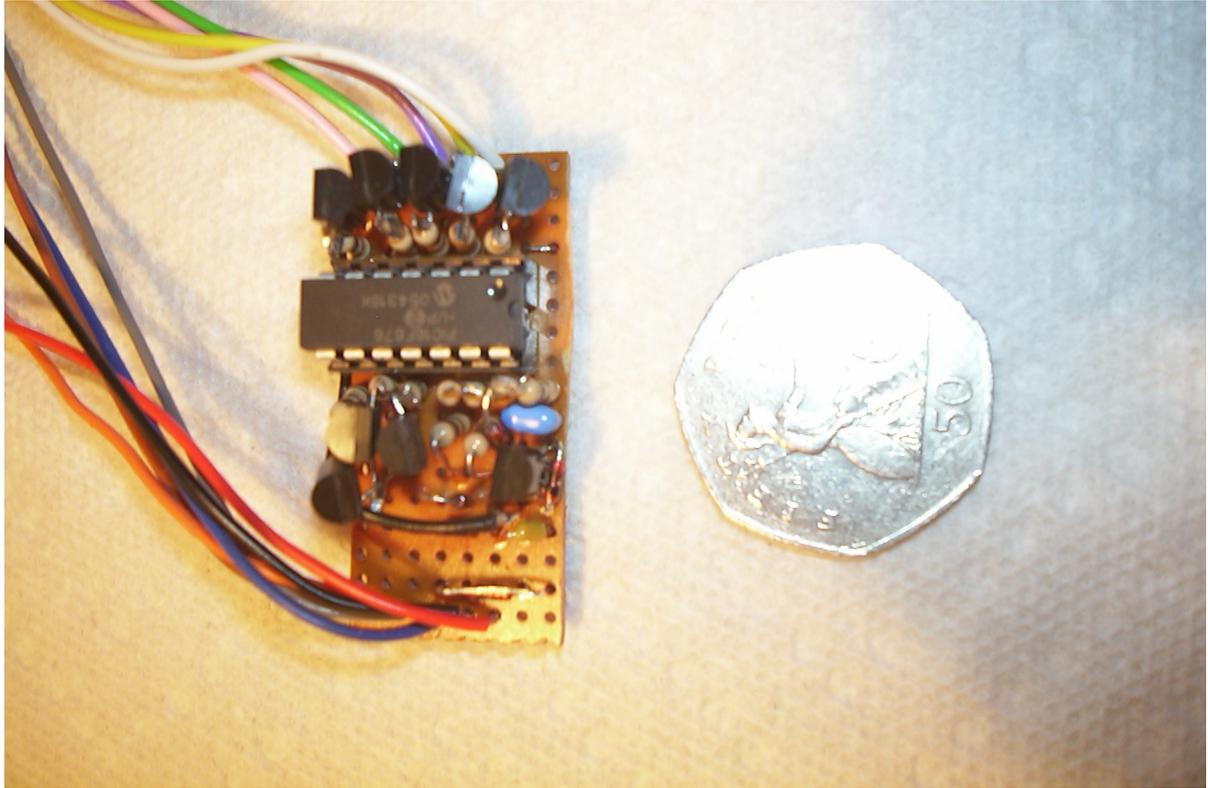


## DIY Decoder Manual



Completed 8-output High Output Power function decoder

### **Overview**

The DIY decoder project started as an example of how anybody with basic soldering skills can produce a useful DCC decoder with a minimum of low cost off the shelf standard sized components at a fraction of the cost of comparable commercial products and with features and functionality not available elsewhere.

The DIY decoder can be made as a 3-output or 8-output version, and with high power outputs or low power outputs (or even a mixture). Because there is no dedicated PCB and the firmware is open-source it is very easy to customise your decoder to meet any particular requirements that you have such as building it on a long piece of stripboard to accommodate LED coach lighting.

The firmware has been designed to be compatible with as many DCC systems as possible and therefore includes programming on the main as well as all four standard programming modes. LokMaus2 programming compatibility is also provided to allow programming with systems that do not support programming full CV ranges and values, and decoder lock is available to allow the decoder to be used in conjunction with other decoders in the same train. As well as working on standard DCC as well as DC it is compatible with both DC braking and asymmetric braking.

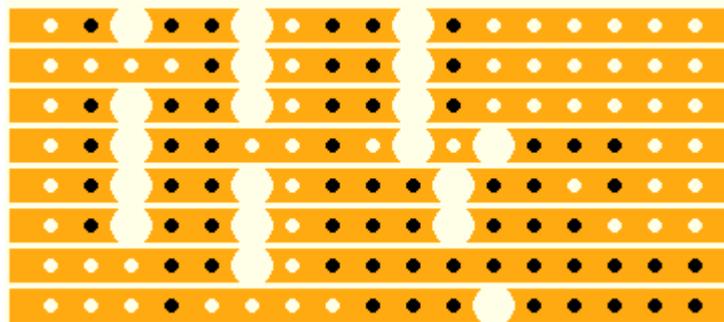
The bipolar outputs available in the low power versions offer extra flexibility in how you connect the LEDs, with the output inversion capability allowing very efficient use of outputs and functions not available on any other decoder. The low power outputs

have a constant regulated voltage of 5V regardless of track voltage giving very good results on DC as well as DCC.

The low cost of the components allows the decoder to be used in places such as brake vans to control a tail light for example that normally would have been discounted on the grounds of cost. Even an ISO container could have a decoder in it for the tail light at the end of a container train.

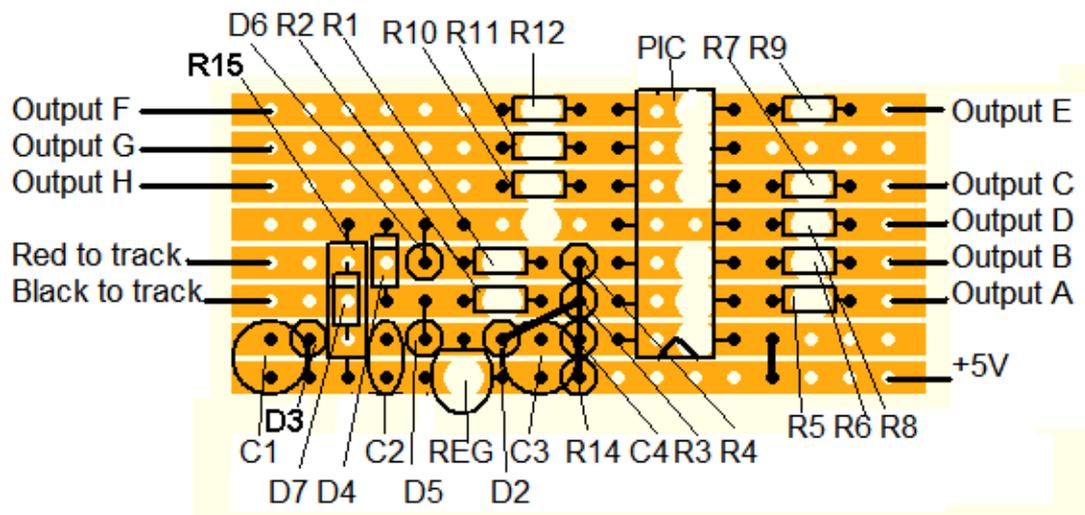
## **Construction**

1. Cut a piece of 0.1" stripboard to the required size. It will need to be at least five strips wide for the three output version or eight strips wide for the eight output version. The stripboard layout diagram shows the suggested way of doing it but you may want to adapt the size of your piece of stripboard to suit the location where it is being fitted. If you are using the decoder to control coach lighting it may be easier to make the stripboard fit the entire size of the roof so that you can mount the illuminating LEDs on it as well as the decoder components. Stripboard is often referred to as Veroboard and available in various sized sheets from electronic component suppliers.

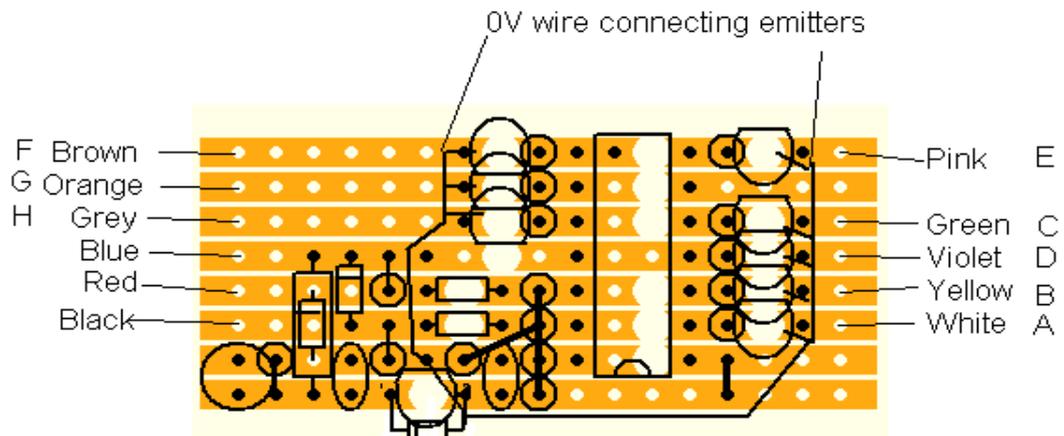


Stripboard cutting guide

2. Cut the tracks in the places marked on the stripboard cutting guide. It is important to do this before you fit the components and is easy to do with a sharp 1/8" drill bit twisted between your fingers if you do not have the proper Vero tool for the job. I have a proper Vero tool but usually just use a drill bit anyway because it is easy and to hand. The diagrams here show 8-output decoders but if you are making a 3-output decoder you do not have the top three strips on the stripboard and just need to leave off all of the components that are shown on the top three strips.

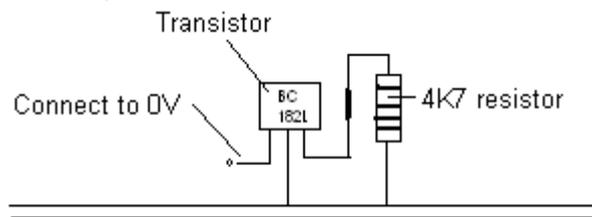


Low power output component placement



High power output component placement

3. Fit the IC socket first (for the PIC IC to go in but leave the PIC out for the time being), followed by the other components working away from the IC socket, soldering them as you go. Take care not to stress the legs on the regulator which can be easily broken and take care not to overheat it or any of the other semiconductors when soldering. In order to fit the components in the minimum space some are vertically mounted and some do not have all their connections to the stripboard, so fix the ends which go to the stripboard first. In the low power output version 1K resistors are specified but these should be adjusted in value to suit the load that is applied.



Transistor mounting detail for high output power decoders

4. You can now test the board offline. A quick visual inspection will pick up most faults where there are solder bridges between tracks but a quick check between adjacent tracks with a buzzing continuity meter will verify that there are no big problems before connecting the power. It is especially important to check that there is no continuity between pins 1 and 3 of the regulator and it is well worth double checking that all of the diodes are the correct way round at this point as well.
5. Apply track power to the red and black wires. You should now be able to measure 5V between pins 1 and 14 of the 14-pin IC socket (or pins 1 and 8 if you have used an 8-pin). Pins 12 and 13 (or 5 and 6 on an 8-pin) should be between 0V and 5V (usually around 1V) while all of the other pins on the IC socket should be 0V, all measured relative to pin 14 (or pin 8 on an 8-pin). the common wire on the top of the vertically mounted components should be 0.65V or thereabouts. Disconnect the track power when this test is complete.
6. Program the PIC with the contents of the .HEX file which contains the firmware. If you do not have a suitable PIC programmer there are some instructions on the web available to enable you to make yourself one that can be connected to a PC parallel port, or you can buy one (best option), or you may be able to find someone that has a PIC programmer who can program the PIC for you. If you want to buy your own programmer I would recommend the Microchip PICkit2 as being both cheap and easy to use. You may need internet access to download the latest firmware in order to be able to program the PIC used in this project. When the PIC is programmed it may fail to verify because it will set up the CVs (which are held in the internal EEPROM) to factory default values while still in the programmer, so do not be too alarmed by this.
7. You can now insert the PIC into the IC socket and connect the red and black wires to the track power. You should be able to control the outputs on loco address 3 and measure the output voltage on each of the output pins of the PIC relative to pin 14 (or pin 8 on an 8-pin ). Note that if you have an 8-pin PIC there is no output C, only A, B and D which will respond respectively to F0 forwards, F0 reverse and F2 by default.
8. You are now ready to connect the decoder to your programming track and program the CVs to your desired settings in the usual way before using your decoder.

## ***Programming***

The configuration values supported by this decoder are listed in the supported CV table on the back page. Most of the standard CVs should be familiar to you and are well documented elsewhere so I will concentrate on those that are more specific to this decoder. All known programming modes are supported including operations mode (Programming On the Main), Address only, Register, Paged and Direct. You should use Direct or POM as preference over the other modes. Programming acknowledgement requires that there is a load of at least 60mA on the decoder outputs when programming and this may be required by some command stations in order to

use paged mode and will be required by all command stations in order to read CVs. If your load is insufficient to meet this requirement it may be possible to put a resistor across the programming track or add an extra load to the outputs of the decoder during programming in order to get acknowledgement to work.

### **CV33-46 and CV83-98 – Function Mapping**

Each of the CVs in this range relate to a function button on the controller. In order to allow the maximum flexibility in being able to map the thirty available functions to the eight outputs all of these CVs are programmed in the same way where a value of 128 will activate output A, 64 output B and so on. This is slightly at variance with the NMRA specification which is rather limited in which outputs can be assigned to which functions

### **CV99 – Output inversion**

Each of the bits in CV99 relates to an output (as in CV33 above), such that when the bit is set to a 1 the related output will be inverted, such that activating the mapped function will deactivate the output. This can be very handy in order to use a single function to switch between day and night lighting for example by mapping one function button to two outputs and inverting one of the outputs.

### **CV100 – Flicker**

Clearing a bit in this CV will give a crude flicker effect on the output when it is activated. This effect is only available on outputs E-H and C and therefore not available on 3-output decoders. It can be used to simulate an oil lamp for example.

### **CV13 and CV14 – Analogue Supported Functions**

The default setting of these CVs enables all of the functions when running on analogue DC. You will need to clear the relevant bits to disable a function from activating automatically on analogue. This is particularly important if you have something that must only be operated momentarily such as a coupling. Only F0-F12 are supported on Analogue.

### **CV21 and CV22 – Consist Supported Functions**

By default all functions will be operated by the consist address when in consist. If you wish some functions to remain under the control of the main address when in consist you will need to clear the relevant bits. Only F0-F12 can be modified by this method.

### **CV15 and CV16 – Decoder Lock**

If you have more than one decoder in a train you should give each decoder its own unique address in CV16. When programming the decoder you will then need to put the decoder's address in to CV15 in order to make any changes to the other CVs apart from changing the address. Putting 255 into CV 15 will override the lock and allow you to program regardless of the decoder address.

### **CV7 – LokMaus mode**

Some command stations have only limited programming capability and do not allow the programming of CVs higher than 99 or with values greater than 99. LokMaus mode allows you to get around this restriction by programming the tens column of both the CV address and CV value into CV7 before programming the CV proper. This

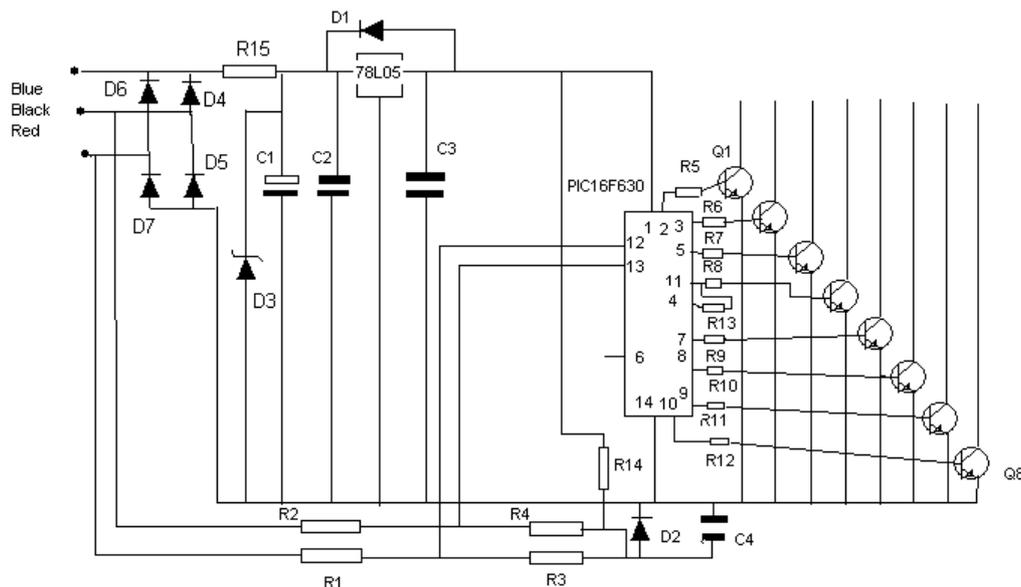
will be required if you need to program your decoder with a Roco LokMaus2 or similar. Program the hundreds column of the CV address into the tens column of CV7 and at the same time the hundreds column of the CV value into the units column of CV7. For example to program CV109 with value 234 you would need to program CV7=12 followed by CV09=34

## Reference

Component list for DIY decoder

- =====
- Preprogrammed PIC16F630 (8-output versions only)
  - Preprogrammed PIC12F629 (3-output versions only)
  - 78L05 regulator
  - Q1-8 BC182L (high output power versions only)
  
  - R1,R2 10K 1%
  - R3,R4 2K2 1%
  - R5-R12 4K7 (high output power versions only)
  - R5-R12 1K (low output power versions only) adjust value to suit LED brightness required
  - R13 0 ohms, link on veroboard between pin 4 and pin 11 (or pin 5 for 12F629)
  - R14 10K
  - R15 47R 2W
  
  - C1 1uF 35V radial tantalum electrolytic
  - C2,C4 100nF axial multilayer ceramic
  - C3 470nF radial multilayer ceramic
  - D1,D2 1N4148
  - D4-D7 1N4148 (low output power versions only)
  - D4-D7 UF4001 (high output power versions only)
  - D3 27V zener diode
  
  - 14-pin IC socket (8-output versions only)
  - 8-pin IC socket (3-output versions only)

DIY Function Decoder



F0 will either enable directional lighting when active (white lights at front, red lights at rear) or parking lighting (red lights at both ends) when inactive.

F1 will switch between night headlights when active and day headlights when inactive.

F2 will switch off the tail lights, but only on a decoder with more than three outputs where the tail lights are connected to output C rather than +5V.

CV No	Front decoder	Rear decoder	Note
16	2	3	For decoder lock
33	128	0	Output A active on F0 forwards only at front
34	0	128	Output A active on F0 reverse only at rear, Output B inactive
35	80	80	F1 activates outputs B and D instead of output C
36	32	32	F2 activates output C instead of output D
99	96	96	Output B and C inverted

CV	Default value	Range	Description	Notes
1	3	0-127	Short address	0=DC only (address zero is broadcast address)
7 (read)	-	-	Software release number	Set to 1 for this release
7 (write)	0	0,1,2,10,11,12	LokMaus2	Used for LokMaus2 programming mode
8 (read)	-	-	Manufacturer ID	Not yet set (still development)
8 (write)	-	8	Factory Reset	8= Reset to factory defaults
13	255	0-255	Functions supported in analogue DC mode	F1 (bit 0) - F8 (bit 7)
14	63	0-63	Functions supported in analogue DC mode	F0 forwards (bit 0), F0 reverse (bit 1) F9 (bit 2) - F12 (bit 5)
15	0	0-7,255	Decoder unlock key	Unlocks decoder when 255 or same as CV16
16	0	0-7	Decoder lock ID	Set lock address of decoder (other values will work)
17-18	0		Long address	
19	0	0-255	Consist address	
21	255	0-255	Consist function control	F1 (bit 0) - F8 (bit 7)
22	63	0-63	Consist directional lighting control	F0 forwards (bit 0), F0 reverse (bit 1) F9 (bit 2) - F12 (bit 5)
29	6	0-255	Configuration bits	Default is analogue DC rather than Brake on DC and 28/126 step
33	128	0-255	Function 0 forward output mapping	128=A,64=B,32=C,16=D,8=E,4=F,2=G and 1=H
34	64	0-255	Function 0 reverse output mapping	See above
35	32	0-255	Function 1 output mapping	See above
36	16	0-255	Function 2 output mapping	See above
37	8	0-255	Function 3 output mapping	See above
38	4	0-255	Function 4 output mapping	See above
39	2	0-255	Function 5 output mapping	See above
40	1	0-255	Function 6 output mapping	See above
41	0	0-255	Function 7 output mapping	See above
42	0	0-255	Function 8 output mapping	See above
43	0	0-255	Function 9 output mapping	See above
44	0	0-255	Function 10 output mapping	See above
45	0	0-255	Function 11 output mapping	See above
46	0	0-255	Function 12 output mapping	See above
83	0	0-255	Function 13 output mapping	See above, CV number may change
84	0	0-255	Function 14 output mapping	See above, CV number may change
85	0	0-255	Function 15 output mapping	See above, CV number may change
86	0	0-255	Function 16 output mapping	See above, CV number may change
87	0	0-255	Function 17 output mapping	See above, CV number may change
88	0	0-255	Function 18 output mapping	See above, CV number may change
89	0	0-255	Function 19 output mapping	See above, CV number may change
90	0	0-255	Function 20 output mapping	See above, CV number may change
91	0	0-255	Function 21 output mapping	See above, CV number may change
92	0	0-255	Function 22 output mapping	See above, CV number may change
93	0	0-255	Function 23 output mapping	See above, CV number may change
94	0	0-255	Function 24 output mapping	See above, CV number may change
95	0	0-255	Function 25 output mapping	See above, CV number may change
96	0	0-255	Function 26 output mapping	See above, CV number may change
97	0	0-255	Function 27 output mapping	See above, CV number may change
98	0	0-255	Function 28 output mapping	See above, CV number may change
99	0	0-255	Output inversion	128 inverts A, 64 inverts B etc.
100	255	0-255	Flicker (outputs C,E,F,G,H only)	Gives flicker effect on outputs, H=1, G=2, F=4, E=8, C=32
105	0	0-255	User ID location 1	For storing user I.D. information (E.G club membership No)
106	0	0-255	User ID location 2	See above

**Supported CV Table for  
V2.16 firmware**

Colour key

Not as per NMRA recommendation

Not NMRA recommended default

For more DIY Decoder information go to the DIY Decoder website:-  
<http://www.dccdiy.org.uk/diy>

DIY decoder design and documentation © Paul Harman 2008-2010

DIY Decoder firmware © Paul Harman 2008-2010