A Fully-Configurable DCC Dual Power Breaker – Part 2

Ready to build ?

Hopefully, if building a Dual Power Breaker (DPB) sparked your interest in Part 1, you now have a printed circuit board (PCB) and a bundle of components ready to be assembled, and have practiced your soldering technique for electronic components.

However, if you do not have any experience of soldering electronic components then you should first have a look at one or two of the guides available on the Internet (such as at www.makerspaces.com/how-to-solder/) and some of the multitude of videos available on YouTube, although there is nothing to beat getting some copper stripboard from one of the component suppliers and practising soldering wires (and a few spare components) to it before tackling the real DPB printed circuit board.

Use resin-cored solder in wire form only – never use solder with an acid flux (as sold for plumbing purposes) – and use a fine-tip soldering iron with a maximum power rating of 25 Watts. All joints should be made as quickly as possible to avoid damaging the PCB and components. The greatest enemy of electronics is heat.

The position and reference number of each component is printed on the top surface of the PCB. Most resistors and capacitors also have their value shown but, where this is not clear, refer back to the component tables in Part 1 to make sure you fit the correct component in each position.

Assembling the Dual Power Breaker

Fit those components with least height to the DPB PCB first, ie. the diodes (D1-D9) and resistors (R1-R43), as shown in [1], so that, when you turn the PCB over and lay it down to solder the component wires on the underside of the board, the components do not fall out of the holes.

Ensure that the diodes are fitted the right way round, with the end marked with a stripe aligned with the corresponding mark on the PCB. It does not matter which way round the resistors are placed on the PCB although I generally try to fit them so that the colour code bands are read from either top to bottom or left to right.

You can also fit the two transient voltage suppressors (TVS1 and TVS2) at the left and right edges of the PCB at this time. These components are not polarised so can be fitted either way round.

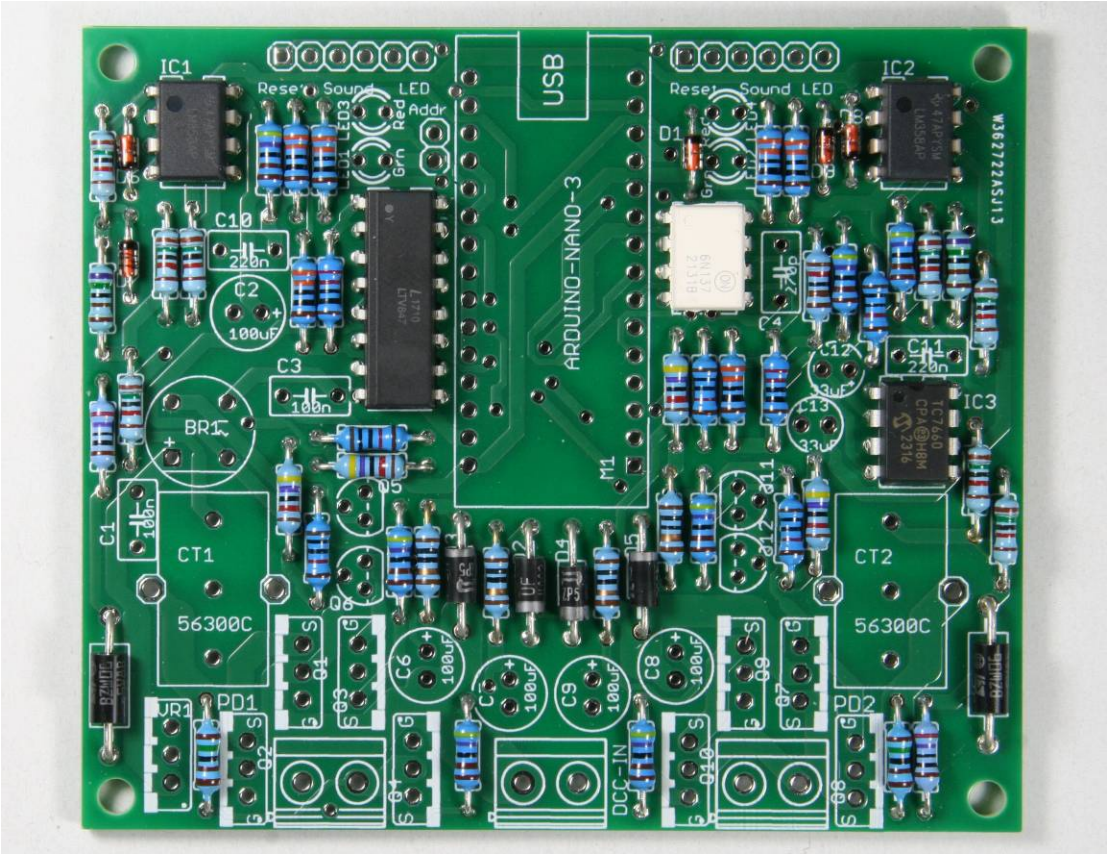
A tip here is to solder just one wire from each component, then turn the PCB over and check that all components are still flush with the PCB. If not, make them so (by briefly melting the one soldered joint while holding the component flat – but watch your fingers – soldered wires are hot!) before soldering the remaining wire(s) of the component. Clip the excess component wires flush with the soldered joints on the underside of the PCB using a small pair of sidecutters.

The next step is to fit the three integrated circuits (chips) (IC1-IC3) and the two optocouplers (OK1 and OK2 – also called optoisolators), as the next tallest components. Ensure that the notch, dimple or dot at one end of each package is towards the top edge of the PCB (the edge where the Arduino Nano USB connection emerges), as indicated by the board markings, and that all pins are through the PCB holes, with none having been bent under the devices.

The pins of integrated circuits are generally splayed out so, to make fitting a little easier, I place the pins on one side of the package in their correct PCB holes then use the end of a steel ruler,

held vertically, to gently push all pins on the opposite side of the chip inwards until they drop into the other set of PCB holes.

Solder two diagonally-opposite pins on the chips first, and then check that the devices are still flat on the PCB. If not, it is easy to melt the solder on the appropriate corner pin while pushing the device down into the correct position, before soldering the remaining pins.



All photographs and diagrams by the author

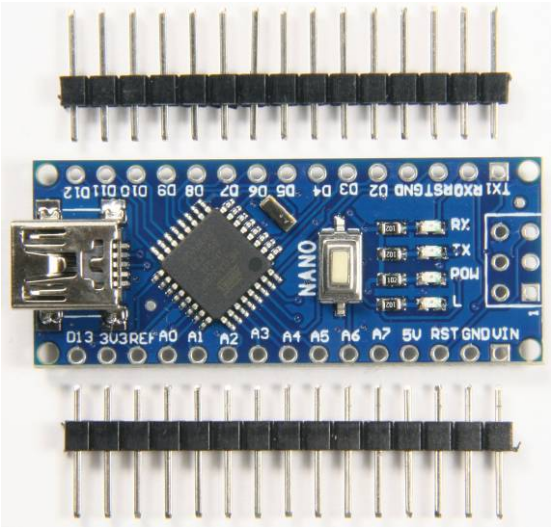
1. Starting assembly of Dual Power Breaker board

The next step is to fit the Arduino Nano-3 module to the DPB PCB but, before you do so, it is highly recommended that you install and configure the Arduino IDE software for the Nano-3, as described in Part 1.

Carefully connect the module to one of your computer's USB ports, using a standard USB-A to USB-Mini-B cable and check firstly that the module powers up correctly. For the Nano-3, you will normally see two LEDs lit – one constant (labelled POW) and one flashing (labelled L) for a short period, depending on the current internal state of the Nano module.

Next, load the sketch for the DPB (DualPowerBreaker_3-8.ino - available from my A-Train Systems website at www.a-train-systems.co.uk/dpb-download) into the Arduino IDE and, after checking that the appropriate settings in the Tools menu are selected, try uploading the sketch to your Nano module. If the upload succeeds then you can carry on with the assembly of your DPB. Otherwise, you will have to try some of the possible fixes outlined in Part 1 – which are much more easily done when the module is not soldered into the PCB.

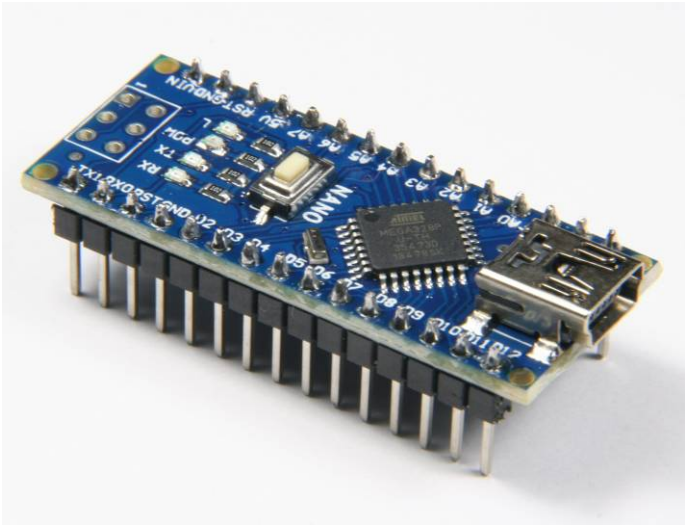
If you purchased the Arduino Nano-3 module 'unsoldered' then you will need to attach the supplied pin header strips [2] to the module before fitting it to the PCB.



2. Arduino Nano – supplied pin header strips

To ensure that you fit the header strips square and flush, a tip is to hold the DPB PCB horizontally above your bench, either in a vice or using a 'third-hand' jig, and then drop the pin headers through the respective rows of holes in the PCB, long pins down.

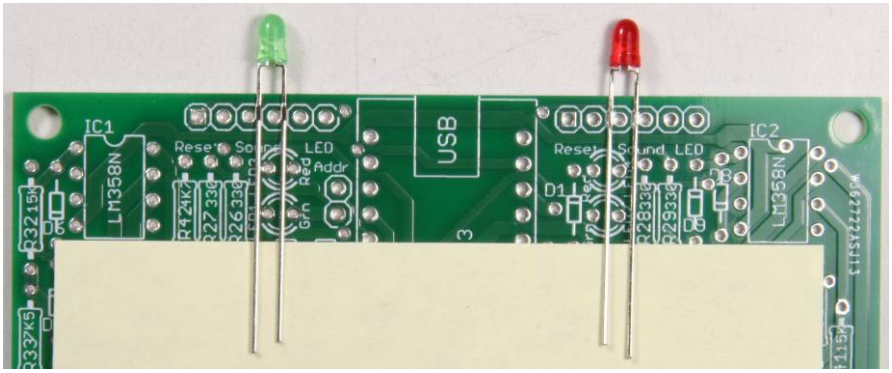
Now fit the Arduino module on to its header strips and proceed to solder it to the header pins – generally starting with the corner pins and checking that the module is sitting square on the header strips before continuing with the rest of the pins. With the Nano-3, make sure that you don't fit any of the corner holes on to the header strips by mistake – each header pin must come through a solder pad, as can be seen in [3] –



3. Arduino Nano – pin header strips fitted

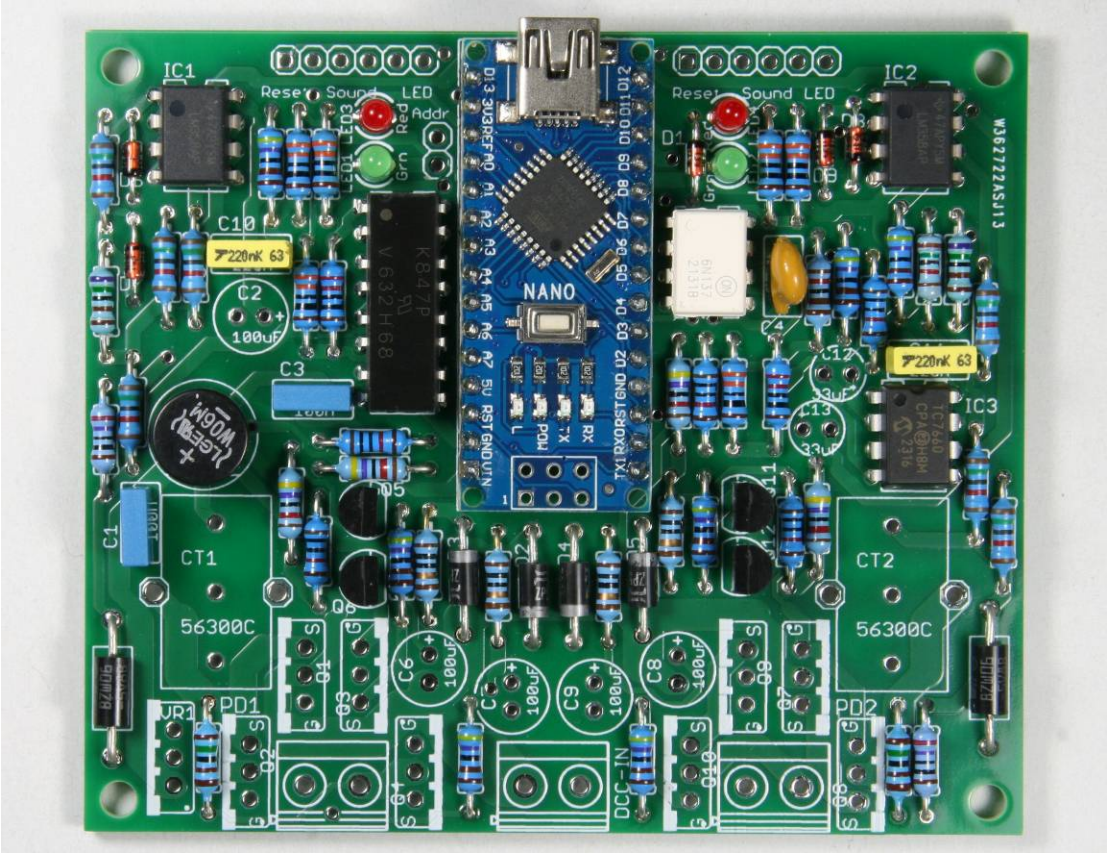
Note that there is no need in this application to fit the additional 6-pin header, which is usually supplied, to the end of the module furthest from the USB connector.

Fit and solder the Nano module on to the DPB PCB, making very sure that the module's USB connector is towards the edge of the board, followed by the four LEDs (LED1-LED4). Note that the LEDs are fitted with their short leads closest to the Nano module, as indicated in [4] –



4. LEDs – correct orientation – short lead towards Nano module

Next, fit the small capacitors (C1, C3, C4, C10, and C11), the bridge rectifier (BR1), and the four 2N3906 transistors (Q5, Q6, Q11 and Q12), ensuring that the flat sides of the transistor packages face inwards towards the centre of the PCB, as indicated by the PCB markings. Your PCB should now look like [5] –

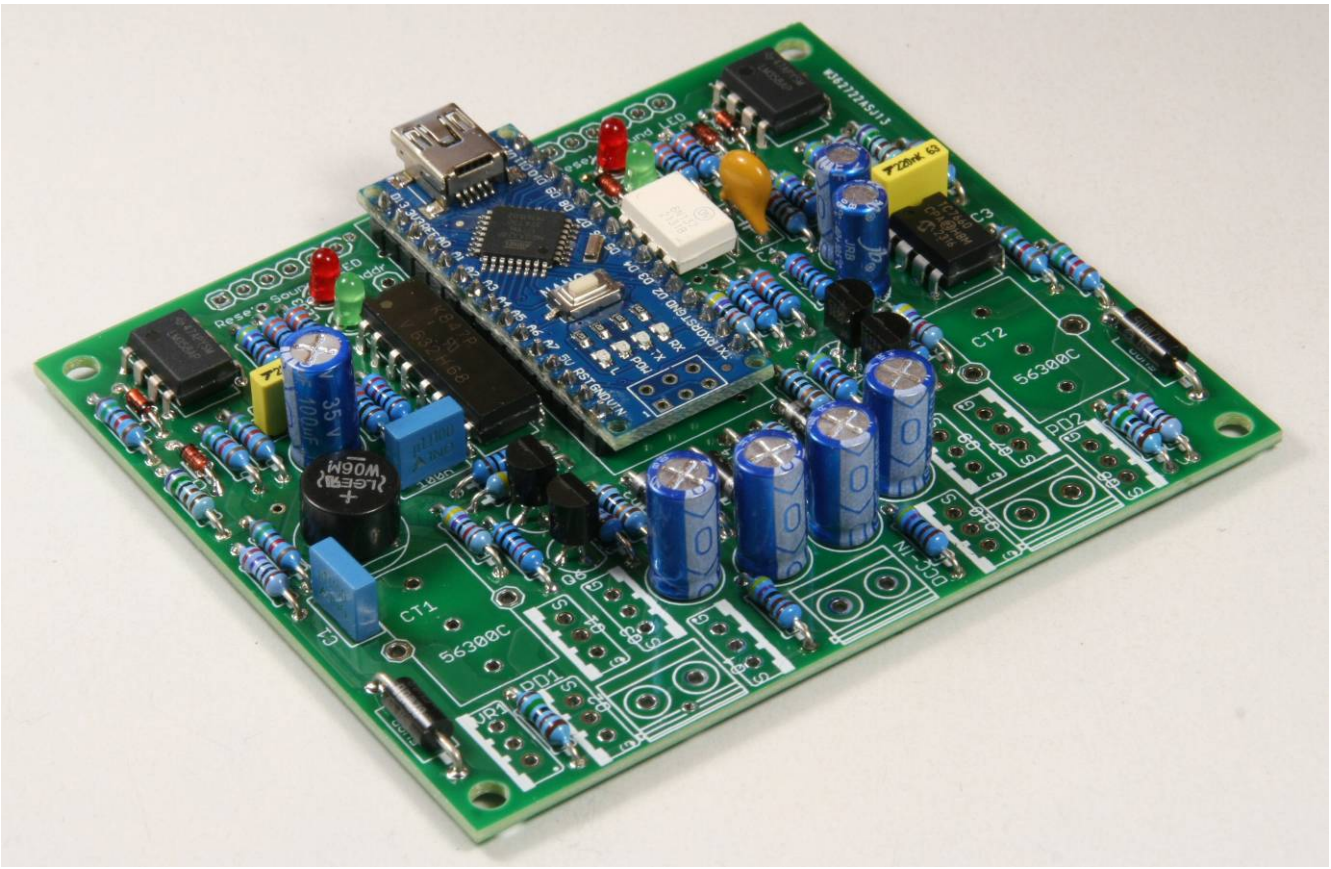


5. Second stage of Dual Power Breaker board assembly

The electrolytic capacitors (C2, C6-C8, C12 and C13) are the next to be fitted. These components are polarised, so take care to fit them the right way round since a reverse-biased electrolytic capacitor can potentially explode. All of these capacitors, except for C12, are fitted with their negative (–) connection (normally denoted by a stripe on the package) facing outwards towards an edge of the PCB. C12, one of the smaller 33uF capacitors located next to pin 1 of IC3, and

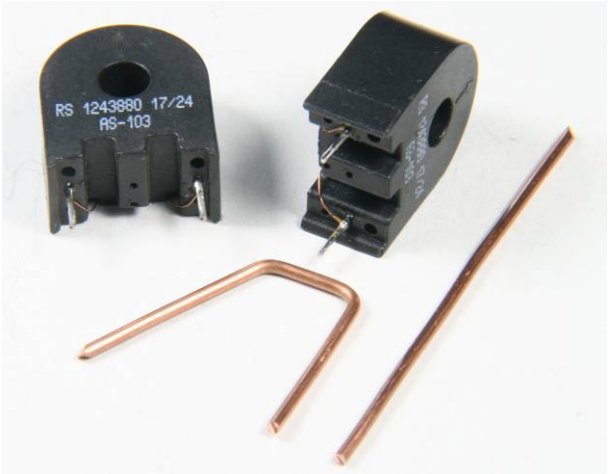
nearest to the 220nF capacitor C11, is fitted with the negative (stripe) connection towards the centre of the PCB.

You can see the correct orientation of all of the electrolytic capacitors in [6] –



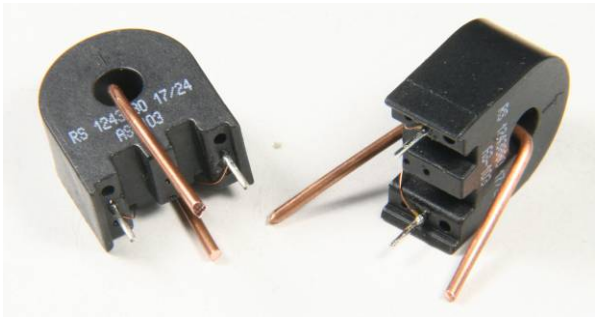
6. Fitment of electrolytic capacitors to DPB board

The next recommended step is to fit the three 2-way terminal blocks for your DCC connections to the PCB, followed by the two current transformers (CT1, CT2). However, before actually soldering the current transformers to the PCB, you need to make the current loops which pass through the hole in the transformers. To do this, for each loop, take a 2” piece of 16AWG copper (not aluminium) wire and bend it as shown in [7] –



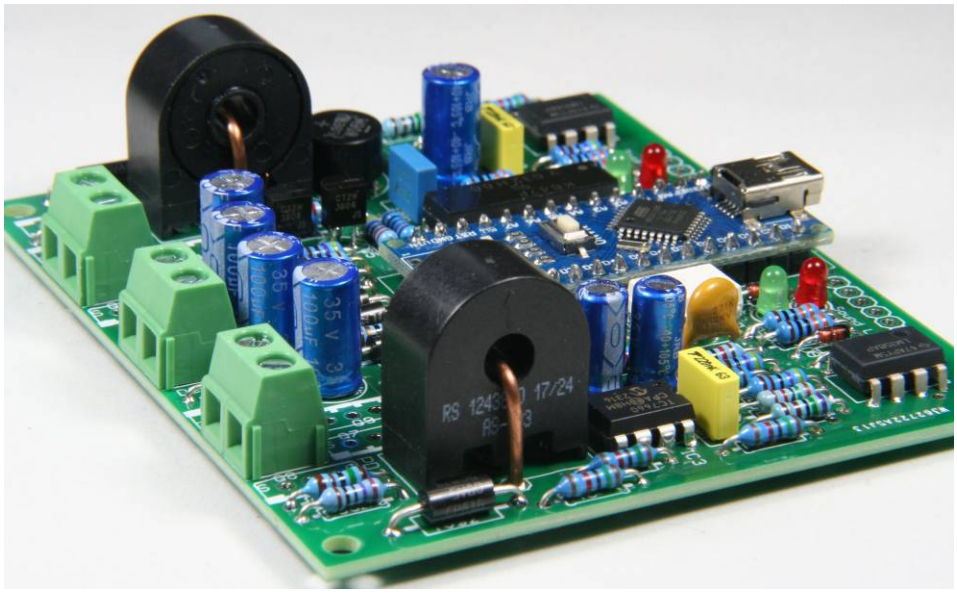
7. Preparation of current loop wire for current transformers

Having bent two pieces of wire as shown, you can now pass each one through the hole in one of the transformers, as shown in [8], and then complete the bend in each wire to bring the “legs” parallel to the transformer sides so that they will fit into the relevant holes in the PCB –



8. Completion of bends in current loop wires

After soldering the transformers and loop wires into the PCB, you should end up with an assembly like that in [9] –

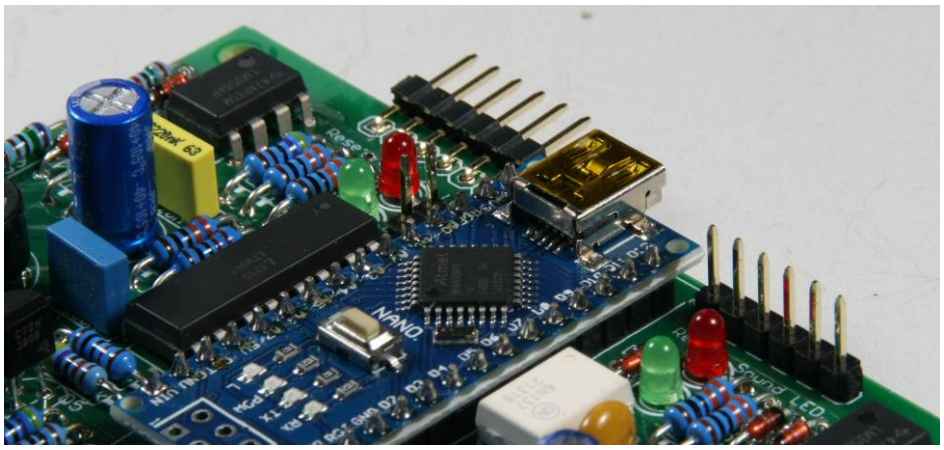


9. Current transformers and main terminal blocks fitted

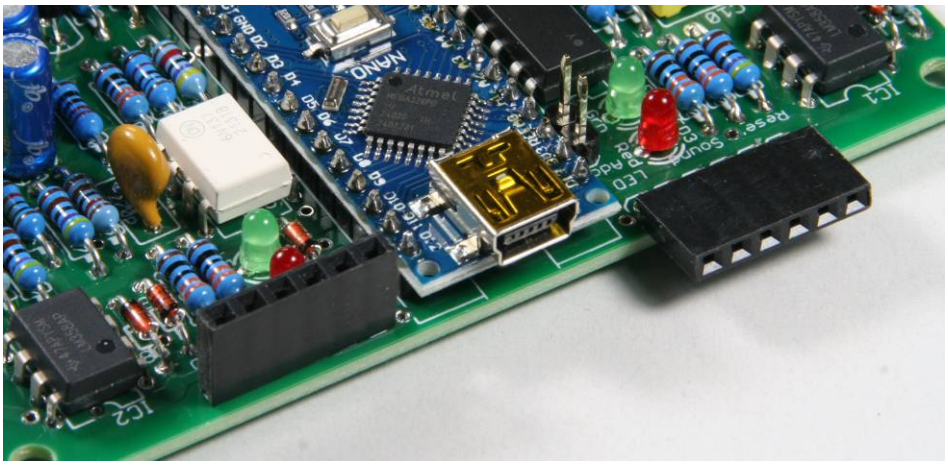
Once you have fitted the 2-pin Address header (Addr) next to the Nano module, you have a choice of options for the DPB auxiliary connectors. You can fit simple vertical or right-angle male 6-pin headers, vertical or right-angle female 6-socket headers, or 6-way screw terminal blocks, as shown in [10], [11] and [12].

You can also, of course, simply choose to solder wires to the PCB using the header holes and take the wires directly to whichever external devices (LEDs, sounders, or pushbuttons) you have selected to install.

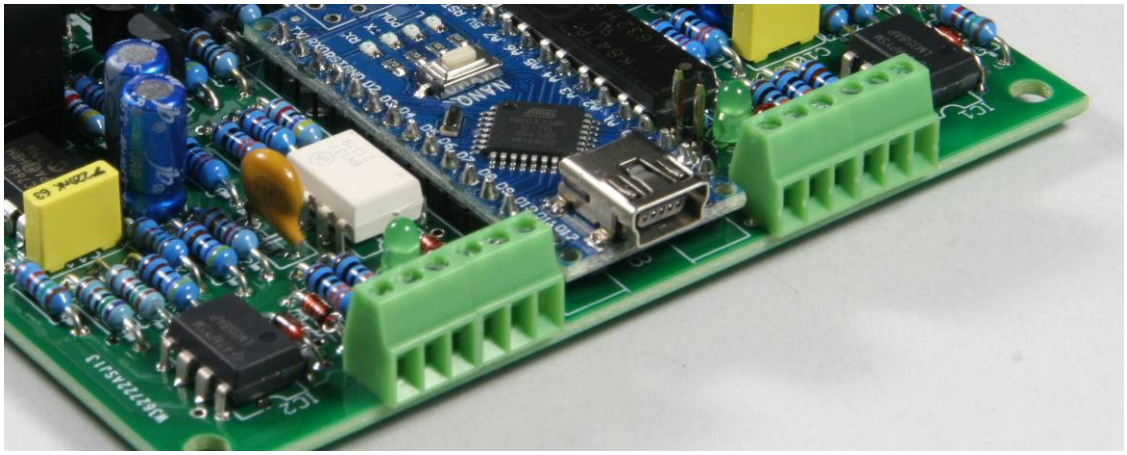
My personal preference is to use screw terminal blocks although they are substantially more expensive than the alternative headers shown. Note that it is often cheaper to buy 2- or 3-way terminal blocks, and fit them side-by-side, than to buy complete 6-way blocks.



10. Male 6-pin headers for auxiliary connections (vertical or right-angle)



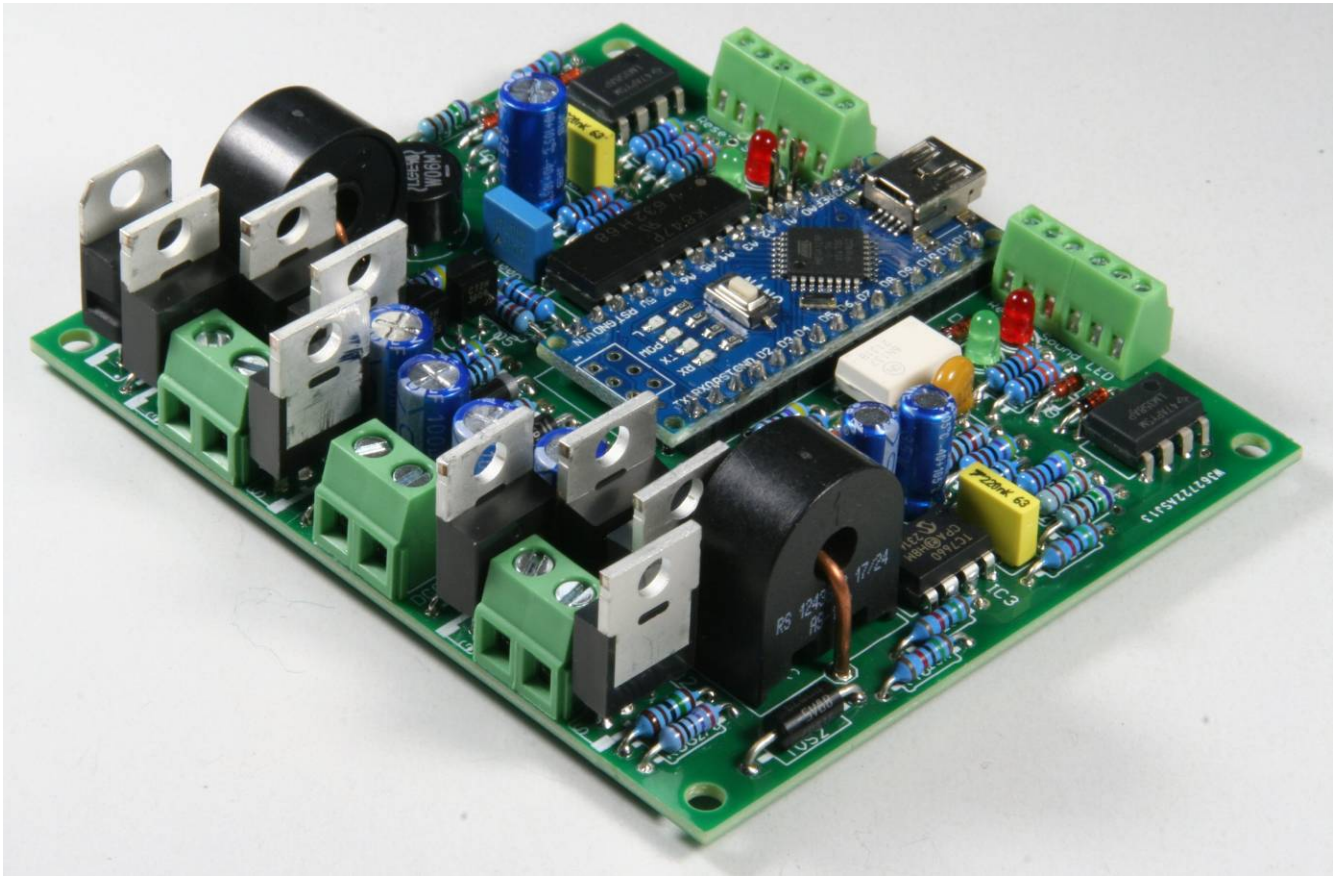
11. Female 6-socket headers for auxiliary connections (vertical or right-angle)



12. Terminal blocks for auxiliary connections (0.1” pitch)

The final components to fit to the DPB PCB are the eight power MOSFETs which are the devices which actually perform the power breaker function, switching the DCC outputs on or off, plus the voltage regulator which supplies power from the DCC input to all of the control circuitry, including the Arduino Nano and the two current sensing blocks.

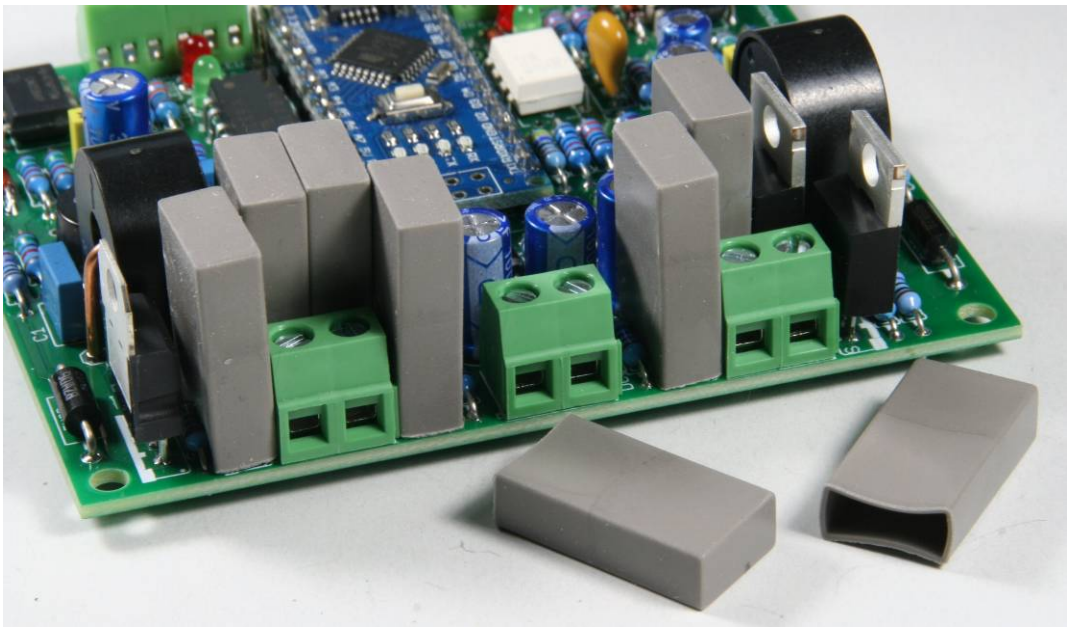
The power MOSFETs and voltage regulator are all 3-pin devices each having a metal tab which must be aligned with the broad white markings on the PCB, so that you end up with the final assembly shown in [13] –



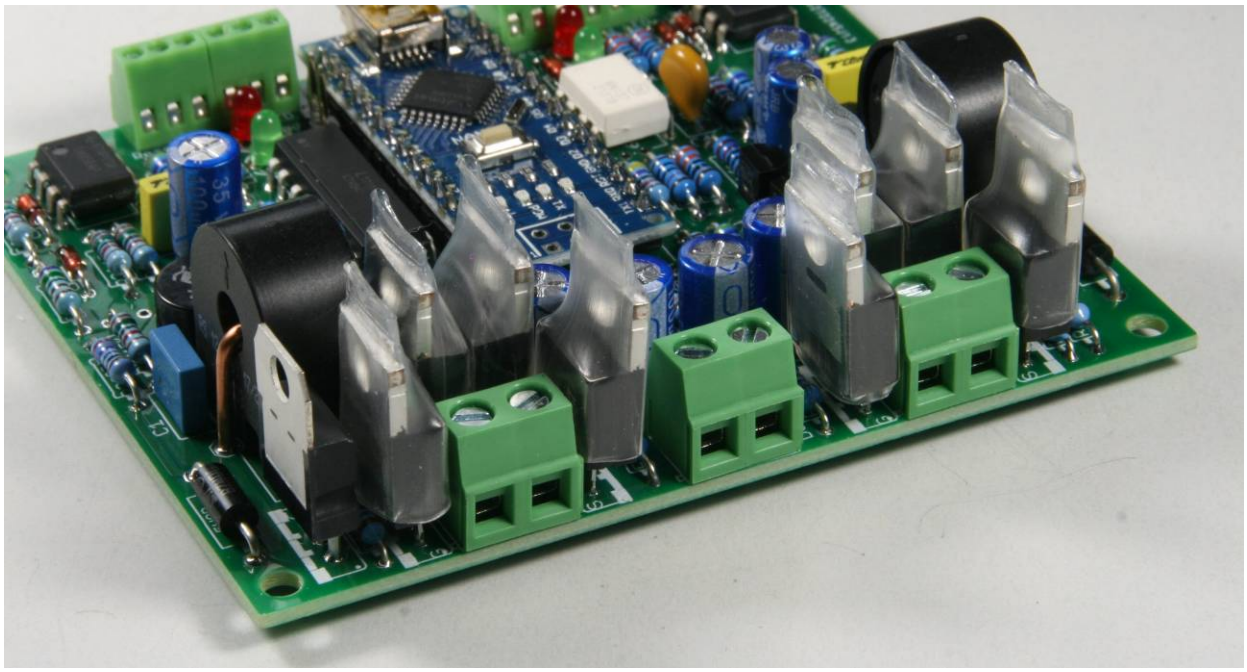
13. Fully assembled Dual Power Breaker

Although the power MOSFETs are positioned on the PCB so that there is no danger of their metal tabs touching if they get accidentally bent out of position, the tabs are still “live”, carrying the full DCC voltage. There is, therefore, a danger that the tabs could accidentally get shorted together by a stray wire or piece of metal, especially if the unit is mounted out of sight under your layout. Such a short could do a lot of damage not only to the Dual Power Breaker but also to your command station or booster.

For safety then, you must insulate the exposed tabs. You can do this either by buying some commercial insulating caps or boots (search eBay or Amazon for “TO220 silicone insulation caps”) as shown in [14] – or you can spare this expense simply by slipping some heatshrink sleeving over each power MOSFET and shrinking it in place with a little applied heat as in [15] below –



14. Insulation caps for the power MOSFETs



15. Insulation by adding heatshrink sleeving to the power MOSFETs

The tab of the voltage regulator is at ground (0V) so you can leave it without any insulation.

In normal operation, with power district currents up to 3A, say, neither the voltage regulator nor the power MOSFETs will get more than slightly warm. Even when a power district carries the maximum rated current of 5A continuously (but definitely not a recommended situation) the temperature rise in the MOSFETs will be less than 30 degC (54 degF), so that no heatsinks are required.

Check your handiwork

Carefully inspect the completed DPB board to check that all of your soldered joints are bright and shiny, and that the solder has wicked through the PCB holes to the component side of the board. Check also that there are no solder bridges between copper pads or component pins anywhere on either side of the PCB. I highly recommend the use of a x5 or x10 hand lens or jeweller's loupe for this inspection.

Next, before you apply power to your completed DPB you should use your multimeter, set to a low resistance range like 2K or 20K, to check the DCC connections for short circuits.

Between the terminals of the DCC In terminal block you will probably see an open circuit if you are using a modern digital multimeter – momentarily you may see a reading on your meter as various capacitors on the board are charged, but thereafter it should show an infinite resistance. However, if you are using an older analog meter which, when measuring resistance, generally applies a higher voltage to the circuit, you will see a resistance of between 3K and 15K depending on which way round you connect the probes (the voltage applied by the meter probes switches on the input diode of optocoupler OK1 or its protection diode D1).

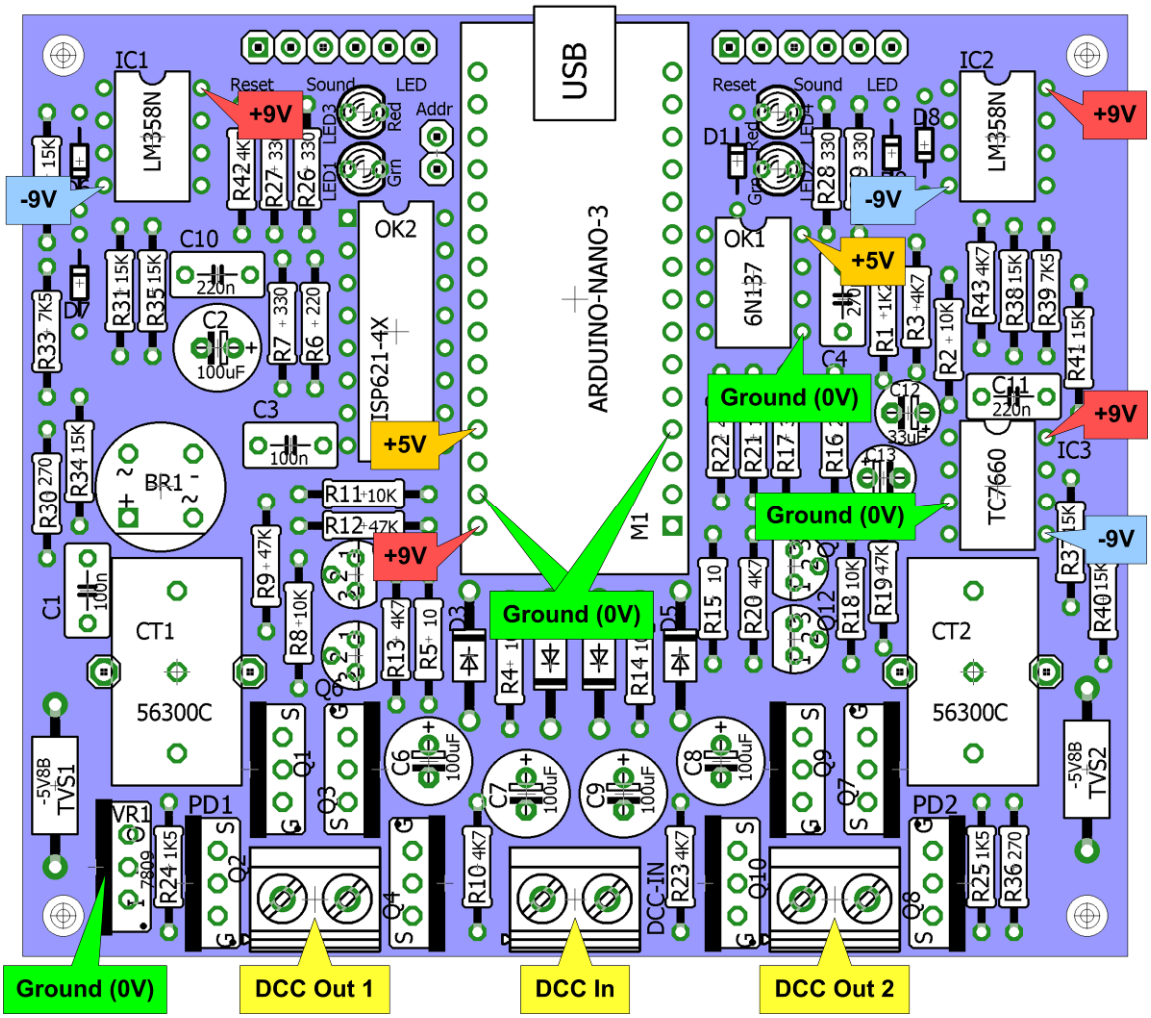
The resistance between the terminals of each power district DCC Out block should be 1K5 ohms. If you see any different value then carefully check your soldered connections, and component values, especially around the loops through the current transformers.

Take care also, if you use the terminal block screwheads as contacts to touch with your meter probes, that the screws are tightened first – loose screws tend to make poor electrical contact and may give you erroneous readings.

You can now connect the DPB DC In terminals to the track output of your command station or booster. This should light up the POW LED on the Arduino Nano together with the two green LEDs on the DPB itself. Using an AC voltage range on your multimeter, check that the voltage at each DCC Out terminal block is the same as voltage you measure at the DCC In terminal block.

Switching to a DC voltage range on your meter, next check that the right supply voltages are reaching the various integrated circuits on the DPB. The metal tab of the voltage regulator (see [15] and [16]) is at ground (0V) so clip one of your multimeter's probes to it as a convenient reference point. Use the other probe to check the voltages on the integrated circuit pins indicated in [16], being very careful not to short any of the pins you are probing to any adjacent pin or to a nearby component.

The measured voltages should be within 0.25 volts of the nominal values, since the voltage regulators here are not precision devices but, if you see any significantly different values, you should disconnect the DPB from DCC power and carefully check the DPB for components with the wrong value, or in the wrong place, and recheck the quality of your soldered joints.



16. Check the DPB integrated circuits for correct voltages

Connect the DPB to the Arduino IDE

If, as described in Part 1, you have installed the Arduino IDE and gained some familiarity with its operation, you are ready to compile and upload the necessary sketch (software program) to the DPB. You can download the sketch (DualPowerBreaker_3-6.ino) from my A-Train Systems website (www.a-train-systems.co.uk/dpb-download) to any convenient folder on your computer, and then open it from the File menu in the Arduino IDE.

Connect a Mini-B USB cable from your computer to the Nano USB socket. This will power the Nano whether or not you have DCC track power connected to the DPB. Next, if you have not already done so by following the steps outlined in Part 1, you need to set the correct board type for the sketch by opening the Tools menu and following the various options appropriate to the type of Nano you have fitted to the DPB.

Now click the Upload (→) button to transfer the compiled sketch to the module. You should see the Tx and Rx LEDs on the Nano light for varying lengths of time as the transfer proceeds. Once the sketch has been uploaded, the Nano will retain all of the code even when it is switched off.

If everything is working correctly both green LEDs on the DPB should be lit, showing that both power districts are enabled. If you have DCC track power connected to the DCC In terminals then you should also have DCC power available from both DCC Out terminals.

Configuring the Dual Power Breaker

The DPB operates as a DCC accessory and is controlled by the set of configuration variables (CVs) listed in Table 1 [17].

CV Number	Default Value	Description	
41	31	Power Breaker Address LSB	Address can range from 0001 to 2043
42	0	Power Breaker Address MSB	
43	6	District 1 Current Limit	Set in 0.25 amp steps, from 0.25A to 5A Step values 1 to 20
44	6	District 2 Current Limit	
45	25	District 1 Trip Delay	Time before breaker acts, from 5 to 255 msec
46	25	District 2 Trip Delay	
47	12	District 1 Reconnect Delay	Time before breaker attempts to reconnect Steps of 250msec, 0.25s to 60s (1- 240)
48	12	District 2 Reconnect Delay	
49	0	Use Dual Power Breaker as Autoreverser if CV = 90	
50	0	Set all CVs to default values if CV50 = 0 – otherwise CV50 = 173	
51	0	Not used at present	
52	10	Auto reconnect District 1 if CV52=10, else reconnect with Manual pushbutton	
53	10	Auto reconnect District 2 if CV53=10, else reconnect with Manual pushbutton	
54	0	Enable output of current values measured by Nano ADCs to the Arduino IDE Serial Monitor if CV54 not zero	
55	0	Enable output of sketch debug messages to Serial Monitor if CV55 not zero	
56	0	Not used at present	

17. Table 1 - Dual Power Breaker configuration variables

The default values of these CVs are set as soon as you load the sketch to the DPB, so that you can use the DPB immediately if these parameters meet your requirements –

- Current limit for each power district = 1.5 amps
- Time current has to exceed limit before switching off = 25 millisecs
- Time after breaking before an automatic reconnect is attempted = 3 secs

Each power district operates independently so that an overload in one district has no effect on the operation of the other. If, after a reconnection attempt, the district current still exceeds the current limit (for 25msec) the DPB will switch that district off again – and the cycle repeats until the district overload is removed.

You can use one of your DCC system's handheld controllers to set the DPB CVs to alternative values. Simply issue a program-on-the main (POM) command to the default DPB address (31) to change the value of any of the CVs in Table 1 [17]. Consult the system manual for your DCC system if you are unsure of how to program CVs in this way.

Be aware that the NmraDcc library code used by the sketch does not check that the value you set for any CV using a POM command is within the allowed range. Please take care that you only set CV values within the ranges defined in Table 1 [17].

If the default DPB address (31) conflicts with an address you are already using on your layout, or you have built several DPBs and wish to give them different addresses, you can set the address to any new value in the range 1 to 2043.

With the DPB connected to your DCC system, fit a jumper across the 2-pin header labelled ‘Addr’, next to Nano pin A0. Both green LEDs should go out and the red LED of District 1 should light. Now send an accessory command from a handheld controller, as though you were throwing a turnout, using the address you want to give to the DPB. You can throw the “turnout” in either direction, straight or diverging. If successful, the lit red LED will start flashing as a signal that the address has been accepted, and that you should remove the ‘Addr’ jumper to return the DPB to normal operation with both green LEDs lit – and now responsive to the new address.

Unfortunately, although you get this positive confirmation that the new address has been set, otherwise the DCC system does not give you any feedback to verify that any of the other CV values have been set as required.

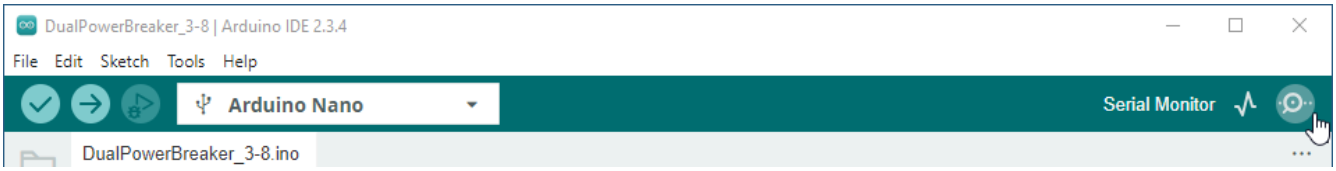
To remedy this, I included a facility within the Arduino sketch to handle the reading and writing of all CV values (including the DPB address) by using the Arduino IDE Serial Monitor and a set of simple typed commands – as listed in Table 2 [18].

Description	Command	Notes	Response
NOP	N	Clears Debug and Output ADC Flags	“OK”
Return CV Status	S	Displays 16 values, comma separated	“CVs - ” + CV41–CV56 values
Return Address	A		“Address = nnnn”
Read Version	V	Displays Major.Minor version	eg. “3.8”
Enable Debug Messages	G		“OK”
Enable ADC Display	K	Displays ADC values, 5 times/sec	“OK” followed by “Dist 1 ADC = ” xxx – “Dist 2 ADC = ” xxx
Set Address	Tnnnn	Replace ‘nnnn’ with up to 4 digits for new address – error message is displayed if outside range 1 - 2043	“Address = nnnn”
Set Trip Current District 1	C1nn	Replace ‘nn’ with Index value 1 - 20 (Trip current = Index x 0.25A)	"Dist 1 Trip Current = x.xxA"
Set Trip Current District 2	C2nn		"Dist 2 Trip Current = x.xxA"
Set Trip Delay District 1	D1nnn	Replace ‘nnn’ with delay value in range 5 to 255msec	"Dist 1 Trip Delay = xxxmsec"
Set Trip Delay District 2	D2nnn		"Dist 2 Trip Delay = xxxmsec"
Set Reconnect Delay Dist 1	R1nnn	Replace ‘nnn’ with Index value 1 – 240	"Dist 1 Reconnect Delay = xx.xxsec"

Set Reconnect Delay Dist 2	R2nnn	(Delay = Index x 0.25sec)	"Dist 2 Reconnect Delay = xx.xxsec"
Set Auto Reconnect Dist 1	M1A		"Dist 1 Reconnect = Automatic"
Set Manual Reconnect Dist 1	M1M		"Dist 1 Reconnect = Manual"
Set Auto Reconnect Dist 2	M2A		"Dist 2 Reconnect = Automatic"
Set Manual Reconnect Dist 2	M2M		"Dist 2 Reconnect = Manual"
Reset CVs to Factory Default	F		"CVs Reset"
Break Power District 1	B1	If in Autoreverser Mode these commands just generate an error message – with no change of which power district is enabled	"Dist 1 Switched Off"
Break Power District 2	B2		"Dist 2 Switched Off"
Break Power Both Districts	BB		"Dists 1 & 2 Switched Off"
Resume Power District 1	P1		"Dist 1 Switched On"
Resume Power District 2	P2		"Dist 2 Switched On"
Resume Power Both Districts	PB		"Dists 1 & 2 Switched On"
Enable Autoreverser Mode	UE	Dist 1 enabled, Dist 2 disabled	"AutoReverser Mode Enabled"
Disable Autoreverser Mode	UD	Leaves both Districts disabled	"AutoReverser Mode Disabled - Both Districts Switched Off"

18. Table 2 – Dual Power Breaker interface command set

Assuming that you still have the Arduino IDE open with the DPB sketch loaded, and the DPB connected to your computer with the USB cable, click the Serial Monitor icon ( ) in the top right corner of the IDE window, as shown in [19] –



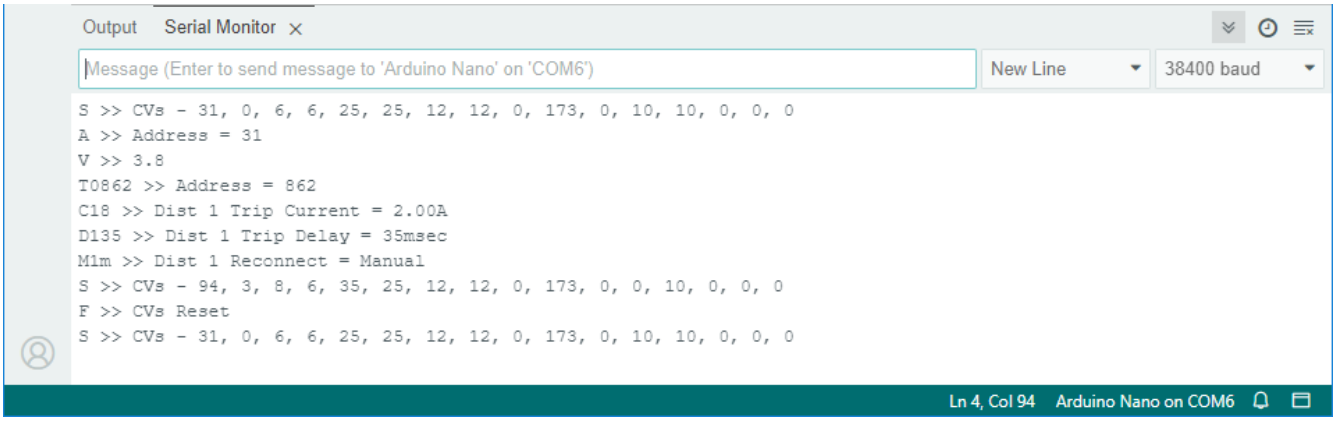
19. Opening the Serial Monitor from the Arduino IDE

This action opens the Serial Monitor in the lower part of the IDE window where the first thing you need to do is to set the baud rate to 38400. You can then type each required command, with its parameters, into the 'Message' box. Leave the terminating selection at 'New Line', and there is no need to enable timestamps (the icon). Then press the Enter / Return key to send the complete command to the DPB.

You can use either upper or lower case letters for all alphabetic commands and parameters, and all numeric parameters and result values are decimal. Note also that, in the case of the numeric parameters which follow the T, C, D, and R commands, any leading zeros are ignored.

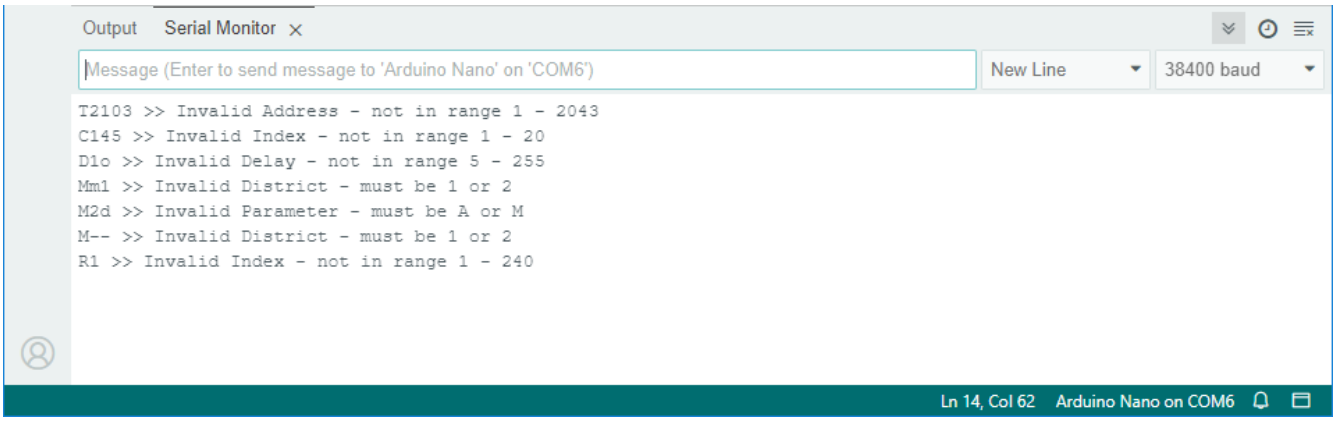
Once a command is accepted, you will get a response in the Serial Monitor window as shown in Table 2 [18]. If for any reason you mistyped the command then you will get an error message instead, with some indication of what was wrong with your entry.

An example of what you will see in the Serial Monitor window after a sequence of commands is shown in [20]. Here, the entered command is “echoed” in the Serial Monitor, with the command character always displayed in upper case, followed by the actual parameter characters that you typed in. The response from the DPB is then displayed following the ‘>>’ symbols.



20. Command display in the Serial Monitor

Generally, the response will confirm that your change to the DPB settings is what you wanted but, if not, it is a simple matter to re-enter the command with the right parameters. Where you enter a command with invalid parameters, you will get an error message such as those shown in [21] – none of the CV values will be changed in these cases.



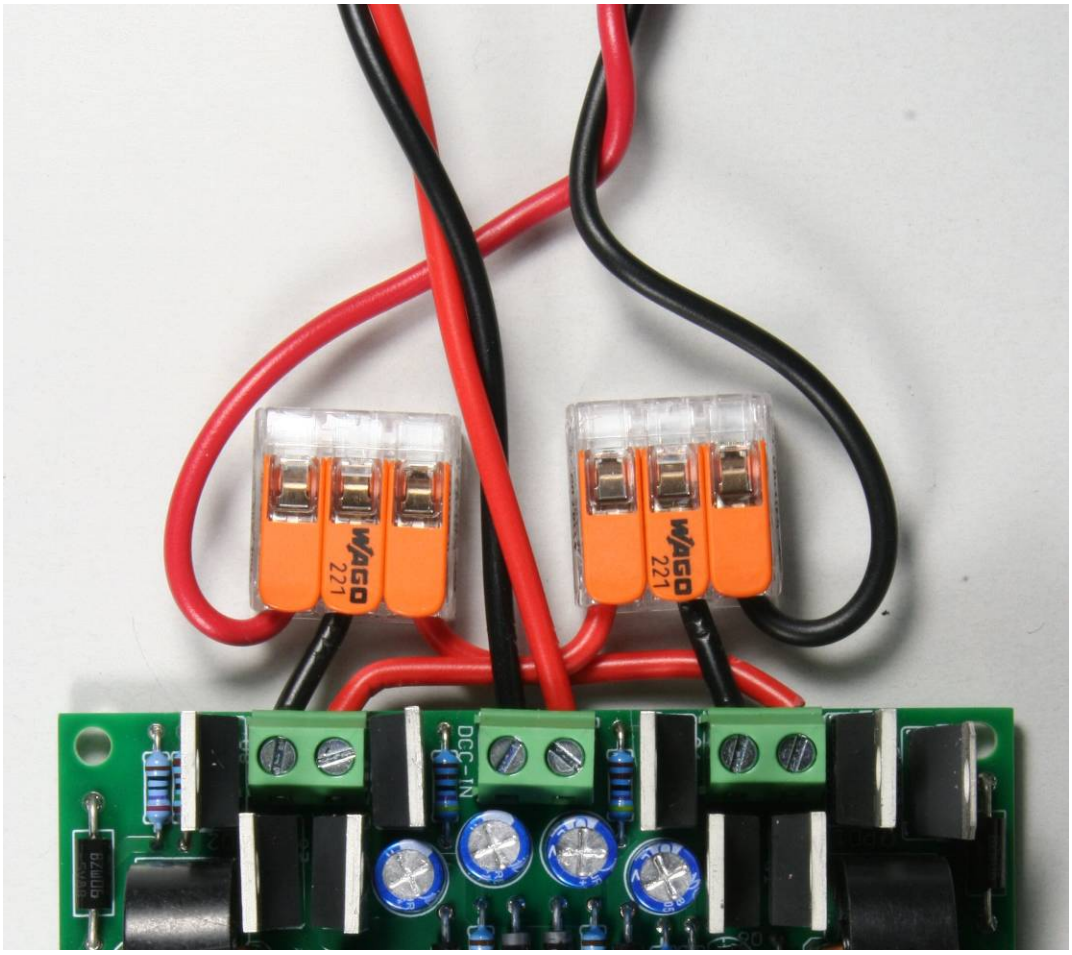
21. Error responses in the Serial Monitor

Using the Dual Power Breaker as an Autoreverser

As mentioned in Part 1, you can use the DPB as an autoreverser. In this mode both channels are used to control the DCC phase applied to the reversing loop or wye, while still providing fully-programmable over-current protection for the district represented by the loop or wye.

Before making any connections to the DPB from your command station or booster, or from the power district outputs to the track, your first step must be to connect the DPB Nano to your computer and use the Arduino Serial Monitor to send a ‘UE’ command to the DPB to enable the autoreverser mode. In this mode the DPB will have District 1 enabled (green LED lit) and District 2 disabled (red LED lit).

You can now link the terminals of the two power districts together as shown in [22], with the left terminal of power district 1 going to the right terminal of power district 2, and vice versa. Connect these commoned terminals to the track of your reversing loop or wye, and connect the DCC In terminals to your command station or booster as usual.



22. Dual Power Breaker connected as Autoreverser

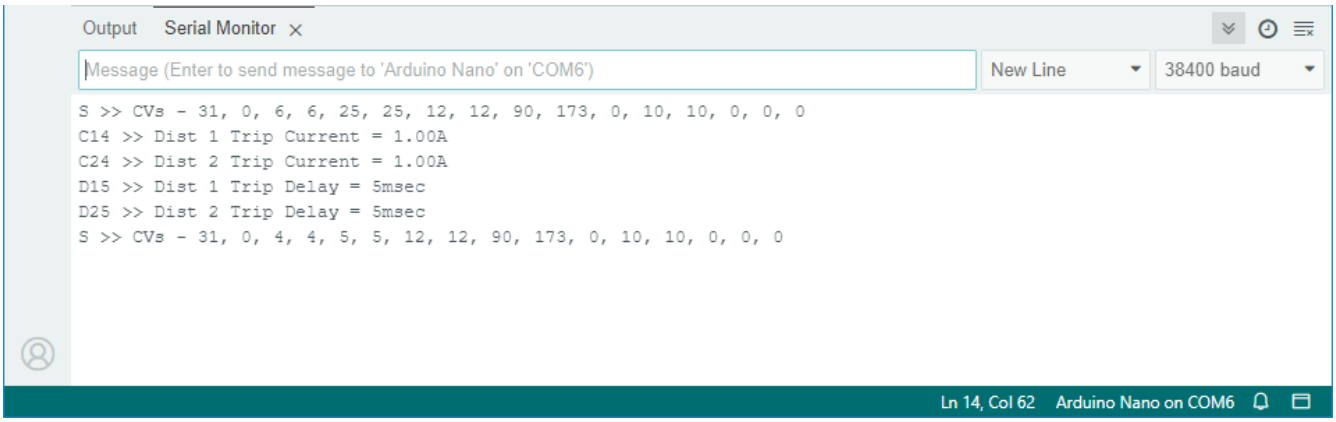
The autoreverser works by detecting the momentary short-circuit, and associated surge in track current, when your locomotive bridges the gap between the main track and the reversing section whenever these sections of track have different DCC polarities (or, more strictly, phases). The DPB reacts by disabling the presently-active power district and enabling the other power district, effectively swapping over the connections (and DCC phase) to the reversing section.

Obviously you don't want the current surge as the locomotive enters the reversing section to trigger the power breaker protecting the mainline track, so I recommend setting the trip current for the autoreverser a little lower. You should also set the autoreverser to trip faster than your main circuit breaker.

Setting the autoreverser to trip more speedily is not only to prevent tripping any upstream breaker but also to allow the autoreverser to handle any other current overload within the reversing section. When the autoreverser detects a current load higher than its trip setting it will first of all switch to the other district but then, if the overload lasts longer than 4 times its set trip delay, it will cut off all power to the reversing section, ie. both districts are switched off.

After the set reconnect delay the autoreverser will attempt to apply power from the last enabled power district. If the overload is still present then the cycle repeats until you fix whatever is causing the problem.

If you have your mainline DPB set to the default values (1.5A trip current and 25msec delay) then I suggest you set a DPB in autoreverser mode to trip at 1.00A after a 5msec delay, using the Serial Monitor commands shown in [23].



23. Setting Autoreverser parameters

Note here the value of CV49 = 90, denoting that the DPB is in autoreverser mode, as a result of having previously sent a 'UE' command to the DPB. The final display of CV values, using the 'S' command, shows the trip currents and delays set as required.

If you wish to return the DPB to normal operation, simply send it a 'UD' command. This will leave both power districts switched off to prevent any damage if you still have the power district terminals wired together. Now, disconnect the cross-coupling, and set up whatever parameters you want in order to use the DPB as a normal power breaker.

Dr Terry Chamberlain

Terry Chamberlain got into model railroading almost by accident in the 1990s when he responded to a request from some modellers in California to build a DCC system based around an Atari personal computer – and he had to build a simple layout to prove that it all worked. Eventually the project evolved into A-Track, a Windows application to provide full computer support for the complete range of NCE DCC systems, with facilities similar to JMRI's Decoder Pro and Panel Pro.

Terry is a professional electronics engineer and spent most of his career in the UK defence industry designing, and managing the development of, large real-time computer systems for the Royal Navy. Now that he is fully retired he is beginning to make progress building the small logging and mining layout he has been planning for many years (after several visits to Colorado) – but keeps getting distracted by new computer and electronics projects for model railroading.