger” and stay at “danger” for a number of seconds after the train passes, regardless of what position the track sensors are in.

"On/Off" cells have a time delay value associated with them. When an output turns "on" as a result of a condition group, the conditions will hold their current setting for the programmed time period before they are re-evaluated. This has the effect of making the output stay "on" for at least as long as that time period.

The delay value is simply programmed as the number of seconds before which the conditions will be re-evaluated. The value can be between 0 & 255 seconds.

CV	**Delay value**	**CV**	**Delay value**
CV 161	Delay for cell 1	CV 165	Delay for cell 5
CV 162	Delay for cell 2	CV 166	Delay for cell 6
CV 163	Delay for cell 3	CV 167	Delay for cell 7
CV 164	Delay for cell 4	CV 168	Delay for cell 8

5.2.4 Advanced Route Operation

Each route can be triggered by a group of conditions. This could be used to allow a route through trackwork to be controlled by the arrival of a train at an input section of track.

Each route is allocated a set of 5 conditions at the following locations:

CV location	**Route**	**CV location**	**Route**
CV 381-385	Trigger route 1	CV 401-405	Trigger route 5
CV 386-390	Trigger route 2	CV 406-410	Trigger route 6
CV 391-395	Trigger route 3	CV 411-415	Trigger route 7
CV 396-400	Trigger route 4	CV 416-420	Trigger route 8

5.2.5 Additional Output Delay

Normally, cell output operate independently of each other. Sometimes however it is necessary for one output not to change until another has finished changing. This can occur on three way points, where there is otherwise a "race" between the two tiebars.

CV421-428 allow a delay to be programmed. When zero the outputs operate normally. When a delay is set, no other output will be allowed to change until this delay has occurred. It will force any other output to wait. The delay value is in units of 0.1 seconds: a value of 25 corresponds to 2.5 seconds.

CV location	**Route**	**CV location**	**Route**
CV 421	Output 1 delay	CV 425	Output 5 delay
CV 422	Output 2 delay	CV 426	Output 6 delay
CV 423	Output 3 delay	CV 427	Output 7 delay
CV 424	Output 4 delay	CV 428	Output 8 delay

6. Programming the DAC22

Like any DCC decoder, the DAC22 needs to have some CVs set to control its operation. There are three methods to program the CVs:

- Using “LEARN” mode
- Using a PC
- Using a programming track

6.1 Learn Mode

The simplest method of programming is “LEARN” mode. This allows the unit to be programmed “live” while connected to the DCC rails in the normal way.

“LEARN” mode is entered and exited by pressing the “LEARN” button for more than one second. When it is released, the DAC22 will change mode. The green STATUS LED will blink on and off approximately once per second while in LEARN mode.

When in “LEARN” mode, the DAC22 responds to both switch (point) commands and “ops” mode (on the rails) programming commands as follows:

Any switch command is used to set the base address and clear **all** of the CVs. If the switch is set to “Thrown”, the outputs are configured for solenoid point motors (e.g. Peco). If the switch is set to “Closed”, the outputs are configured for continuously powered point motors (e.g. Tortoise). **We recommend using this method if difficulties are being encountered.**

Any “ops” mode programming commands (the same as used to program a locomotive while on the rails) will program CVs on the DAC22 while in this mode. To use this method: use a throttle to select a locomotive number **that does not exist on the railway**, and enter “ops mode programming” mode. Use the throttle to select a CV number and a new value, and the new value will be programmed. The green LED on the DAC22 will blink. Be careful using this mode: the potential to reprogram a loco inadvertently by selecting its loco number is great!

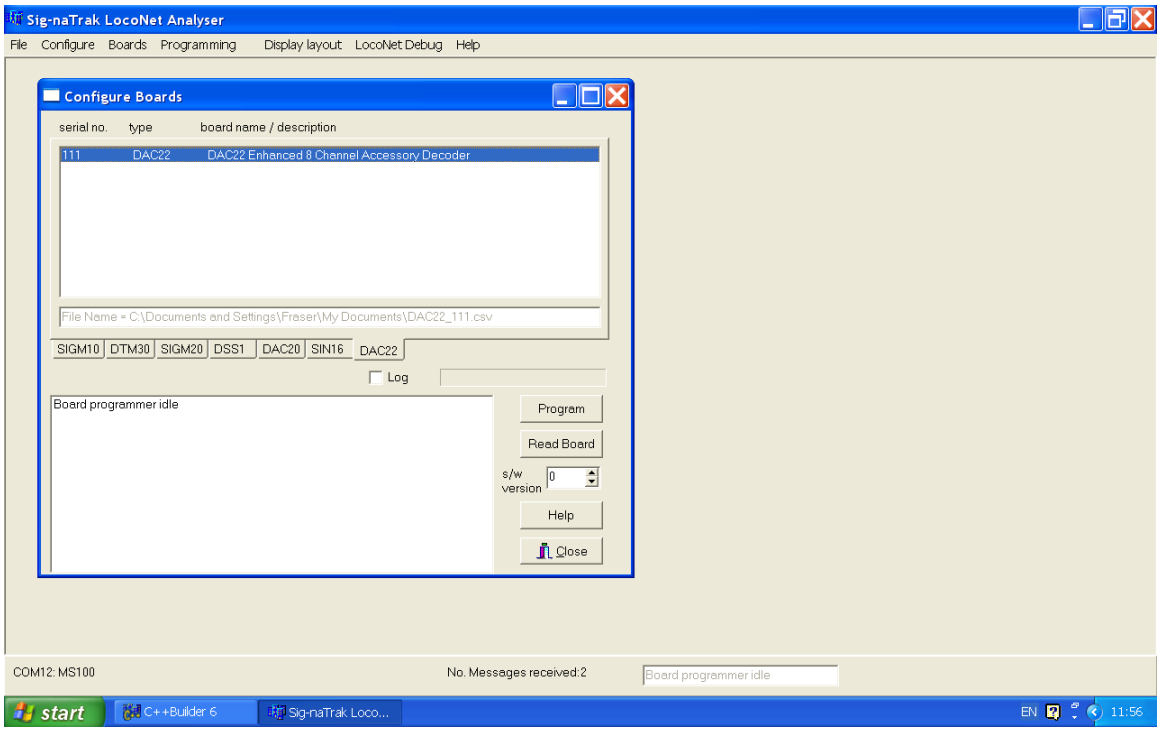
6.2 PC Programming – Requires Locoanalyse V3.1.0.0 (or later)

If you have a PC with an interface to LocoNet (for example a Digitrax PR3, or a LocoBuffer) then all of the settings can be programmed using the PC. We provide a free program "Locoanalyse", which can program the DAC22 and other LocoNet-compatible boards in the *Sig-naTrak* range. Their settings can be stored and printed for later reference. The program can be downloaded free of charge from our web site.

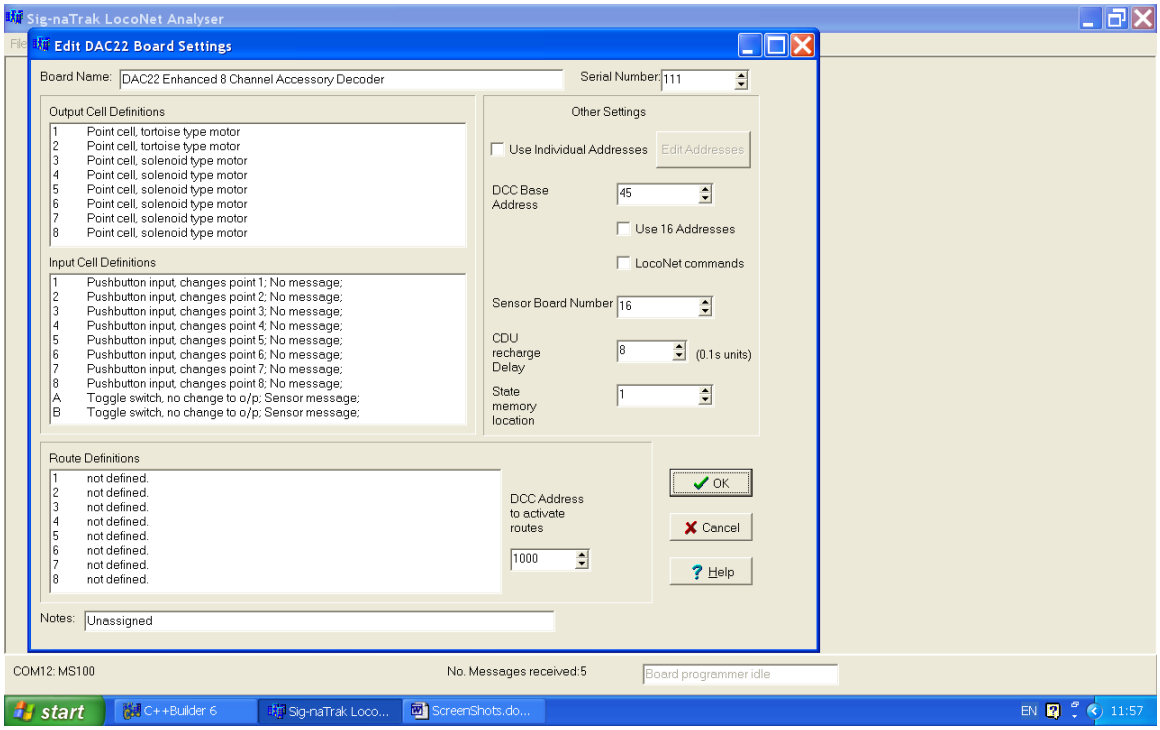
There is a "quick start guide" on the website covering installing and setting up the "Locoanalyse" program.

To use Locoanalyse to program a DAC22, follow these basic steps. Where details are needed about what to enter into any of the boxes, consult the "help" file installed with the program.

Run the program by using the Windows "Start" menu. Select "Configure boards" from the "Configure" menu. Select the "DAC22" tab on the window. Then select "Add new board" from the "boards" menu. An entry for a new DAC22 is now visible.

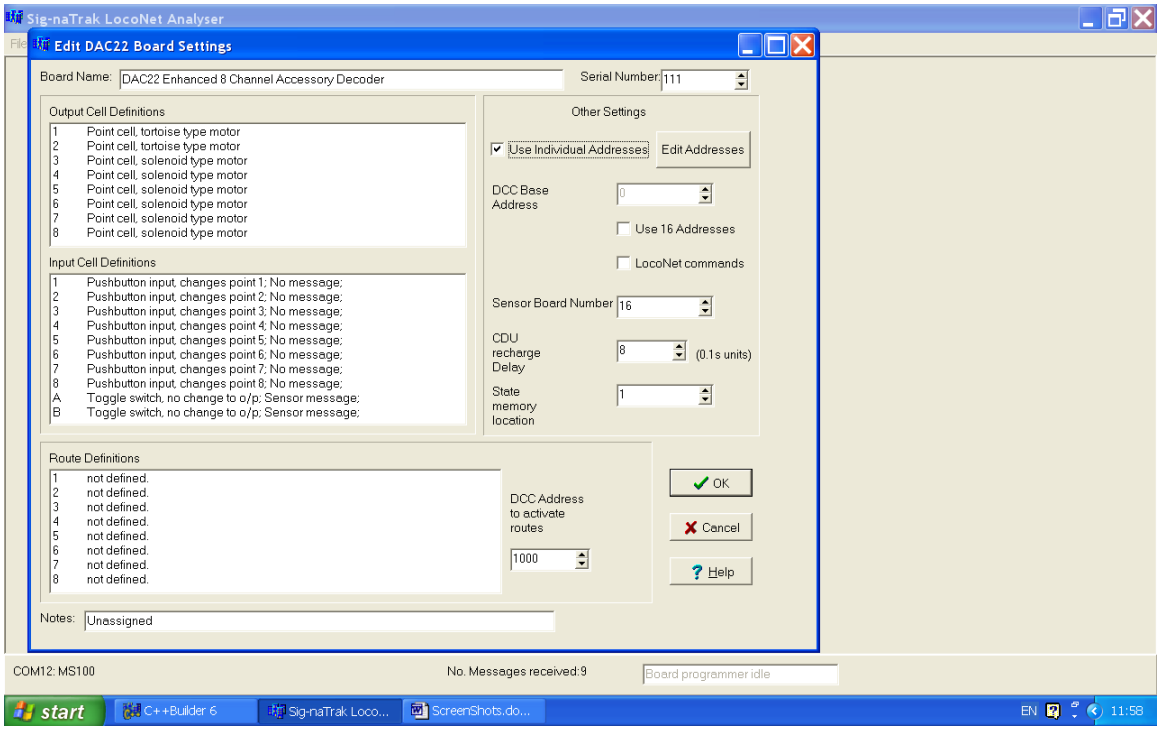


Select that entry in the window, then select "Edit board" from the "Boards" menu or double-click on the selection. The DAC22 editor window opens; this is where the settings are programmed.

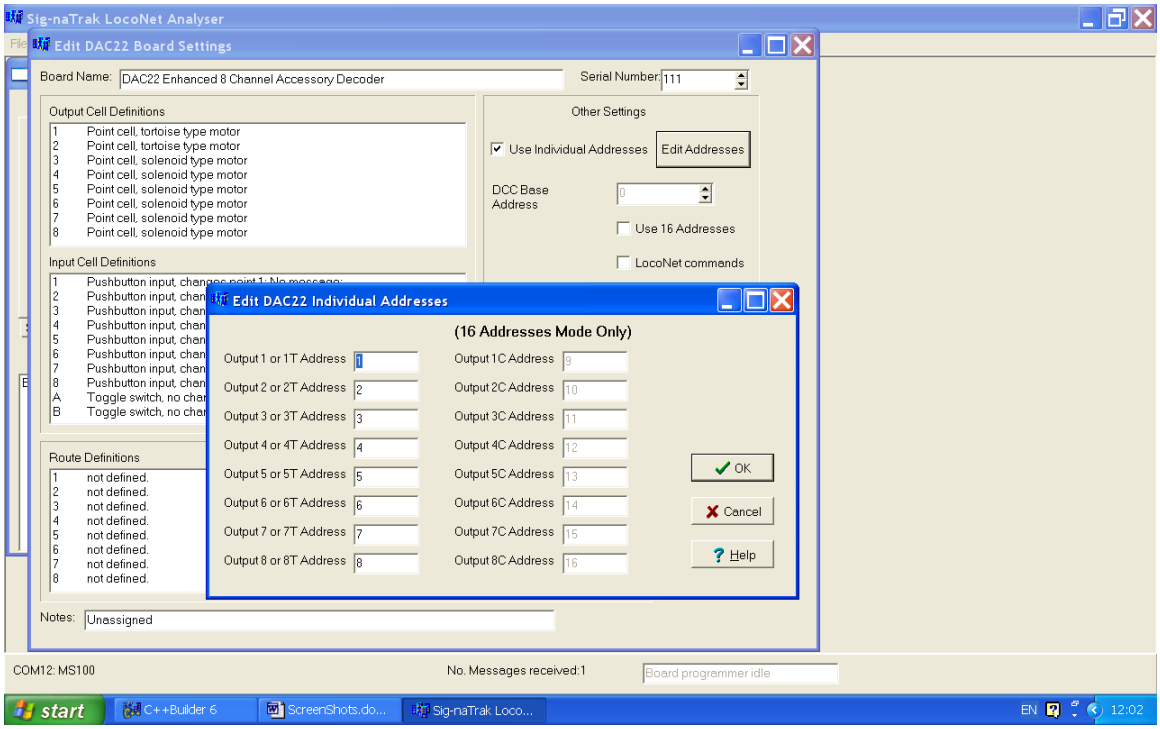


The settings for output cells, input cells and routes are opened for edit by double clicking on the entry in the lists to the left-hand side of the DAC22 window. The general settings for the board are on the right-hand side of the window. These include the output addresses, which will almost always need to be changed.

Normally, the “Edit DAC22 Board Settings” window shows the “Use Individual Addresses” box as unchecked, with the “Edit Addresses” button greyed out. If the “Use Individual Addresses” box is checked, then the “Edit Addresses” button will become enabled, as shown below:-



From here, clicking on the “Edit Addresses” button will then display the “Edit DAC22 Individual Addresses” window, as shown below, allowing each of the 8 (or 16) outputs to have their own specific address assigned.



Click on “OK” to close the “Edit DAC22 Individual Addresses” window, if used, click on “OK” on the “Edit DAC22 Board Settings” window; then click “Save” on the “File” menu. The program will ask for a filename; choose an appropriate place to save your board’s settings.

Use the “Edit DAC22 Board Settings” window to change the settings; remember to save changes often. Use the “Help” file where needed.

Once the settings are ready, close the “Edit DAC22 Board Settings” window and save the settings. Then select your DAC22 board in the “Configure Board” window and press the “Program” button. The settings will be sent through LocoNet, and your board will be ready for use in approximately 10 seconds.

6.3 Using a Programming Track

The DAC22 can be programmed using a programming track. This uses the programming facilities offered by all DCC command stations. Each system is different: the specific use of any one command station is not covered by this manual.

The DAC22 supports **Paged Mode** and **Direct Mode p**rogramming. To prepare the DAC22 for programming, several steps need to be taken:

- Remove programming jumper JP9;
- Connect the DAC22 to the command station's programming track output;
- Select the command station to generate Paged mode or Direct mode programming commands.
- CVs can be programmed and read back in just the same way as with locomotive decoders.

Most DCC command stations will use normal decimal numbers to describe the values to be programmed into each register, and decimal values to describe register addresses. The DAC22 manual follows this convention.

Some older DCC systems require hexadecimal (base 16) values to be used: for example the Digitrax DT100 handheld throttle uses this system. This simply requires that the numbers be converted from decimal to hexadecimal, so a decimal to hexadecimal conversion chart is provided in Appendix C for this purpose. For example, a value of 154 is equivalent to the hexadecimal number 9A.

Appendix A DAC22 Configuration Variables (CVs)

CV	Function	Meaning	Type	Initial Value
1	Memory size	No user meaning!	read-only	1
2	Software Version number	Software version installed	read-only	4.4
3,4	Serial Number	Board serial number, used for PC programming.	normal	marked on label
5,6	Base address	start accessory address used for normal operations		CV5=45; CV6=0
7	Version ID	As CV 2	read-only	4.4
8	Manufacturer ID	DCC ID for GFB Designs	read-only	46
9	Address size	See section 4.4	normal	0
10	Bus select	See section 4.4	normal	0
11-18	Output settings	See section 4.2.1	normal	3
19	CDU timing	See section 4.2.3	normal	8
20	Brightness control	See section 4.2.3	normal	0
21-28	Output following	See section 4.6	normal	0
29	Decoder config	(has no effect)	read-only	128
30	Output memory	See section 4.5	normal	0
31-38	Output message settings	See section 4.2.2	normal	4
39	Sensor board number	See section 4.3.2	normal	16
41-50	Input settings	See section 4.3.1	normal	1
51-60	Input message settings	See section 4.3.2	normal	51-58=0 59-60=1
61-62	Route base address	See section 4.7	normal	61=232; 62=3
63-158	Local routes	See section 4.7	normal	0
161-168	Output Condition Delay	See section 5.2.3	normal	0
171-250	External device table	See section 5.1	normal	0
251-378	Cell conditions	See sections 5.2.1, 5.2.2	normal	0
381-420	Route conditions	See section 5.2.4	normal	0
421-428	Additional output delay	See section 5.2.5	normal	0
431-462	Individual output addresses	See Section 4.1.2	normal	1 to 16

Appendix B Decoder Address Chart

This chart tabulates the decoder addresses obtained from different settings of CVs5,6 and 431-462 **for non ZTC systems**. It is also used for setting route and individual output addresses. Although this table shows steps of 10, note that all intermediate addresses can be used.

Every address in the DAC22 occupies TWO CVs, an odd-numbered CV and an even-numbered CV. For example: the decoder base address is stored in CVs 5 and 6; the individual address for output 1 is stored in CVs 431 and 432.

Examples of addresses:

- Address = 71: set even-numbered CV to 0, odd-numbered CV to 71
- Address = 645: set even-numbered CV to 2, odd-numbered CV to133

Odd CV (X value)	**Even-numbered CV (Y 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.

Appendix C Hexadecimal Conversion Chart

If your DCC system needs “hexadecimal” values for programming, use this chart to convert from decimal values to hex values. (This usually applies to old systems only)

dec	hex	dec	hex	dec	hex	dec	hex	dec	hex	dec	hex
0	00	44	2C	88	58	132	84	176	B0	220	DC
1	01	45	2D	89	59	133	85	177	B1	221	DD
2	02	46	2E	90	5A	134	86	178	B2	222	DE
3	03	47	2F	91	5B	135	87	179	B3	223	DF
4	04	48	30	92	5C	136	88	180	B4	224	E0
5	05	49	31	93	5D	137	89	181	B5	225	E1
6	06	50	32	94	5E	138	8A	182	B6	226	E2
7	07	51	33	95	5F	139	8B	183	B7	227	E3
8	08	52	34	96	60	140	8C	184	B8	228	E4
9	09	53	35	97	61	141	8D	185	B9	229	E5
10	0A	54	36	98	62	142	8E	186	BA	230	E6
11	0B	55	37	99	63	143	8F	187	BB	231	E7
12	0C	56	38	100	64	144	90	188	BC	232	E8
13	0D	57	39	101	65	145	91	189	BD	233	E9
14	0E	58	3A	102	66	146	92	190	BE	234	EA
15	0F	59	3B	103	67	147	93	191	BF	235	EB
16	10	60	3C	104	68	148	94	192	C0	236	EC
17	11	61	3D	105	69	149	95	193	C1	237	ED
18	12	62	3E	106	6A	150	96	194	C2	238	EE
19	13	63	3F	107	6B	151	97	195	C3	239	EF
20	14	64	40	108	6C	152	98	196	C4	240	F0
21	15	65	41	109	6D	153	99	197	C5	241	F1
22	16	66	42	110	6E	154	9A	198	C6	242	F2
23	17	67	43	111	6F	155	9B	199	C7	243	F3
24	18	68	44	112	70	156	9C	200	C8	244	F4
25	19	69	45	113	71	157	9D	201	C9	245	F5
26	1A	70	46	114	72	158	9E	202	CA	246	F6
27	1B	71	47	115	73	159	9F	203	CB	247	F7
28	1C	72	48	116	74	160	A0	204	CC	248	F8
29	1D	73	49	117	75	161	A1	205	CD	249	F9
30	1E	74	4A	118	76	162	A2	206	CE	250	FA
31	1F	75	4B	119	77	163	A3	207	CF	251	FB
32	20	76	4C	120	78	164	A4	208	D0	252	FC
33	21	77	4D	121	79	165	A5	209	D1	253	FD
34	22	78	4E	122	7A	166	A6	210	D2	254	FE
35	23	79	4F	123	7B	167	A7	211	D3	255	FF
36	24	80	50	124	7C	168	A8	212	D4		
37	25	81	51	125	7D	169	A9	213	D5		
38	26	82	52	126	7E	170	AA	214	D6		
39	27	83	53	127	7F	171	AB	215	D7		
40	28	84	54	128	80	172	AC	216	D8		
41	29	85	55	129	81	173	AD	217	D9		
42	2A	86	56	130	82	174	AE	218	DA		
43	2B	87	57	131	83	175	AF	219	DB		

Appendix D LocoNet Interrogation Sequence

The DAC22 participates in the LocoNet accessory and sensor interrogation sequence which is controlled by some command stations (e.g. DCS100 & DCS240). This sequence is invoked after power is applied to the layout by the command station sending accessory commands to special addresses. The DAC22 responds by generating LocoNet output state & sensor feedback messages according to the settings of each point output and the status of each input (which may be driven by a LocoNet sensor).

This behaviour allows the LocoNet system to "discover" the state of all attached points, accessory devices and input sensors after power is applied. No further messages are generated until a point output or a sensor input changes state.

Appendix E Operating without Track Power

For railways using Digitrax command stations only, the DAC22 can be operated from an auxiliary supply with no track power connection. This has the advantage that track power shorts do not cause the DAC22 to lose power. This means that if a train drives onto an incorrectly set point, causing a track short, the point can be changed.

To use this configuration:-

1. Connect an auxiliary supply to SK1A pins 1&2 (i.e. where "track power" would normally be connected).
2. Add a "ground" wire from SK1B pin 2 to the command station ground.
3. Make sure JP9 is installed.
4. Program CV10=1 (this makes the DAC22 respond to LocoNet commands, not to DCC commands).

Note that the DAC22 will not respond to routes programmed into a DCS100 command station in this configuration. Routes from DAC22, and DTM30 Tower Master units, are fine.

The Green STATUS LED will NOT be lit in this mode. However it will light momentarily if a command for the board is detected.