Quad Servo Decoder-Monitor – Part 1
A DCC Accessory Decoder with Feedback

Introduction

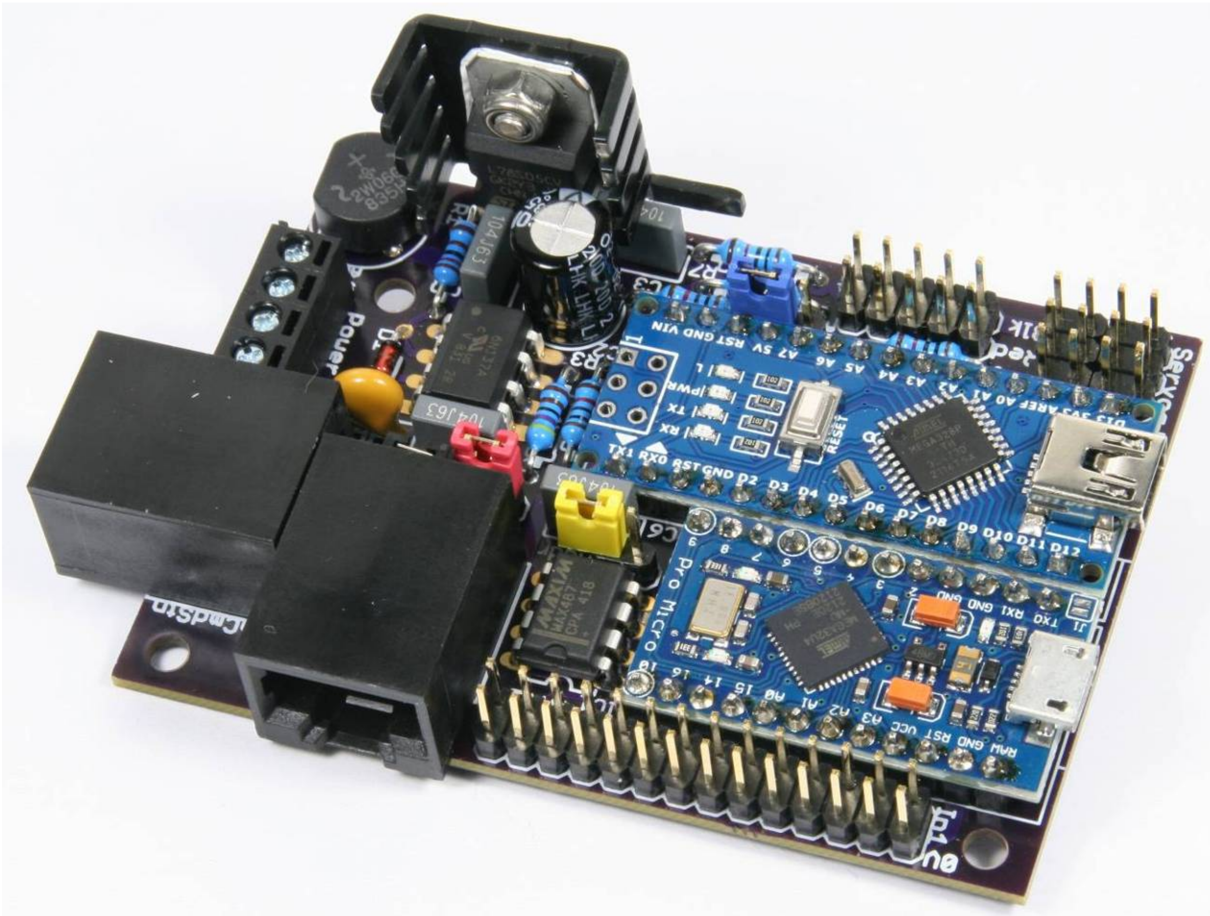
Some readers may remember (and possibly have built) my Quad Servo DCC Decoder (QSDD), as published in *Model Railroad Hobbyist February (online.fliphtml5.com/buups/wpwb/index.html#p=49) and March (online.fliphtml5.com/buups/hfkw/index.html#p=51) 2020*, together with a couple of updates on the MRH Forum (forum.mrhmag.com/post/quad-servo-dcc-decoder-%E2%80%93-improved-version-12217939 and forum.mrhmag.com/post/quad-servo-dcc-decoder-%E2%80%93-arduino-software-update-12218580). On your DCC layout, this little unit acts as a standard accessory decoder which gives you control over four servos, each of which can switch a turnout. In set-up mode, the QSDD also provides precise control over the throw and rate of movement of each servo arm (and hence of the linked turnout) – which was the main reason for designing and building the QSDD in the first place. A short video is available (https://youtu.be/Ox-X1uAWssc) which shows the straightforward set-up process.

My brother Derek has quite a large layout, operated using an NCE DCC system, and he was keen to use a set of QSDDs to control around 80 turnouts. However, he also wanted to get feedback from each turnout to check that a sent DCC switch command had been acted upon, and that the current position of each turnout could be shown on an indicator panel or mimic diagram on a computer screen. He also has ambitions to fit block occupancy detectors to all sections of the layout track to let him see, at his central location, where all his locomotives and rolling stock are at any given time.

While he could have implemented this feedback monitoring function by purchasing a set of NCE Auxiliary Input Units (AIUs), each of which could handle feedback from up to 14 individual turnouts or occupancy detectors, he was somewhat discouraged by the potential cost of the commercial kit, so turned to me for a cheaper DIY solution to match the proposed set of QSDDs.

As it happened, I had come across the Arduino NceCabBus library developed by Alex Shepherd. One of his example applications was for an AIU so, using this as a starting point, I thought I could have a go at producing a cheaper and neater version of an AIU. Initially, I had hoped simply to expand the functionality of the Arduino Nano module which is at the heart of the existing QSDD design but I discovered that, for various technical reasons, the Nano is unable to meet all of the AIU requirements, and that an alternative type of Arduino module, the Pro-Micro, is required.

Although I could have built the AIU look-alike as a completely separate unit to monitor and report all of the feedback status, a tidier solution was to expand the size of the QSDD printed-circuit board (PCB) to incorporate the additional Pro-Micro module, and have only one unit instead of two to fit and connect up. The result, the Quad Servo Decoder-Monitor unit, is shown below [1] –



All photographs and diagrams by the author

1. Quad Servo Decoder-Monitor – two modules on one printed-circuit board

The two sections making up the Quad Servo Decoder-Monitor (QSDM) are functionally independent, and share only the on-board 5-volt voltage regulator. The QSDD section, using an Arduino Nano module and furthest from the camera, operates in exactly the same way, and has the same set-up procedure, as the original standalone unit, driving up to four servos in response to commands received over the DCC (track) bus.

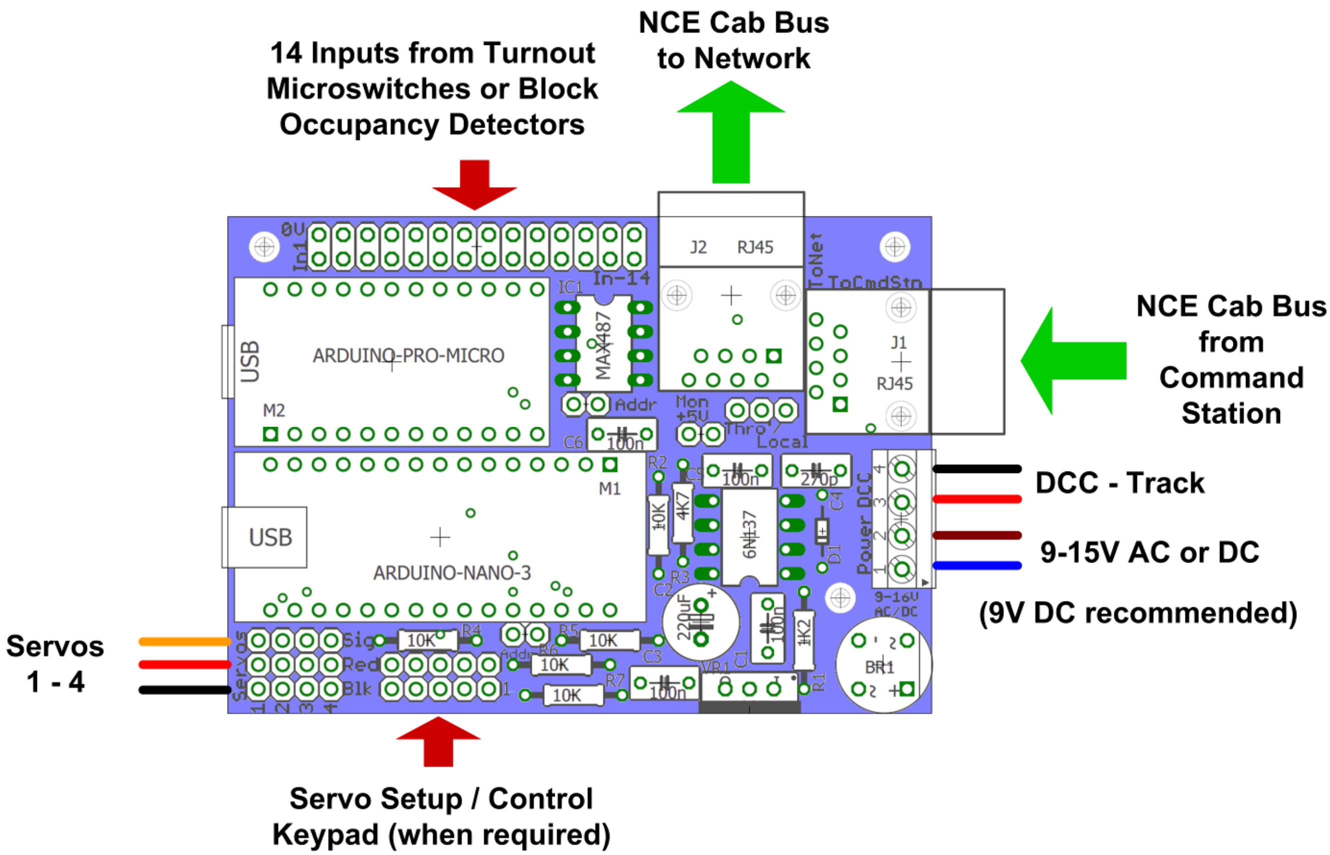
In the foreground is the Monitor section, based around an Arduino Pro-Micro module (a cutdown version of the Arduino Leonardo, but **not** to be confused with the Pro-Mini module which is based on a completely different processor). The Pro-Micro can use its serial port independently of the USB interface, unlike the Nano, to transfer the state of up to 14 inputs (from turnout switches or block occupancy detectors) wired to the 14x2 pin header directly to the NCE Command Station's Cab Bus. The Cab Bus connection is implemented using cheap and available Ethernet Cat5/Cat6 cables with 8-way RJ45 connectors rather than the normal – and more expensive – NCE 6-way RJ12 cables.

Because the two QSDM sections operate independently, it would be quite feasible, if you have already built one or more of the original QSDDs, simply to fit only the Monitor section components to the expanded PCB (with the voltage regulator, of course) and just use the unit as a direct substitute for an NCE AIU. However, for anyone who would like to follow this route, an alternative, smaller PCB is available to hold just the Monitor section. More information on this option can be found later in the article where there are details of how to go about building your own QSDM.

Fairly obviously, control over the functions of both QSDM sections is exercised through an attached computer application. You can use either the JMRI suite, which runs on a variety of computer systems, or my own A-Track application running on a Windows PC, both with a USB connection to either an NCE Interface Unit for Power Cab systems, or a USB-to-Serial converter for Power Pro systems. Note, however, that while JMRI works fine with a Power Pro command station, it appears to be unable to handle feedback from the QSDM Monitor section (or from standard NCE AIUs) when using even the most up-to-date Power Cab system. I cannot claim to be any form of JMRI expert so, if anyone would like to tell me differently, I will be happy to be corrected.

Connections

All of the connections to the QSDM are shown in the diagram below [2] –



2. Quad Servo Decoder-Monitor – layout connections

The complete QSDM can be powered either from the DCC track or, preferably, from a separate 9 to 15-volt AC or DC supply capable of providing up to 1 amp (or better, 2 amps, if a number of servos are to be operated simultaneously). When operating the QSDM it is best to use the minimum external supply voltage (9V AC or DC), reducing the power dissipation in the on-board voltage regulator, and keeping its heatsink as cool as possible.

Power is supplied to the QSDM, and hence to the attached servos, through terminals 1 and 2 of the 4-way terminal block. A connection is also taken from the track DCC supply to terminals 3 and 4 of the terminal block. This connection supplies commands to the Decoder (QSDD) section of the QSDM but, in normal operation, does not supply any substantial power to the module.

If you wish to operate everything from the DCC bus then you can connect terminals 1 and 3, and 2 and 4, together. However, the higher DCC input voltage will substantially increase the power dissipated in the voltage regulator, so that a larger heatsink would be strongly recommended.

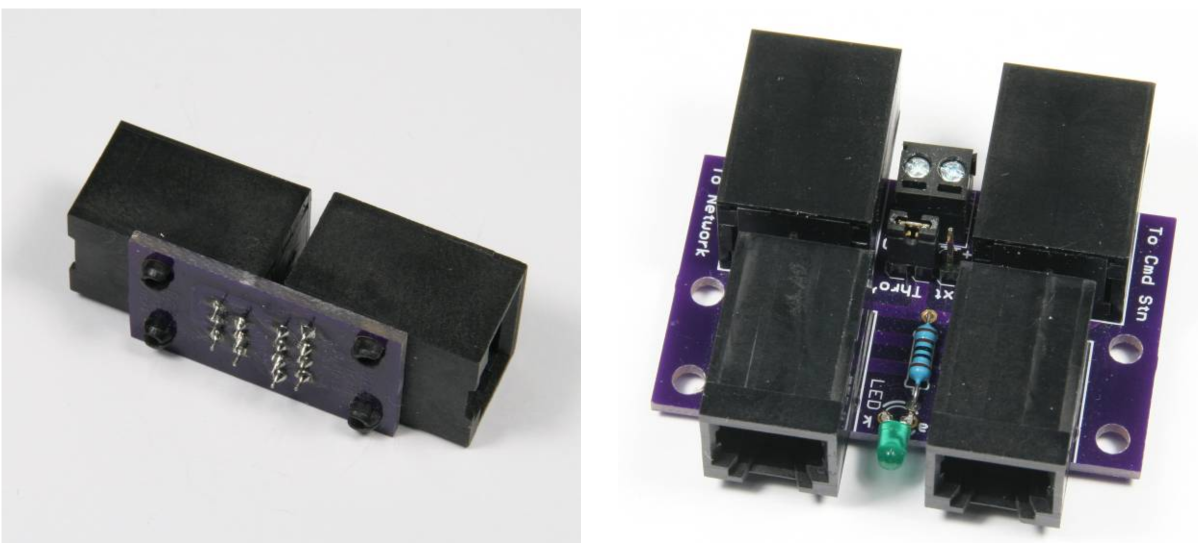
The Monitor section of the QSDM is connected to the Command Station Cab Bus via one of the 8-way RJ45 sockets (J1) using a standard Ethernet cable, and the Cab Bus is continued in daisy-chain fashion to the next QSDM (or AIU or other NCE device) from the second RJ45 socket (J2). Connections to the Cab Bus are only made after the QSDM software and address has been set up as described in Part 2 this article.

Cab Bus network

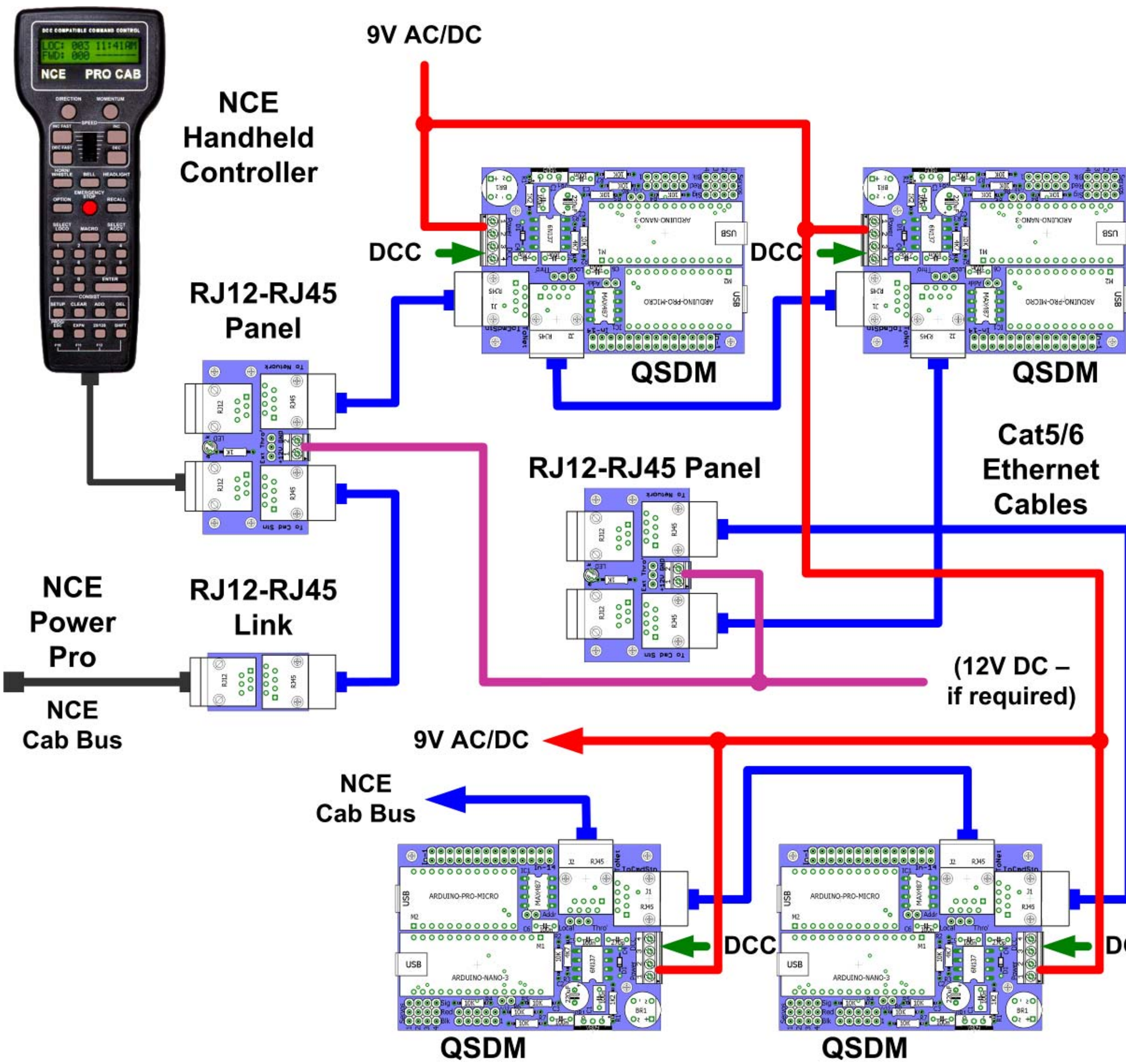
The NCE Cab Bus connects the Command Station to all attached Handheld Controllers and other NCE devices (including AIUs), and so is used by the QSDM to return status information regarding turnout positions and block occupancy (where implemented) to the Command Station. Whatever computer application you are using can then fetch the status data from the Command Station.

The standard Cab Bus uses 6-way RJ12 connectors but here, with the QSDM, in order to use cheap and widely-available Cat5/Cat6 Ethernet cables for the Cab Bus Network, it is converted to use 8-way RJ45 connectors. As well as digital information, the Cab Bus also carries power, at a nominal 12 Volts DC, from the NCE Command Station to supply any Handheld Controllers or AIUs that are attached to the Bus.

To link to the Ethernet cables, I created a couple of simple PCBs, the RJ12-RJ45 Link which provides a straight-through connection from an RJ12 socket to an RJ45 socket, and the slightly larger RJ12-RJ45 Panel which interconnects two RJ12 and two RJ45 sockets. Both units are shown in [3] below, together with diagram [4] which shows an example of one way the various units could be connected together in a functioning system –



3. RJ12-RJ45 Link and RJ12-RJ45 Panel



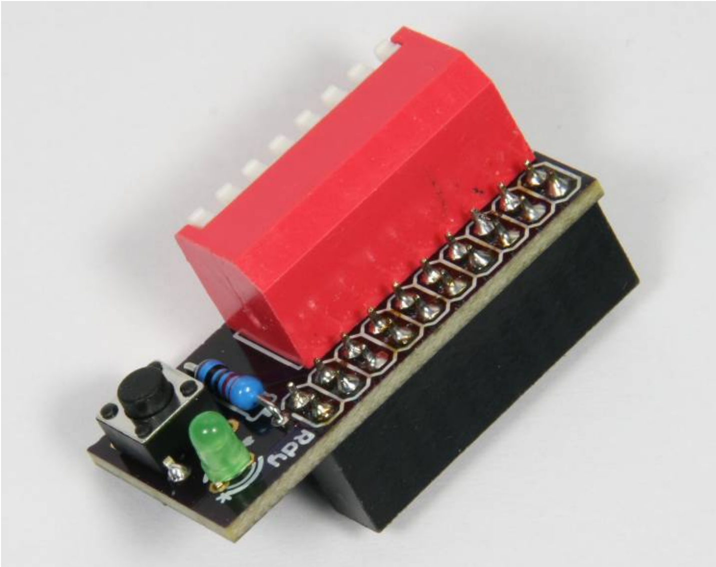
4. Possible configuration for a Cab Bus network

The Cab Bus daisy-chain can be extended to incorporate as many attachments as required, bearing in mind that the ultimate limit is 62 devices, since each attached device must have a unique address, and can stretch as far as 1000 feet.

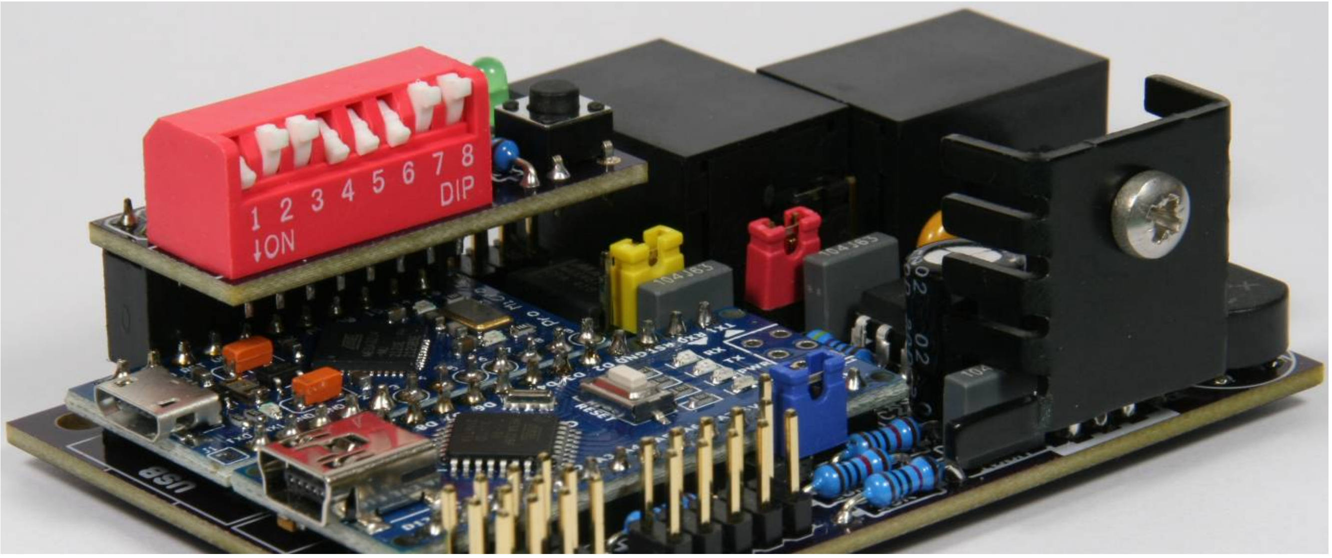
However, very long cables will result in excessive voltage drops which may prevent Handheld Controllers plugged into the end of the cable run from working. This does not affect the QSDMs since they will be powered from their own independent supply but, for cable runs in excess of 40 to 50 feet (12 to 15 metres), it is recommended that 12V DC auxiliary power units are connected to the Cab Bus, as shown in [4] above, to keep any Handheld Controller at this distance operational. However, **note carefully** that this scheme only works with the RJ12-RJ45 Panel if the NCE Command Station is connected to the left RJ45 socket as shown above – auxiliary power routing will not work if the Command Station is connected to any of the other three sockets on the Panel. If auxiliary power is not required then the Command Station can be connected to any socket.

Assigning a Cab Bus address

As noted above, the Monitor section of each QSDM has to have a unique NCE Cab Bus address. The NCE AIU has a package (DIP) of eight dual-inline switches fitted to its PCB for this purpose but, since the switches are only used very occasionally, and to keep the size of the QSDM as small as possible, I decided to create an additional small module which could be plugged in only when required, as shown in [5] and [6] below –



5. Plug-in module to set Monitor Address



6. Set Address module plugged on to the QSDM 14x2 header

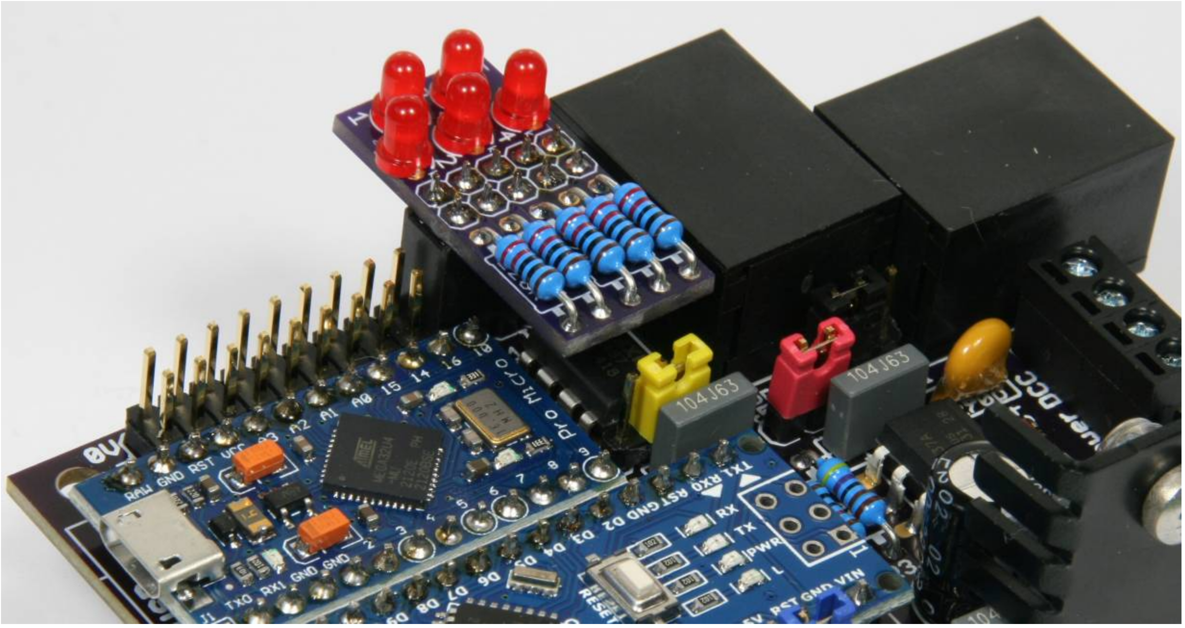
The Set Address module is fitted with a 10x2 header plug which fits on to the first 10 pairs of pins of the QSDM 14x2 header. The desired Cab Bus address for the Monitor section is entered as a binary number using switches 1 to 6 (just like the NCE AIU). Full details on how to set up and use the Set Address module are given in Part 2 of this article.

An optional QSDM feature

Unlike the commercial NCE AIU, the QSDM does not have a set of 14 LEDs which clearly show the present state of the inputs to the device. Such LEDs are very helpful in providing an immediate check that the wiring from a particular layout element to the AIU input is correct.

Unfortunately, the Arduino Pro-Micro module used in the QSDM Monitor section does not have enough pins to drive an LED for each of its inputs, but its operation can be modified to provide a partial solution which can assist when initially setting up connections to the QSDM Monitor inputs.

The 5 pairs of pins of the QSDM 14x2 header nearest to the RJ-45 socket can be programmed as outputs which can each drive an LED, and a small module (Status View) with five LEDs can be plugged on to these pins, as shown in [7] below –



7. Status View module plugged on to the QSDM header pins

The option is controlled by the setting of switches 7 and 8 of the Set Address module when programming the Monitor Cab Bus address, to allow you to display the state of either the leftmost or the rightmost 5 pairs of header pins selected from the remaining 9 pairs.

Hence, assuming we have a sensor such as a microswitch on a turnout servo, or a block occupancy detector, connected to one of the selected group of header pins, the state of the sensor will be indicated immediately by whether the corresponding LED on the Status View module lights up or not (when power is applied to the QSDM, of course).

Once you are sure that all of your sensor connections are sound, you can simply unplug the Status View module, ready for use on the next QSDM. Again, full details of how to set up this optional feature are given in Part 2 of this article.

Interested in building your own QSDM ?

(Note that the details which follow were omitted from the published article in MRH, and failed to be added to the related Bonus Downloads).

If you would like to build your own QSDM (or set of QSDMs), together with the associated modules, the first step is to purchase a set of printed circuit boards. Note that, as well as the boards described so far, you will also need a PCB to build a small keypad to set up the servo decoder section, as described in the original article on the QSDD (online.fliphtml5.com/buups/wpwb/index.html#p=49). If you have previously built a QSDD then you can, of course, just use the Keypad you already have.

All of the PCBs are available from OSH Park, a small company located in Lake Oswego, Oregon. They supply PCBs in multiples of three boards, with the cost based solely on the area of the PCB, including free shipping to any destination worldwide. You can see what OSH Park has to offer by following these links –

Quad-Servo_Decoder+Monitor	$29.75	oshpark.com/shared_projects/wSyQ3v3Y
QSDM-SetAddress	$4.55	oshpark.com/shared_projects/HtSfqW6E
QSDM-StatusView	$2.60	oshpark.com/shared_projects/s7i7xqR1
QuadServo-Keypad	$5.25	oshpark.com/shared_projects/7ATX5aqB
RJ12-RJ45-Panel	$10.95	oshpark.com/shared_projects/Y8NySuSt
RJ12-RJ45-Link	$3.65	oshpark.com/shared_projects/gjSHhLM6

A complete set of PCBs (three of each) will cost $56.75, or $51.50 if you already possess a QuadServo-Keypad.

On the OSH Park website, if you then want to order a set of PCBs (in multiples of three), click the button labelled “Order Board” next to each board, enter your e-mail address, name, and a password of your choice to establish an account with OSH Park, then follow their ordering process. You can pay either with a credit card or via PayPal. Your boards will be manufactured and delivered within two or three weeks depending on where you are in the world.

If you prefer to use an alternative PCB supplier then, instead of clicking “Order Board”, just click on “Download” to download a copy of the relevant file in Eagle board (.brd) format which you can then send off to your preferred supplier.

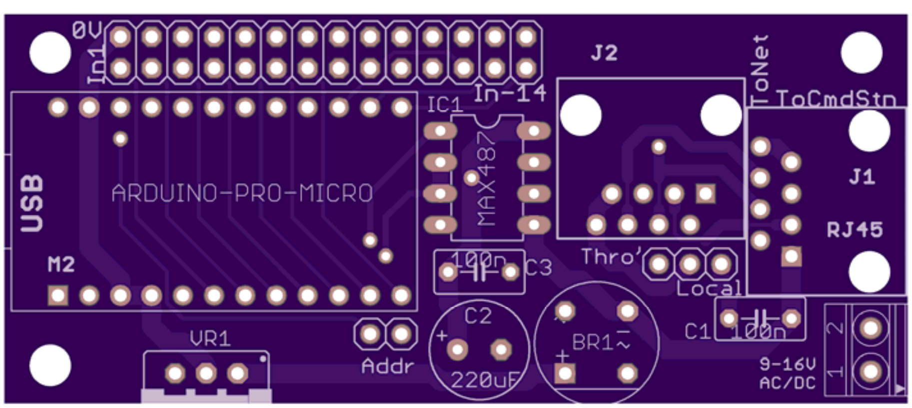
Please note that neither A-Train Systems nor myself have any connection with OSH Park other than as a very satisfied customer of their services.

Note: As suggested in the introduction to the article, if you have already built a Quad Servo DCC Decoder from my previous articles, or if you are not using, nor intending to use servos to drive the turnouts on your layout, but would still like to use the Monitor section to get feedback from block detectors or other sensors, then you can simply build the QSDM without the Decoder section, which will then provide you with a substitute for an NCE Auxiliary Input Unit.

Alternatively, if you would like to use a smaller (and cheaper) dedicated PCB to build the Monitor section on its own, then a set of three boards is available from OSH Park via the link –

Layout-Input-Monitor	$18.10	oshpark.com/shared_projects/4eQ2rG2w

An outline of this board is shown in [8] below –



8. Layout-Input-Monitor board

If just building the Monitor section on the QSDM PCB, you can omit the Decoder section components as listed in tables [9] and [10]. These components are the Arduino Nano-3 module (M1), the 6N137 optoisolator (OK1), diode D1, resistors R1 to R7, capacitors C4 and C5, and the 4x3, 5x2 and 2x1 headers next to the Nano-3. There is also no need to fit the second 2-way terminal block for the DCC connection since this is not used by the Monitor.

Additionally, if you are using the Layout-Input-Monitor PCB, you can also omit capacitor C6 and the 2x1 header next to RJ45-8 socket J2, although these are still required on the larger QSDM PCB.

The parts required to build a complete QSDM (decoder-monitor plus keypad, including cable), and the additional units described, are listed in the two tables, [9] and [10], below.

Part - Decoder-Monitor	**Reference**	**Quantity**	**Value**
Diode Bridge Rectifier	BR1	1	W01G
Capacitor - Polyester	C1, C3, C5, C6	3	100nF
" - Electrolytic	C2	1	220uF 35V
" - Disc Ceramic	C4	1	270pF
Resistor – Metal Film, 0.25 Watt	R1	1	1K2
" "	R2, R4 - R7	5	10K
" "	R3	1	4K7
Diode	D1	1	1N4148
Voltage Regulator - 5 Volt	VR1	1	LM7805
Optoisolator	OK1	1	6N137
Arduino Module	M1	1	Nano-3
Arduino Module	M2	1	Pro-Micro
RS-485 Transceiver	IC1	1	MAX487CPA
Socket	J1, J2	2	RJ45-8
Terminal Block - 3.5mm pitch	DCC+Power In	2	1 x 2
Pin Header - 0.1" (2.54mm) pitch	JP1 – JP4	4	1 x 3
"	JP5, JP6	2	1 x 5
"	JP7	1	1 x 2
"	JP8	1	2 x 14
"	JP9	1	1 x 2
"	JP10	1	1 x 2
"	JP11	1	1 x 3
Jumper Link Open - 0.1" pitch	-	4	
Heatsink – to fit TO220 package + a small amount of heatsink paste	-	1	
Cat5/6 Ethernet Patch Cable	-	As reqd	
USB Cable - A Plug to Mini-B Plug	(Nano-3)	1	
USB Cable - A Plug to Micro-B Plug	(Pro-Micro)	1	

9. Table 1 – Quad Servo Decoder Monitor – List of components

Part - Keypad	**Reference**	**Quantity**	**Value**
Resistor – Metal Film, 0.25 Watt	R1 – R5	5	220R
Light-Emitting Diode - Red	LED1	1	3mm Red
Light-Emitting Diode - Green	LED2 – LED5	4	3mm Green
Tactile Switch	S1 – S4	4	6mm
Socket Header - 0.1" (2.54mm) pitch	JP1	1	2 x 5
IDC Socket Ribbon Cable Connector		1	10-way (2 x 5)
IDC Plug Ribbon Cable Connector		1	Box Header (2 x 5)
Ribbon Cable – 0.05" (1.27mm) pitch		As reqd	10-way
Part - Set Address	**Reference**	**Quantity**	**Value**
Resistor – Metal Film, 0.25 Watt	R1	1	220R
Light-Emitting Diode - Green	LED1	1	3mm Green
Tactile Switch	S1	1	6mm
DIP Switch	S2	1	8-way
Socket Header - 0.1" (2.54mm) pitch	JP1	1	2 x 10
Part - Status View	**Reference**	**Quantity**	**Value**
Resistor – Metal Film, 0.25 Watt	R1 – R5	5	220R
Light-Emitting Diode - Red	LED1 – LED5	5	3mm Red
Socket Header - 0.1" (2.54mm) pitch	JP1	1	2 x 5
Part - RJ12-RJ45 Panel	**Reference**	**Quantity**	**Value**
Resistor – Metal Film, 0.25 Watt	R1	1	1K
Light-Emitting Diode - Green	LED1	1	3mm Green
Socket RJ12	J1, J3	2	RJ11-6
Socket RJ45	J2, J4	2	RJ45-8
Terminal Block - 3.5mm pitch	+12V Power In	1	1 x 2
Pin Header - 0.1" (2.54mm) pitch	JP1	1	1 x 3
Part - RJ12-RJ45 Link	**Reference**	**Quantity**	**Value**
Socket RJ12	J1	1	RJ11-6
Socket RJ45	J2	1	RJ45-8

10. Table 2 – QSDM subsidiary units – List of components

Suggested suppliers for the parts listed above are RS Components (uk.rs-online.com/web/) or Farnell (uk.farnell.com/) for users in the UK, or Newark (www.newark.com/) for users in the USA (part of the same company as Farnell). Mouser (www.mouser.com/) or Digikey (www.digikey.com/) are alternative sources in the USA, although their prices tend to be a little higher than Newark. Both Mouser and Digikey also have European-based operations, but still tend to have higher prices than RS Components or Farnell.

The Arduino Nano-3 and Pro-Micro modules can best be obtained from one of the many suppliers operating on eBay, although you will have to make your own judgement as regards who will give you the best service, based on their feedback from previous customers. An alternative is to source the parts directly from China, using a website such as AliExpress, where you will benefit from substantially lower prices, but will again have to make a judgement on the reliability of a particular supplier based on their published customer feedback.

The tables below, [11] and [12], give suggested part numbers for each QSDM component from each of the listed suppliers. Click on the part number to view the relevant webpage with details of the part or, if the direct link does not work, copy the part number and paste it into the search box on the relevant supplier's website –

Ref-Decoder	**RS Cmps**	**Farnell**	**Newark**	**Mouser**	**Digikey**
BR1	7082668	2675385	99AC4581	625-W01G-E4	B250C1000G-E4/51GI-ND
C1, C3, C5-6	3121469	2429342	18AC7634	80-R82DC3100AA50J	399-19335-ND
C2	7111264	8126690	62W6211	80-ESK227M035AG3AA	P5166-ND
C4	7167226	2860060	57AC2084	594-S271K43SL0N6TK5R	BC2679CT-ND
R1	1650230	9341226	95W7689	71-CCF071K20GKE36	S1.2KCACT-ND
R2, R4 - R7	1651031	9341110	95W7695	71-CCF0710K0JKE36	S10KCACT-ND
R3	1650319	9341951	95W7764	71-CCF074K70GKE36	S4.7KCACT-ND
D1	7390290	2675146	05AC0533	512-1N4148	1N4148FSCT-ND
VR1	7931346	1467758	72K6018	511-L7805CV	497-1443-5-ND
OK1	8051267	2453244	31Y6274	859-6N137M	160-1791-ND
IC1	1900831	2518622	81Y9489	700-MAX487CPA	MAX487CPA+-ND
M1	eBay	eBay	eBay	eBay	eBay
M2	eBay	eBay	eBay	eBay	eBay
DCC+Power	8971332	3882615	68C9065	651-1985807	277-6043-ND
JP1 – JP4	2518632	1593422	08N6754	517-929834-01-24-RK	3M9457-24-ND
JP5, JP6	"	"	"	"	"
JP7 – JP11	"	"	"	"	"
Jumper	2518682	3226076	47AC9509	855-M7583-46	S9337-ND
RJ45 – J1, J2	2400935	1137974	56AC3196	530-SS-90000-001	277-1149868-ND
Heatsink	7124257	1611415	81F046	532-507302B00	HS115-ND
Cat5/6 Cables	eBay	eBay	eBay	eBay	eBay
USB Cables	eBay	eBay	eBay	eBay	eBay
Ref-Keypad	**RS Cmps**	**Farnell**	**Newark**	**Mouser**	**Digikey**
R1 – R5	1650814	9341528	95W7736	71-CCF07220RJKE36	S220CACT-ND
LED1	1780909	1581111	14N9386	859-LTL-4211N	160-1139-ND
LED2 – LED5	1808502	1581114	14N9374	859-LTL-4231N-1	160-1958-ND
S1 – S4	3786476	1555982	95M4260	688-SKHHAM	450-1650-ND
JP1	2518222	1593490	08N6807	855-M20-7830546	S7108-ND
IDC Socket	8323483	2215247	45W6459	517-D89110-0131HK	732-2102-ND
IDC Plug	6741205	4139045	94F7977	710-61201025821	732-5452-ND
Ribbon Cable	eBay	eBay	eBay	eBay	eBay

11. Table 3 – QSDM and Keypad – Component supplier references

Ref-SetAddress	**RS Cmps**	**Farnell**	**Newark**	**Mouser**	**Digikey**
R1	1650814	9341528	95W7736	71-CCF07220RJKE36	S220CACT-ND
LED1	1808502	1581114	14N9374	859-LTL-4231N-1	160-1958-ND
S1	3786476	1555982	95M4260	688-SKHHAM	450-1650-ND
S2	1748287	2864304	60AJ4795	490-DS02C-254-1L08BE	2223-DS02C-254-1L-08BE-ND
JP1	2518244	1593494	08N6810	855-M20-7831046	S7078-ND
Ref-StatusView	**RS Cmps**	**Farnell**	**Newark**	**Mouser**	**Digikey**
R1 – R5	1650814	9341528	95W7736	71-CCF07220RJKE36	S220CACT-ND
LED1 – LED5	1780909	1581111	14N9386	859-LTL-4211N	160-1139-ND
JP1	2518222	1593490	08N6807	855-M20-7830546	S7108-ND
Ref-RJ12-45 Panel	**RS Cmps**	**Farnell**	**Newark**	**Mouser**	**Digikey**
R1	6832768	9341102	95W7687	279-LR0204F1K0	S1KCACT-ND
LED1	1808502	1581114	14N9374	859-LTL-4231N-1	160-1958-ND
J1, J3	7350282	1137973	56AC3195	530-SS-90000-003	380-1043-ND
J2, J4	2400935	1137974	56AC3196	530-SS-90000-001	277-1149868-ND
Term Block	8971332	3882615	68C9065	651-1985807	277-6043-ND
JP1	2518632	1593422	08N6754	517-929834-01-24-RK	3M9457-24-ND
Ref-RJ12-45 Link	**RS Cmps**	**Farnell**	**Newark**	**Mouser**	**Digikey**
J1	7350282	1137973	56AC3195	530-SS-90000-003	380-1043-ND
J2	2400935	1137974	56AC3196	530-SS-90000-001	277-1149868-ND

12. Table 4 – QSDM subsidiary units – Component supplier references

Notes :

1. At the time of writing it is difficult to give you a firm estimate for the cost of the parts listed above because of the current worldwide shortage of electronic components. Available stocks at all suppliers fluctuate rather unpredictably, as do their prices, particularly of semiconductor chips, where cost has often doubled compared to that before the Covid pandemic. However, my best estimate for the various units is given in the summary table [13] below –

Unit	**Components**	**PCB**	**Total**
Quad Servo Decoder-Monitor	$23.40	$9.92	$33.32
Keypad + Extension Cable	$6.85	$1.75	$8.60
Set Address	$2.00	$1.52	$3.52
Status View	$1.30	$0.87	$2.17
RJ12-RJ45 Panel	$4.75	$3.65	$8.40
RJ12-RJ45 Link	$2.10	$1.22	$3.32

13. Table 5 – QSDM units – Estimated costs

– which is, I think, considerably less than the equivalent commercial items.

2. You may be able to source equivalent parts locally at a lower cost, using the details available for each suggested part by clicking on the links above (assuming that you have sufficient electronics knowledge to understand the specifications). Although eBay is a very good source for components, especially for small quantities of passive components, headers, connectors or hardware, you need to be wary of very low cost parts, since these are often of low quality or may be manufacturers' substandard rejects. Similar considerations apply to buying from AliExpress.
3. Buying electronics components singly or in small quantities is much more expensive than buying in bulk (in quantities of 10 or more), so it is well worth considering carefully at the outset how many modules you might build, and then procuring all of the required components in a single purchase. This will also reduce any shipping charges.
4. Although you can buy individual pin headers to match all of the sizes used, it is generally much cheaper to buy a single long strip of at least 24 pins, and carefully cut or snap it into the required sizes. Beware of buying the very cheapest headers on eBay, for example, since the plastic used tends to shatter easily, exposing the end pin and often failing to hold it in position.
5. The 4-way terminal block on the QSDM board is made up from two 2-way blocks since they are generally a cheaper option than purchasing a complete 4-way block.

Once you have acquired your PCBs and a full kit of components, the next step is to assemble the units you have decided to build. 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 module PCBs.

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 (or 50 watts if the iron is temperature-controlled). All joints should be made as quickly as possible to avoid damaging the PCBs and components. The greatest enemy of electronics is heat.

Step-by-step details of the assembly and connection of the various QSDM units will be covered in Part 2 of this article.

Software for the Quad Servo Decoder-Monitor

For anyone who is not familiar with the Arduino hardware and software which form the basis for this project, the article by Geoff Bunza in the December 2016 issue of *MRH* "A modeler's introduction to the Arduino" (mrhpub.com/2016-12-dec/online/html5/index.html?page=132&noflash) provides a very good introduction.

The files provided in Bonus Extras for the December 2016 *MRH* (mrhmag.com/magazine/mrh-2016-12-dec/bonus-extras) include a guide to setting up the Arduino Integrated Development Environment (IDE) on your computer, including links to the Arduino website for downloads (www.arduino.cc/en/Main/Software) and a guide to getting started (www.arduino.cc/en/Guide/HomePage). There are also links to tutorials (create.arduino.cc/projecthub/Arduino_Genuino/getting-started-with-the-arduino-desktop-ide-623be4) and setting up the necessary hardware, including the Arduino Nano (www.arduino.cc/en/Guide/ArduinoNano) which is used here in the QSDM.

Once you have the Arduino IDE installed and set up on your computer, and you have mastered the basics by working through some of the examples supplied, you need to use the Include Library function from the Sketch menu in the IDE to add a couple of extra libraries to the system.

These are the NmraDcc library (www.arduinolibraries.info/libraries/nmra-dcc) and the Bounce2 (www.arduinolibraries.info/libraries/bounce2).

Although Alex Shepherd's NceCabBus Arduino library provided the inspiration for the project, only a few of the constituent routines were required to implement the AIU functions, and I found that the library structure also made it quite difficult to add in some extra necessary functions. In the end, I decided not to use the library, but simply to write my own version of the routines within the Pro-Micro sketch.

You can now download the sketches for the Nano-3 (QuadServo_DCC-Decoder_5-3.ino) and for the Pro-Micro (QuadServo_DecoderMonitor_2-5.ino) from a special download section of my A-Train Systems website (www.a-train-systems.co.uk/qsdm-download) to any convenient folder on your computer, and then open it in the Arduino IDE.

Note that, when loading the sketch for the Pro-Micro, you should set the Board type (from the Tools menu) as 'Arduino AVR Boards/Arduino Leonardo'. The Board type for the Nano-3 is, fairly obviously, 'Arduino Nano', and you will have to select the processor as either 'ATmega328P' or 'ATmega328P (Old Bootloader)' depending on the issue of Nano-3 module you purchased.

Check that the sketches will compile with the included libraries, by clicking the Verify () button on the Arduino IDE toolbar. Any errors are most likely to be caused by the Arduino IDE failing to find the required libraries where it expects them to be, or that the Board, Processor, and Port settings under the Tools menu are not set correctly.

Part 2 of this article will cover all aspects of the necessary software upload, set-up and operation of both Arduino modules when they are connected to your NCE system.

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.