Quad Servo Decoder-Monitor – Technical Details

Introduction

For anyone who is interested, the circuit schematics of the Quad Servo Decoder-Monitor and its subsidiary modules are shown on the following pages, with brief descriptions of how everything operates and fits together. Also included is a section on accessing the Configuration Variables (CVs) of the Decoder section using JMRI Decoder Pro or my own A-Track application.

QSDM Decoder Section and Decoder Keypad

Power for the board from an external source (9 - 15 volts AC or DC) is rectified as necessary by diode bridge BR1 to supply the voltage regulator VR1 which, in turn, supplies +5 volts to the Arduino Nano, its associated circuitry, and the attached servos. Although the normal current through the regulator, with servos inactive, of about 45 mA keeps its power dissipation below 0.5 watts, this could rise briefly to as much as 10 watts if the maximum input voltage is being applied and all four servos are commanded to drive simultaneously. Hence the need for the heatsink attached to the regulator (and the recommendation to keep the external supply down to 9 volts).

The DCC input circuitry is adapted from the design by Wolfgang Kuffer (https://mrrwa.org/dcc-decoder-interface/) and is used by Geoff Bunza as the basis for several of his projects.

The input DCC signal is connected to the input of optoisolator OK1 via resistor R1. Capacitor C4 filters out any high-voltage spikes from the track, and diode D1 prevents the optoisolator input diode from being fatally reverse-biased by the negative-going part of the DCC signal.

The output from optoisolator OK1 is a replica of the DCC waveform, but at a safe +5 volt level, so that DCC command packets can be input to digital input D2 of the Arduino Nano module. Here they are decoded by the NmraDcc library functions, and relevant accessory commands then passed to the QSDD sketch code.

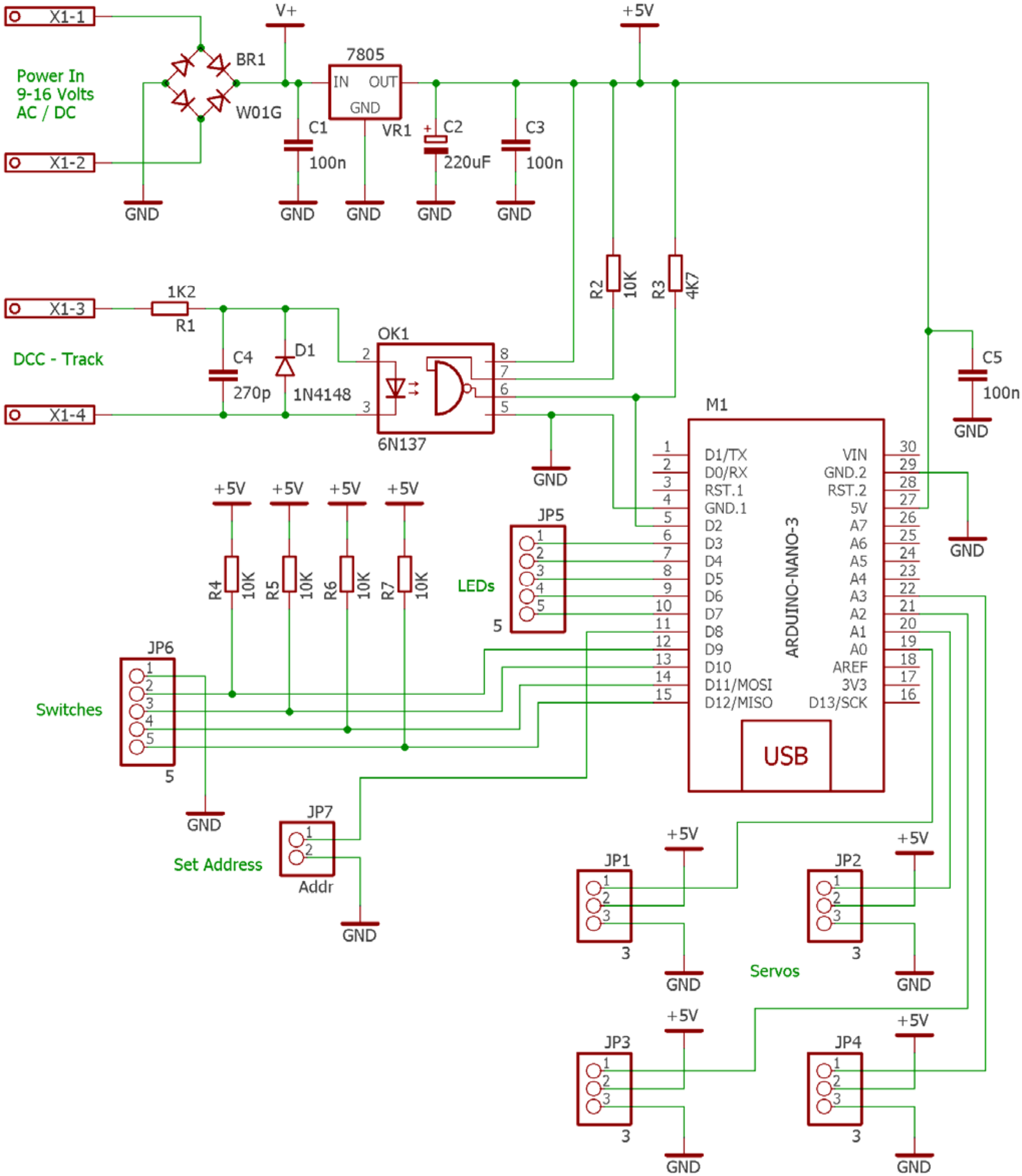
The remainder of the circuitry covers the various inputs to, and outputs from, the Arduino Nano.

The LEDs mounted on the keypad are driven from outputs D3 – D7 via connector JP5 on the decoder and connector JP1 on the keypad with resistors R1 – R5 on the keypad setting the current through each LED to approximately 13 mA.

Pins D9 – D12 are set as inputs which are connected to the four pushbuttons mounted on the keypad via connector JP6 on the decoder and connector JP2 on the keypad. The inputs are normally pulled up to a HIGH level by resistors R4 – R7 on the decoder, but are taken to a LOW level (GND or 0 volts) whenever the appropriate pushbutton is pressed.

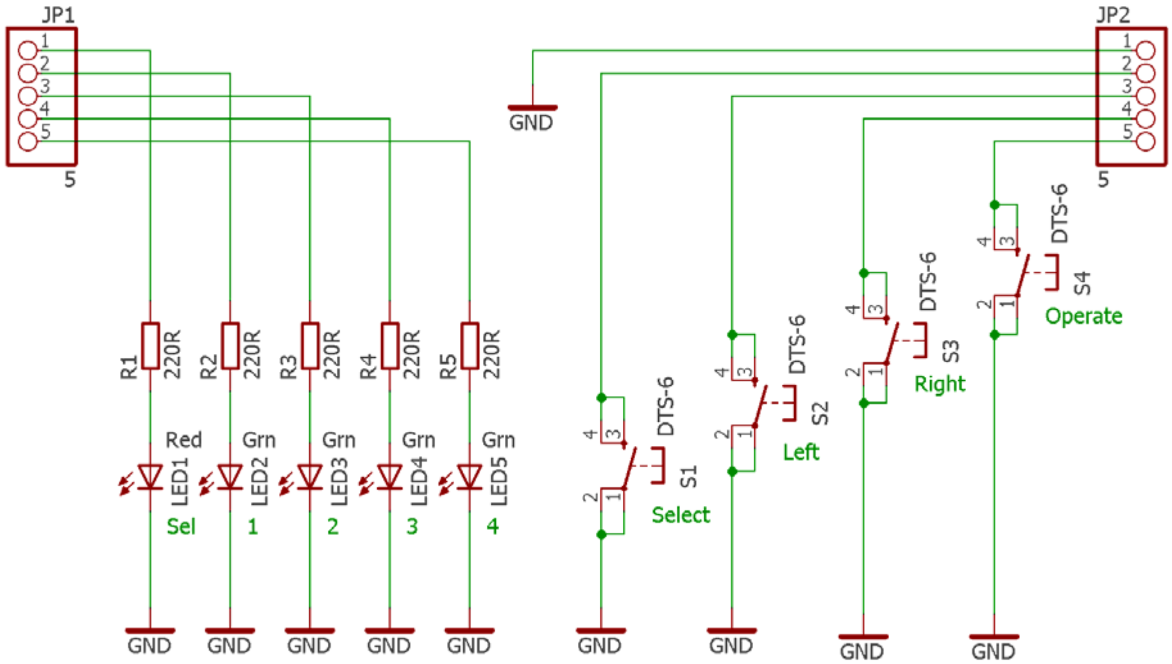
Pin D8 is similarly taken to a LOW level whenever the programming shorting link is fitted to header JP7. This pin uses an internal pull-up, rather than an additional external resistor, to maintain its level HIGH when the link is not fitted.

Finally, Pins A0 – A3 are each set as outputs to drive one of the four attached servos, connected via the three-pin headers, JP1 – JP4 on the decoder board.



Quad Servo Decoder-Monitor – Decoder Section

Copyright © 2021-2022 JT Chamberlain, A-Train Systems



Quad Servo Decoder-Monitor – Decoder Keypad

Accessing Configuration Variables and DCC Operations

You can access the QSDM Decoder section Configuration Variables (CVs) either directly using your DCC system's Handheld controller, and following the manufacturer's instructions, or through your computer (and your system's Command Station) using either JMRI Decoder Pro or my own A-Track application.

However, in all cases you will need to connect your QSDM to the programming track and arrange connections so that only the Decoder section of the QSDM is powered from the DCC supply. To do this you need to disconnect the external power supply (and any USB cables) from the QSDM, and connect both sets of terminals (DCC and external power) to the programming track. You also need to remove the Monitor Power Enable shorting link from the relevant 2-pin header (JP9 – see the QSDM Monitor section schematic) so that the QSDM Monitor section is not powered.

These power supply changes ensure that the Command Station receives proper feedback from the QSDM Decoder, and its attached Keypad, when reading or writing CVs, in the form of a small current surge when all of the Keypad LEDs are briefly switched on.

The set of Configuration Variables used by the QSDM Decoder to hold its working data remains more or less as presented in Part 2 of the original Quad Servo DCC Decoder (QSDD) article, published in *Model Railroad Hobbyist March 2020* with the addition of an Extended Decoder Version group (CV109 – CV112) to identify the decoder as a QSDD (or QSDM Decoder) together with the software version, as listed in the table below –

CV No.	Default Value	Description
01	1	Board Address LSB – internal use – ignore any value loaded here
07	89	NmraDCC Version
08	13	Manufacturer (Do-It-Yourself)
09	0	Board Address MSB – internal use – ignore any value loaded here
29	226	Decoder Configuration – Extended Accessory + Output Addressing
41	1	Output 1 Address LSB
42	0	Output 1 Address MSB
43	2	Output 2 Address LSB
44	0	Output 2 Address MSB
45	3	Output 3 Address LSB
46	0	Output 3 Address MSB
47	4	Output 4 Address LSB
48	0	Output 4 Address MSB
50	0	Load Default CV Values if Not = 173 (0xAD), Auto set = 173 after load
55	6	Servo 1 - Rate (1 = Fast to 16 = Slow)
56	1	Direction - 1 = Normal Operation, 0 = Reverse Operation
57	70	Right Limit
58	110	Left Limit
59	70	Current Position
60	6	Servo 2 - Rate (1 = Fast to 16 = Slow)
61	1	Direction - 1 = Normal Operation, 0 = Reverse Operation
62	70	Right Limit
63	110	Left Limit
64	70	Current Position
65	6	Servo 3 - Rate (1 = Fast to 16 = Slow)
66	1	Direction - 1 = Normal Operation, 0 = Reverse Operation
67	70	Right Limit
68	110	Left Limit
69	70	Current Position
70	6	Servo 4 - Rate (1 = Fast to 16 = Slow)
71	1	Direction - 1 = Normal Operation, 0 = Reverse Operation
72	70	Right Limit
73	110	Left Limit
74	70	Current Position
109	81	Extended Decoder Version 1 – “Q”
110	83	Extended Decoder Version 2 – “S”
111	68	Extended Decoder Version 3 – “D”
112	83	Software Version – Hex = 0x53

Copyright © 2021-2022 JT Chamberlain, A-Train Systems

In addition to this "working set", there are three other CVs which control "special" functions –

CV No.	Default Value	Description
120	0	Set = 120 to load Default CVs – must be cleared manually
121	24	Address to set CV Values in Operations Mode (Program on Main) – LSB
122	0	Address to set CV Values in Operations Mode (Program on Main) – MSB

Dealing first with the "special" functions, the address held in CVs 121 and 122 is used to write a new value to any CV within the decoder using Operations Mode (otherwise called Programming on the Main). Set your DCC system via your Handheld controller to Operations Mode and prepare to use a Locomotive Address (**not** an Accessory Address) equal to the address held in CVs 121 and 122, which is 24 by default.

Select the CV number to be programmed, enter the new value, and press ENTER (or whatever key is required by your system to complete the command). Since it is not possible to read back the values of CVs in Operations Mode, you will have to judge, by the subsequent behaviour of the decoder, whether the change of CV value was a success.

The other "special" function, initiated by programming a value of 120 into CV120 (via Operations Mode), is handled by the NmraDcc library and loads all CVs with their default values the next time the decoder is either reset or powered up. Reset is accomplished by pressing the Reset button on the Arduino Nano or, by holding down both the **L** and **Op** pushbuttons on the keypad and then releasing them.

However, it appears that CV120 is not cleared automatically by the NmraDcc library, so that CVs continue to be reset to their default values each time the decoder is restarted until you change the value in CV120 yourself.

To avoid this inconvenience, the QSDM Decoder has been set up to use CV50 as an alternative. Any value *except* 173 in this CV will reset all CVs to their default values when the decoder is next started or reset, following which the value of CV50 will be set to 173 (hex 0xAD = "All Default") automatically. This means that the default load only occurs once, and will happen automatically when the decoder sketch is loaded into the Arduino Nano for the first time (when all CVs will normally contain the value 255).

Assuming that you have the QSDM Decoder connected to your programming track, as described in the introduction to this section of the document, and have the keypad attached to the decoder, you can then read or write any of the CV values. Following the software update to software version 5.2 any variety of Command Station can be used for programming, and all of the restrictions as outlined in Part 2 of the original article no longer apply.

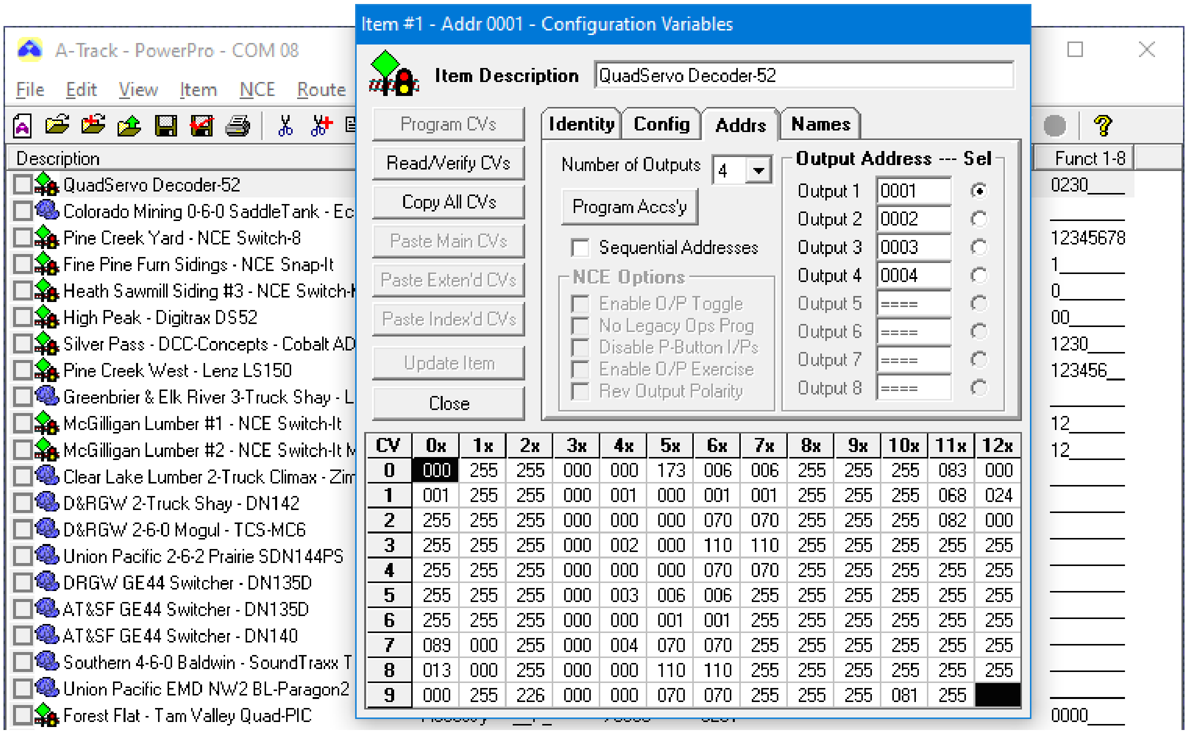
If you have any type of NCE DCC system plus a Windows computer, my own application, A-Track (www.a-train-systems.co.uk/atrack), will happily read and program the Decoder CVs and allow you to save a record of them to file. You can download A-Track and its full documentation from my website (https://www.a-train-systems.co.uk/download.htm#ATWindows).

The principal advantage of being able to save a complete set of CVs to file shows itself when you have a layout with a lot of turnouts and multiple QSDM Decoders to drive them. After setting up one decoder, and its four associated turnouts, to your satisfaction, you can then take a copy of the amended CVs and transfer them in a single operation to all of your other QSDMs.

Setting up these other QSDMs, when they are transferred from the programming track to the layout, will consist only of small adjustments to the servo throws to compensate for differences between individual servos (and perhaps turnouts or linkages).

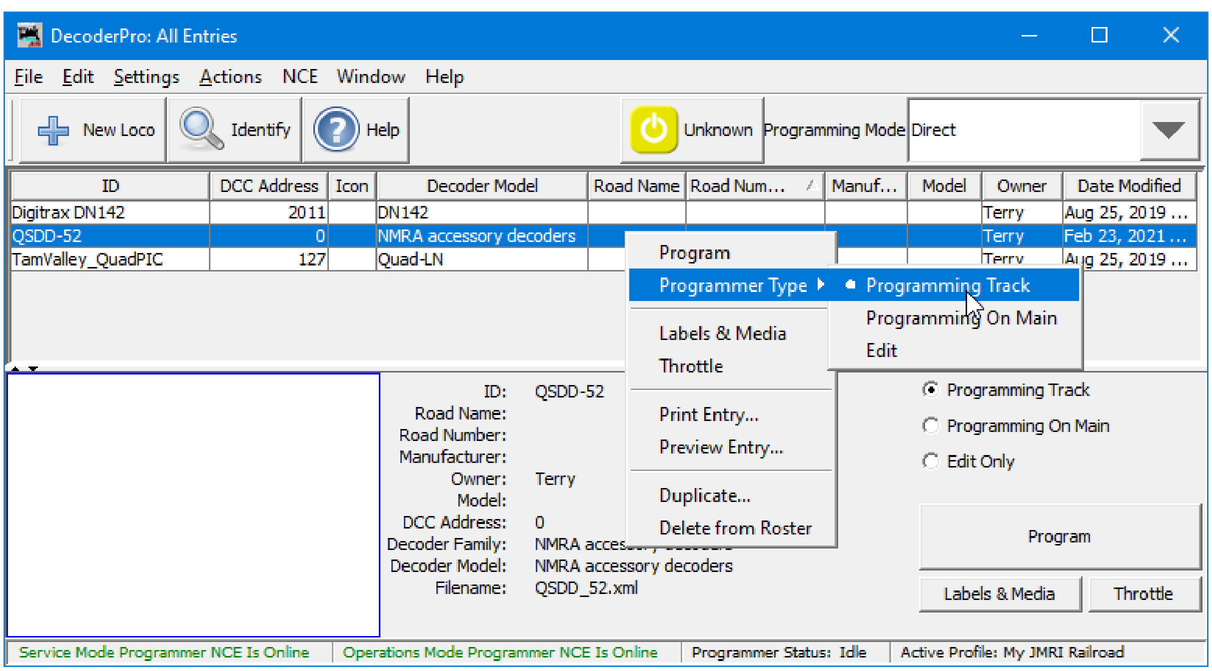
The screenshot below of the A-Track application shows the Configuration Variables window for the decoder with all of the relevant CVs set at their default values –

Copyright © 2021-2022 JT Chamberlain, A-Train Systems

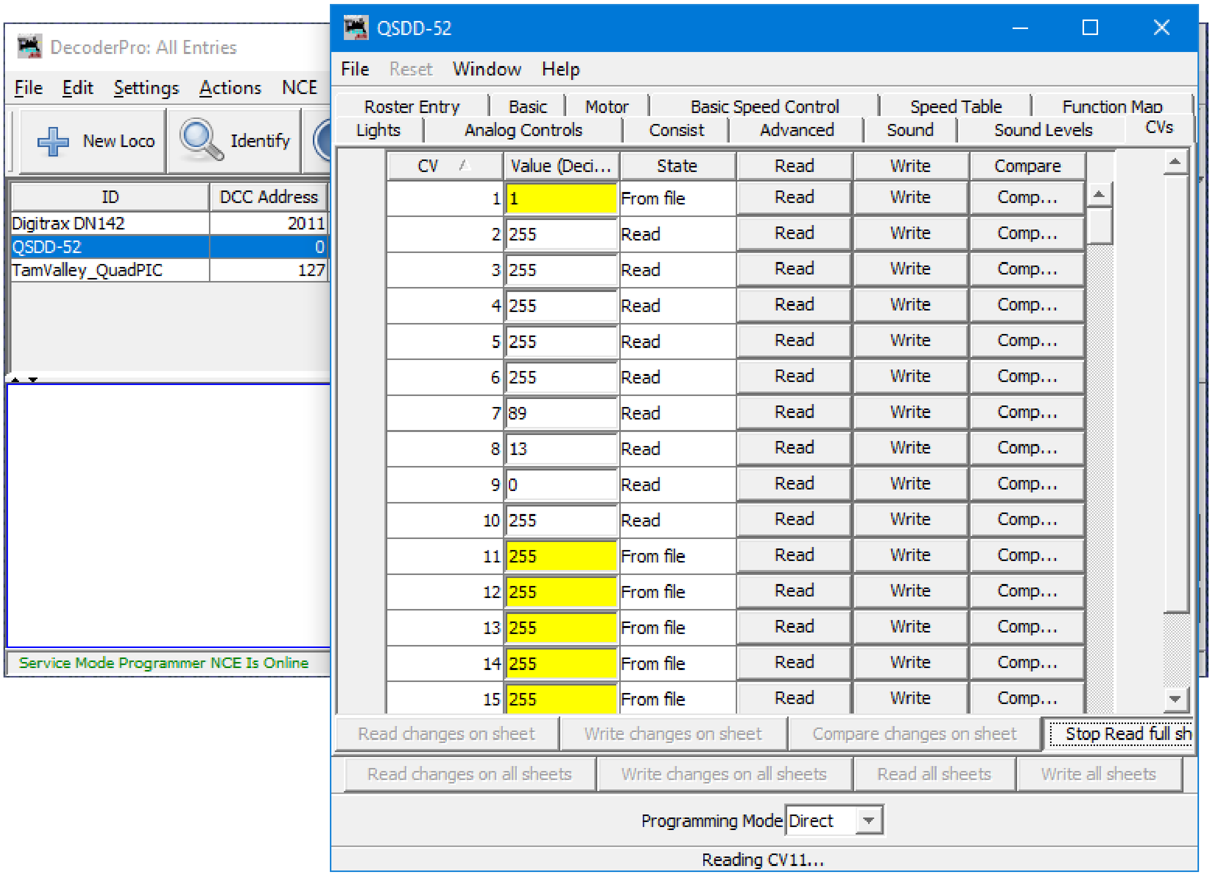


Adjustments to any of the CV values can be made simply by typing in a new value (or set of values, and then using the application's Program CVs function to transfer them to the connected decoder.

With Command Stations other than NCE, you can use JMRI's Decoder Pro software (https://www.jmri.org/help/en/html/apps/DecoderPro/index.shtml) to similarly read and program QSDD CVs, and to save a copy to a file on your computer. Adding the QSDM Decoder to your roster in Decoder Pro is done by clicking New Loco, selecting NMRA as the manufacturer (at the top of the list) and then 'NMRA accessory decoders' as the decoder type. You can then proceed to access the QSDM Decoder on the programming track –



– and then click 'Program' to open the programming window, select the 'CVs' tab, and click 'Read full sheet' to start reading the QSDM Decoder Configuration Variables –



QSDM Monitor Section

Although built on the same PCB as the QSDM Decoder section, the QSDM Monitor section is functionally completely separate. The only shared part is the +5 volts supply from the Decoder section voltage regulator VR1, which is connected to the Arduino Pro-Micro module and the MAX487 RS-485 tranceiver via a shorting link (Monitor Power Enable) fitted to header JP9.

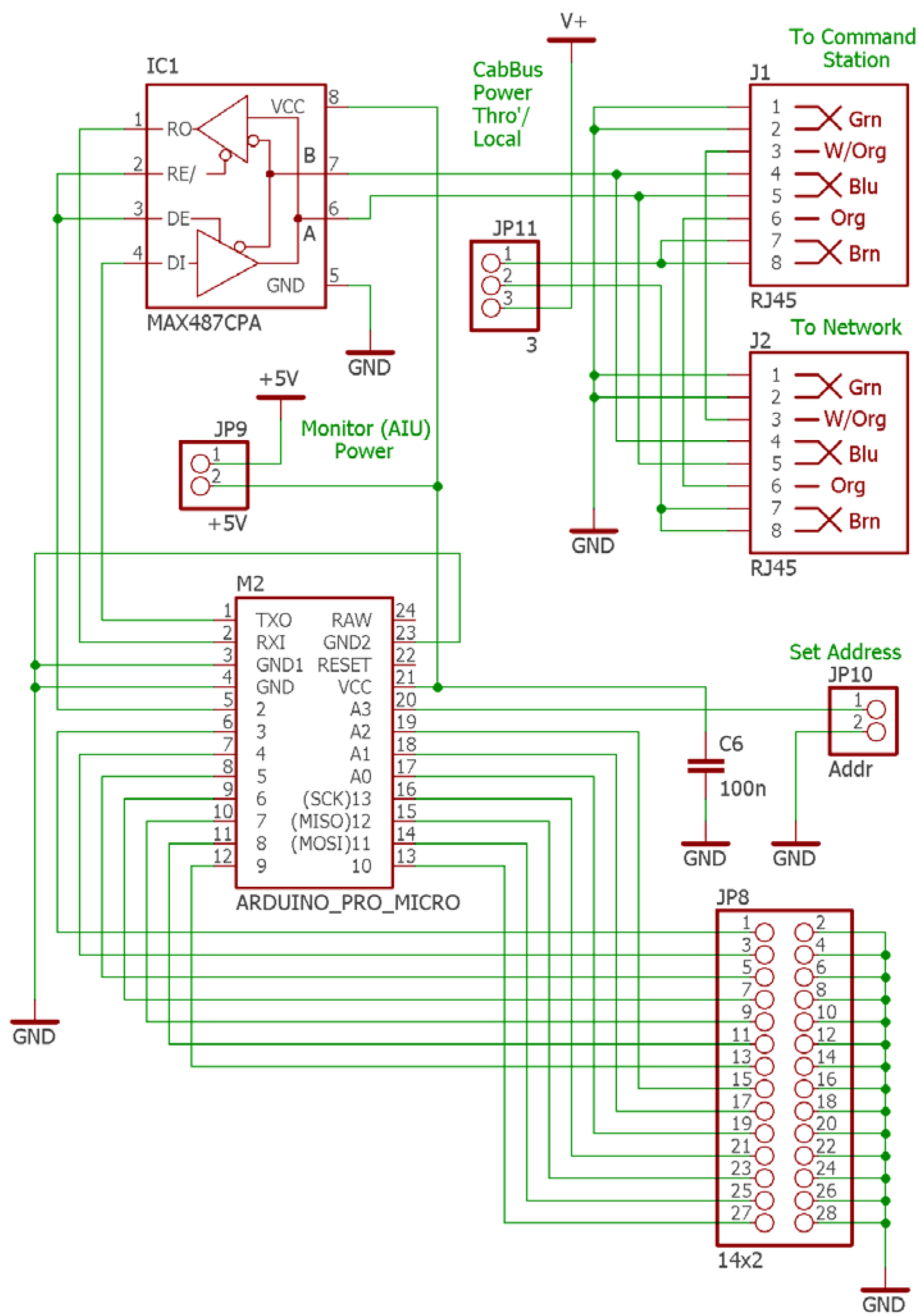
The MAX487 tranceiver receives commands from the NCE Cab Bus which is looped through the Monitor section via connectors J1 and J2 using standard Cat5/6 Ethernet patch cables. The commands are passed to the Arduino Pro-Micro module pin RX1, decoded by the uploaded code, and the appropriate response returned from the Pro-Micro pin TX0 back to the NCE Cab Bus via the MAX487 tranceiver. The output from pin D2 of the Pro-Micro enables the tranceiver to transmit the response over the Cab Bus at the correct time.

Note that the USB interface of the Pro-Micro (not shown on the schematic) is completely independent of the TX0/RX1 serial link and can, therefore, be used to monitor operation of the Pro-Micro at any time without interfering with NCE Cab Bus data traffic.

The +12 volt supply carried by the NCE Cab Bus, to power handheld controllers or other devices plugged into the bus, is normally passed through from J1 to J2 (pins 7 and 8) via a shorting link (Cab Bus Power Select) placed on pins 1 and 2 of header JP11. The link can be changed to short pins 2 and 3 if you wish to power downstream Cab Bus devices from the QSDM. In this case the power supply connected to the QSDM Power-In terminal block X1-1 and X1-2 must be 13.5 volts in order to compensate for the voltage drop across bridge rectifier BR1 (see the QSDM Decoder section schematic) and provide +12 volts out to the Cab Bus. This, in turn, will increase the power dissipated in voltage regulator VR1 which should, therefore, be fitted with a larger heatsink.

Pins D3 to D13 together with pins A0 to A2 are normally configured, with internal pull-up resistors, as the 14 sensor inputs to the QSDM Monitor provided by the 14x2 pin header JP8.

Pin A3 is also configured as an input to put the Monitor into Set Cab Bus Address mode when a shorting link (Set Monitor Address) is placed on header JP10. When this is done, pin A1 is reconfigured as an output to drive the LED fitted to the Set Address module, which will be plugged on to the first 10 pairs of pins of header JP8 (see below).



Quad Servo Decoder-Monitor – Monitor Section

QSDM – Set Address Module

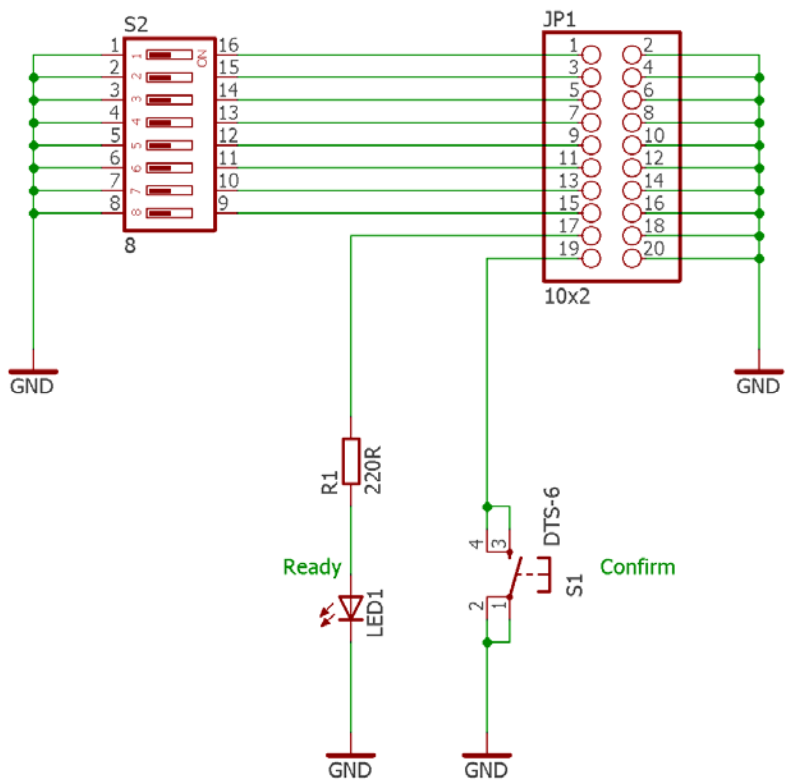
When the QSDM Monitor is set into Set Cab Bus Address mode by fitting a shorting link (Set Monitor Address) to header JP10, and the 10x2 Set Address module header socket JP1 is fitted to the first 10 pin pairs of header JP8, LED1 (labelled “Ready”) will light.

The required Cab Bus address is set as a binary value on switches 1 to 6 of DIP module S2 (switch ON = 1, OFF = 0), and switches 7 and 8 are set to configure how the QSDM Monitor will use the subsidiary Status View module (or not).

If either switch 7 or 8 are set ON, pins D10 to D13 together with pin A0 of the Pro-Micro module will be configured as outputs instead of inputs (pin pairs 10 to 14 of Monitor header JP8), ready to drive the LEDs of the Status View module. If switch 7 is set ON (regardless of the state of switch 8), the five output pins will reflect the state of input pins D3 to D7 (pin pairs 1 to 5 of Monitor header JP8) or, if switch 8 is set ON, the five output pins will reflect the state of input pins D7 to D9 plus A1 and A2 (pin pairs 5 to 9 of Monitor header JP8).

With the address and Status View options set, pressing switch S1 will signal the Pro-Micro code to accept the settings, and LED1 will be extinguished as an acknowledgment. As described in Part 2 of the article, status messages will also be output on the USB interface at this time, and can be viewed using the Arduino Serial Monitor.

Note : It is **essential** that the QSDM is **powered off after programming** (and removal of the Set Monitor Address unit) – the newly entered values will not become effective until the QSDM is restarted.

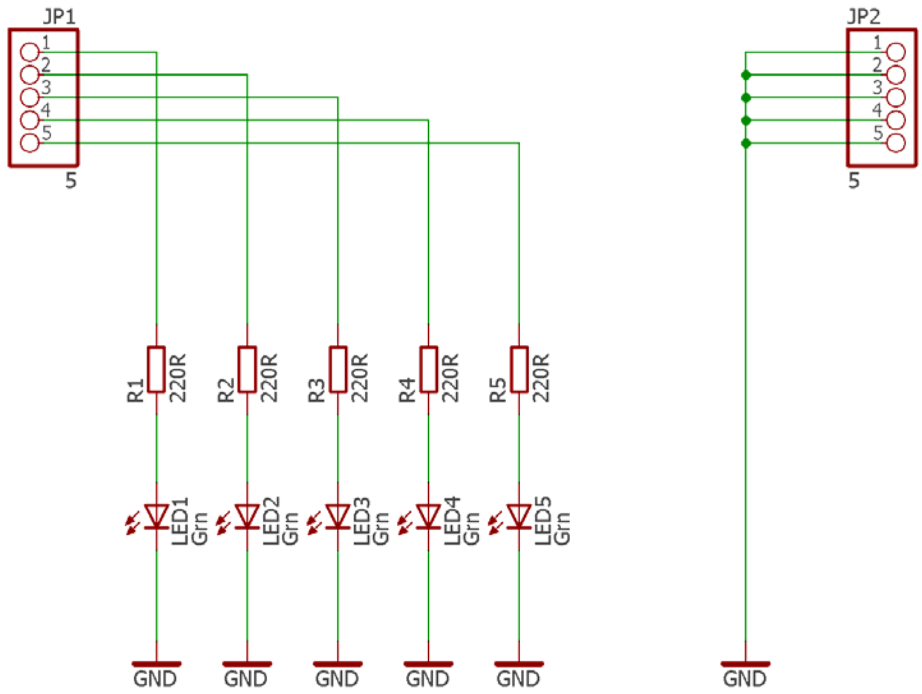


Quad Servo Decoder-Monitor – Set Address

QSDM – Status View Module

This is a simple panel with five LEDs connected between the input pins of header socket JP1 and the corresponding pins of header socket JP2. The latter set of pins will all be connected to ground (0 volts) when the module is plugged on to pin pairs 10 to 14 of the QSDM header JP8.

The LEDs then indicate the state of the set of Monitor input pins selected by the setting of switches 7 and 8 of the DIP module S2 on the Set Address module. An LED will be lit when the corresponding Monitor input pin is connected to ground by an attached sensor.



Quad Servo Decoder-Monitor – Status View

RJ12-RJ45 Cab Bus Network Panel

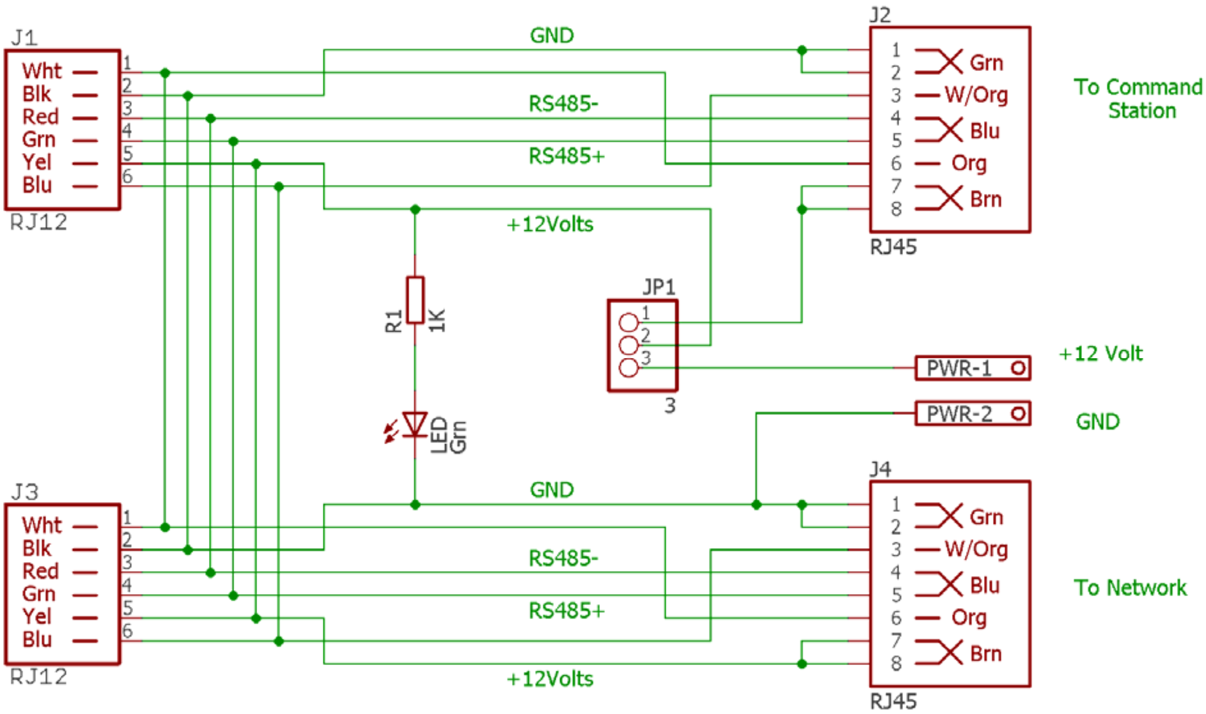
The panel is intended to allow an NCE Cab Bus implemented using standard Cat5/6 Ethernet patch cables to provide intermediate connection points to the Bus by standard NCE devices such as handheld controllers.

The Command Station is connected either directly or via other intermediate panels to socket J2 (RJ-45) which provides a straight-through connection of the standard NCE 6-wire Cab Bus to the other three sockets J1, J3 (RJ-12) and J4 (RJ-45).

If required for long cable runs, an auxiliary +12 volts supply can be injected via terminal block PWR-1 and PWR-2 to supply power to the three downstream sockets J1, J3 and J4.

The LED provides an indication that power is currently connected to the panel.

Copyright © 2021-2022 JT Chamberlain, A-Train Systems



RJ12-RJ45 Panel

Copyright © 2021-2022 JT Chamberlain, A-Train Systems