DCC Impulses Column

by Bruce Petrarca

Photos & illustrations by the author, unless otherwise indicated

Some thoughts on programming & tracks.

In prior columns, I've talked around programming tracks. I decided it was time to go straight ahead and talk about them. Part of this was prompted by a recent discussion on an eMail group about programming track isolation and other issues. While a lot of this material has been mentioned before in passing, the focus of this column is to bring all the data into one place. Repetition reminds us of things we may have forgotten. I'll bet that there will be at least one tidbit here for every DCC user who reads this missive.

What is programming?

While there are many things on a DCC layout that can be customized, most frequently, when one is ***programming***, one is setting up the personality of a locomotive by adjusting CV values stored in the decoder. That is the definition of ***programming*** that I'm addressing in this column.

A tale of two worlds

At this juncture, I'm reminded of the joke about people: "There are two kinds of people in this world, those who divide the world into two sections and those who don't."

There are two ways of programming and two kinds of programming track and two ways of programming on-the-main. And just to make things interesting, different manufacturers use different names for the same thing.

One way of programming: on a programming track

Sometimes referred to as ***service mode programming***, programming on the programming track is where you can write to and read from your decoder on a dedicated track.

The command station function of your DCC system (see my column "Anatomy of a DCC System" in the May 2012 issue of MRH) is responsible for controlling the programming track: the command station turns on and off the power to it, monitors the current flow into it, sends commands to it, and receives acknowledgements (ACK pulses) back from it. ACKs are the decoder's way of talking to the command station.

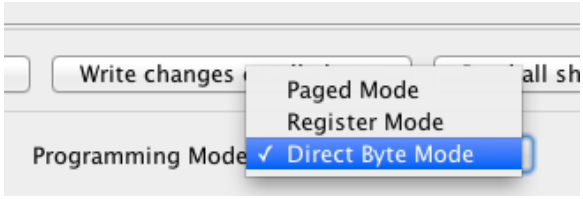
1: The programming track station and selector switch at the PebbleCreek Model Railroad Club

The programming track must be isolated from the remainder of the layout in order for the ACKs to be heard by the command station. The electronics that drives the programming track is ***not*** protected against current flowing from the track back into the command station. This back flow would happen if the programming track were to be connected to the layout DCC signal. The programming track must ***never*** be connected, even one rail at a time, to the layout track or bus. Never is never; not for even a fraction of a second. You may get away with a "5-second" rule when you drop your sandwich, but not here.

There are multiple modes (or languages) that may be utilized in this dialog between the command station and the decoder. Here are some of the modes, with the abbreviation shown on Digitrax' DT4xx series throttles:

- Address Mode is an early programming method in the NMRA standards. It is outdated and modern decoders have no reason to go there.
- Register Mode (Ph in Digitrax parlance) grew out of Address Mode. It is a very limited way to access a few CVs. It is mostly extinct, too. A few older and low end decoders need to be accessed this way.
- Paged Mode (Pg in Digitrax parlance) grew out of Register Mode. It allows access to virtually all CVs. Of the two modern programming modes, Paged Mode is significantly slower than Direct Mode. Per their web site (**http://www.digitrax.com/tsd/KB464/dt402-dt400-programming-mobile-decoder-addresses/**), Paged Mode is "Digitrax Preferred Programming Mode".

- Direct Mode (Pd in Digitrax parlance) has been around for many years now and is measurably faster than any of the older methods. Most modern decoders support it and some respond only to it. While there are two variations of direct mode, the differences are transparent to the DCC user.



2: DecoderPro offers you the choice of programming modes that are compatible with your DCC system and the decoder in question.

So, if you are looking for a common language between your command station and your decoder, start with Direct Mode. If that doesn't fly, move up the list until you find one that does, understanding that Address Mode may not be available on modern command stations.

DecoderPro uses its knowledge of your DCC system and the selected decoder to automatically select the best programming mode for the combination.

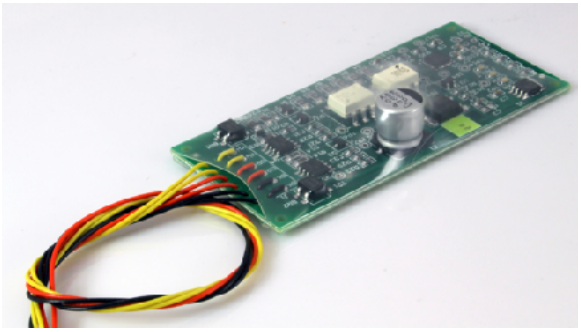
If you are programming a sound decoder or a decoder with large amounts of energy storage, such as a TCS Keep-Alive™, you may need a Programming Track Booster between your command station and your programming track.

Programming Track Boosters

I discuss these guys on my web site. But I want to make a couple of observations here.

PTBs are another misnomer, as they don't really boost anything. They translate from one language to another.

PTBs should not be confused with DCC boosters, despite the similar name. They are very different animals.



3: SoundTraxx PTB-100 Programming Track Booster

As sound decoders developed, more on-board energy storage was required. This added storage would overwhelm the original (NMRA compliant) command stations. The decoder and command station would not communicate. The NMRA standards were changed after 2005 to reflect the new designs and more recent command stations work just fine. However, there are a lot of systems out there that were designed before the change.

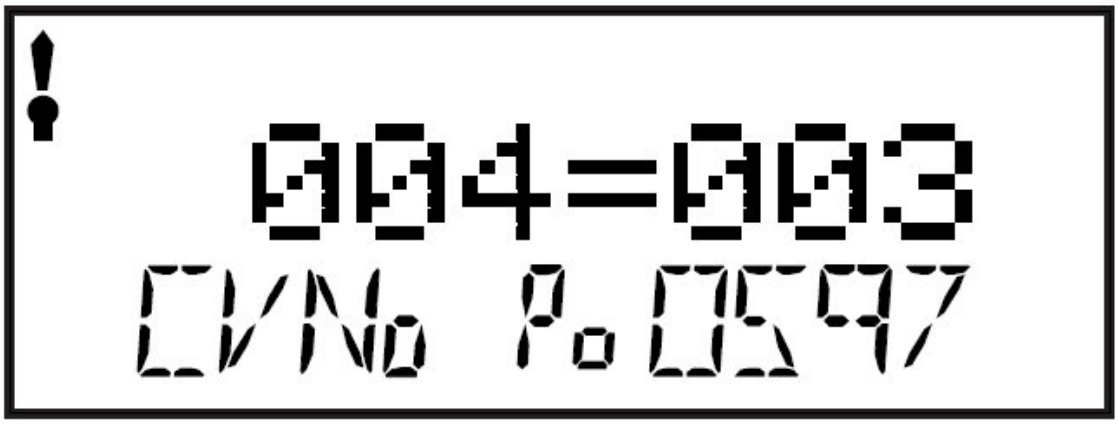
PTBs are the translators that allow easy communication between older command stations and ***all*** decoders. They in essence interpret the current flow through the newer decoders and tell the older command station what is really happening, based on the new NMRA standard.

With my spotty Spanish, I can order a beer and find the restrooms in Mexico. But I cannot communicate easily with native speakers. Similarly, there are "fixes" for sound decoder programming problems recommended on the web. They include adding light bulbs or resistors to the programming track. They are patches. They may or may not work with a specific command station and a specific decoder.

There is nothing about a PTB that prevents it from being used with any decoder of any vintage. So, I recommend, if you are working with decoders that your command station isn't understanding, buy a PTB for about $60 and install it between your command station and your programming track and leave it. The SoundTraxx PTB-100 gets the most consistent "thumbs-up" from users.

Or get a newer system, like a PowerCab or a V3.6 Lenz system.

The other way of programming: on the main



4: Drawing of a DT400 series throttle set for Ops Mode programming. What is shown is programming loco number 597, CV 4 (deceleration) to a value of 3 in Ops Mode (Po). Note that the display is showing you that it is in Ops Mode and what loco it is talking to.

Sometimes referred to as Ops-Mode Programming (Po in Digitrax parlance) or sometimes abbreviated POM,

Programming on the Main allows you to write CVs to locomotives as they are operating on the layout. During a recent op session, I was doing a switching job on our club (**TINY URL for http://www.PCMRC.org**) layout and the loco that I was using was running a bit faster than I wanted at full throttle. So, I went over to our

computer (running JMRI) and opened the file for the culprit. A quick change of the forward and reverse trim settings and I was back operating with a slower loco. The new setting was stored in the decoder on the loco and in the computer as a back up.

I love POM. Yup. No question. As I said in my column about ***A Dozen DCC Myths*** (March 2014), there is nothing intrinsically dangerous about POM.

Several folks got on the MRH forum connected with that March column and disagreed. Check it out for more details. However, they failed to show that POM was dangerous. What they did show is that ANY programming possibility in the hands of someone who isn't fully cognizant of their actions can cause problems. There is a solution: don't let guest operators (including club members) use throttles that have any possibility of programming things on the layout via any mode.

There are, as I said above, two methods of POM. What I'll call traditional and Blast Mode.

5: Digitrax Zephyr - the system that put Blast Mode programming on the map - photo courtesy Digitrax

Let's look at traditional first. When you enter this mode, the command station will ask you what loco you want to program. That is a strong clue that you are entering POM mode. You need to verify or correct-and-verify that address. Once you get past that gate, nothing you do will impact any other loco on the layout. With this method, you can run the loco and tweak to your heart's content. That's why I love it.

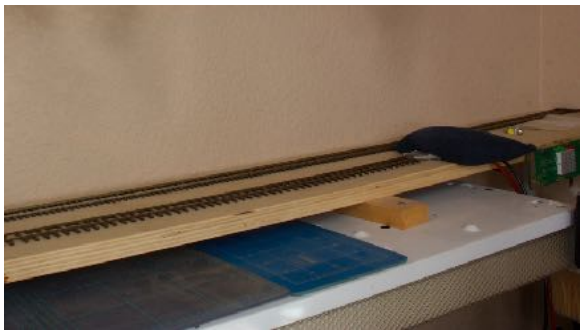
A similar method is Blast Mode programming. This came into vogue as part of the Digitrax Zephyr system as a work-around for programming sound decoders without a PTB. If you choose to use this method, follow the directions on the Digitrax web site (**http://www.digitrax.com/tsd/KB33/programming-soundtraxx-tsunami-decoders-with-digit/**) very carefully. Failure to do so may result in messed up loco programming. Because of this danger and the intrinsic inability to read what you've written, I recommend against Blast Mode programming. Get a PTB instead.

Two types of programming track

Okay, this may be a bit of a stretch, but there really are two types: one on the layout and the other on a workbench. There are special precautions necessary with the layout version to assure isolation from the DCC track in all circumstances.

Programming Track on the workbench

I have a track on a shelf above my workbench and use a NCE PowerCab to run it. The PowerCab has enough oomph to run most of the locos I work on, even some G-scale models. It doesn't need a PTB to talk to any decoder I've ever put on it. It reads out the amount of DCC current being consumed on the test track. With a NCE USB adapter, it allows me to run DecoderPro. I know of many dyed-in-the-wool Digitrax users who have a similar setup on their workbench.



6: My shelf test / programming track

This is one place where the PowerCab's shifting from DCC track to programming track and back is invaluable. No switches to throw or think about. Let's show how this works with a real-life example.

Real-life Example

When I install a decoder (or do significant rework on an installed decoder) my first place to go is the programming track. Why? Because the current available is limited to a few thousandths of an amp, as opposed to the many amps or many tens of amps available on the DCC track. If I can write to and read from a decoder on the current limited programming track, there is a 90+% chance that I've wired the motor and track portion of the decoder correctly. If I've wired it wrong, I probably won't have damaged the decoder due to the current limits on the programming track.

So here's how I finish an installation. My goal is to put the loco on a current limited track and test it before I take it to the high power DCC track.

- I set the PowerCab into programming track mode (press PGM and then 4), which will start it reading the manufacturer's ID. I don't worry about what number comes up, as it will most likely be wrong. Why? Because I can't put the loco on the track until the PowerCab starts to read the manufacture's ID and have it be limited. Since the decoder gets connected somewhere in the middle of the read process, it won't necessarily return the correct value.
- The loco goes onto the track next. Yes, it is in the middle of a read, but that way I know that the full DCC power is off the track.

- When the PowerCab finishes its read, I press ENTER which moves it into reading the decoder type. Since the loco is on the track for this entire sequence, the number returned should be accurate. I check for a good read back of the decoder type number. For example, "64" if I'm working on a SoundTraxx Tsunami.
- I then follow through the sequence to read the short address (default 3 for new decoders), write a new short address (frequently the right two digits of the cab number), and read and write a long address.
- If all of these reads and writes give the expected results (the addresses I asked to write were read back okay), then I move on to programming the decoder. Otherwise, I check my decoder wiring for opens, shorts, bad motor and such.
- After starting DecoderPro, I walk through the Programming a New Loco sequence in the programming track mode. Then I do a READ ALL SHEETS, so that DecoderPro has exactly the same data in it as the locomotive. Saving this file makes sure that the computer retains the same data.
- Finally, I go to POM mode (with the PowerCab, it is as simple as changing modes in DecoderPro and opening the saved file). This allows me to fiddle with speeds, sounds, momentum, and all. As long as I do all the adjustments in DecoderPro, the copy I save onto the computer represents what is actually in the loco.

Programming Track on the layout

Lots of folks say that they want to be able to read from their decoders without having to remove the loco from the layout. To do so, you need a place on the layout where you can run a loco into a track that can be isolated from the rest of the layout and connected to the command station programming track output. This is a great place to insert a PTB between your command station and the programming track itself.

The programming track needs to be longer than your longest locomotive, so that all of it is on the track. Given that locos can move when the ACK current pulses through the motor, the longer the better. The isolation section(s) need to be longer than your loco, too. When the programming track is active, the isolation sections need to be

7: Programming track surrounded by isolation tracks with connections to the layout on both sides

dead - nothing connoted to them. That way, if a loco runs off the end of the programming track, it will be contained within the isolation track and not bridge to the DCC track. This prevents DCC power from being fed back into your programming track output or your PTB and damaging them. See figure 7.

If your programming track is in the middle of a run, you'll want isolation sections on both sides of it. If it is at the end of a run, you only need an isolation section on one side.

What makes for a good programming track on the layout? I like:

- Convenient location where a loco can be put on and taken off easily
- Able to be electrically isolated from the rest of the layout
- Close to the main track or the center of the action

The switch that selects the program or run mode needs to have four poles, be double throw and handle the DCC current with a break-before-make design. A center-off style switch automatically is a brake-before-make style. Check out your local auto parts store for these. It may be difficult (and pricey) to locate with an appropriate (5 - 10 amp) rating.

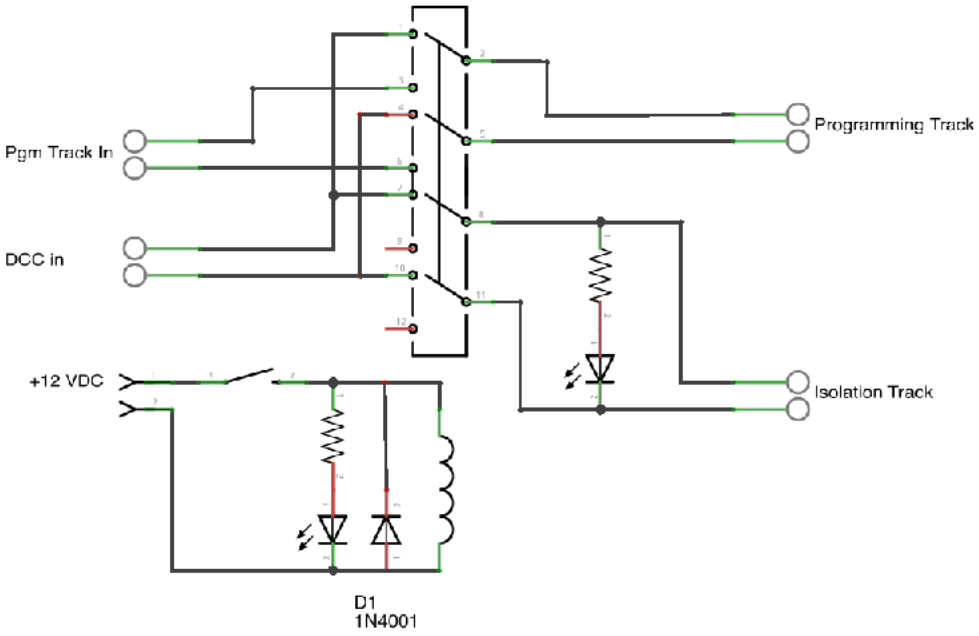
For a method of achieving this isolation with readily available parts, see the ***Mr. DCC's Workbench*** after this column.

Now, if you have read the entire column, tell me that you didn't learn one thing, no matter what level of DCC expertise you have. I know I learned a few new things researching the column. If you found this column helpful, please click on the Reader Feedback link here and rate it **awesome**. Please join in the conversation that invariably develops there. Check back next month. Until then, I wish you green boards.

Mr. DCC's Workshop

Isolating the Programming Track on your layout

Recently, I've come up with an easy way to switch the isolation and programming tracks and have a readout of what you are doing. The schematic diagram is shown for those of you who relate to those things. I didn't show the PTB or the wall-wart to power the relay in the schematic. There is a block diagram later for those of you who don't relate to schematics.



8: Schematic diagram of programming track switch relay circuit

Parts List

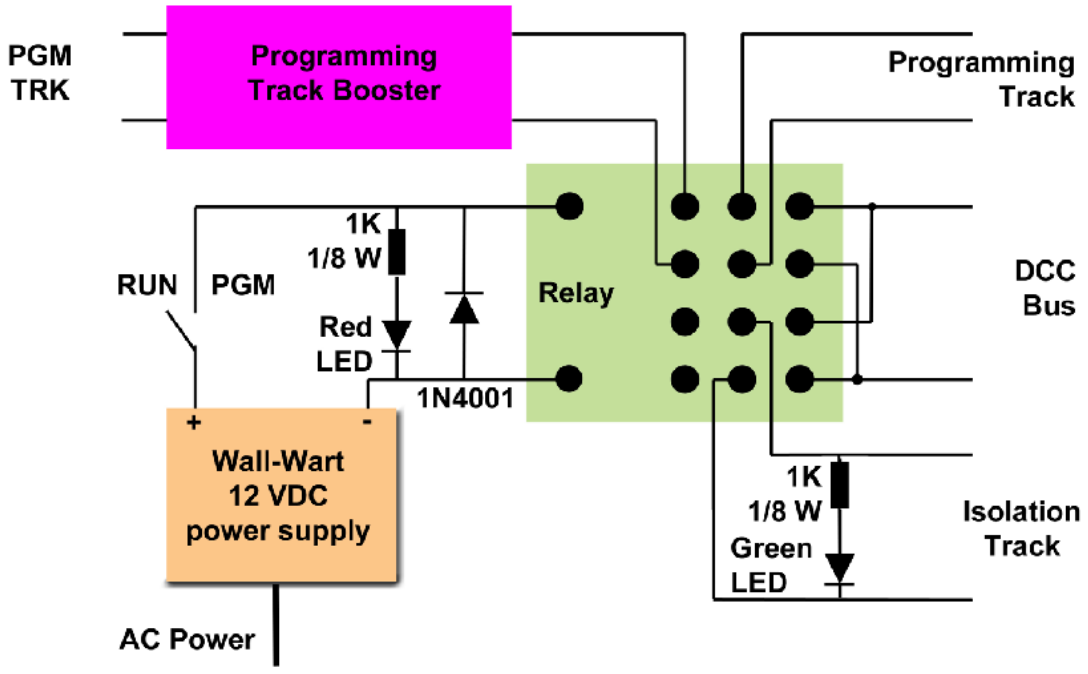
QTY	ITEM	DIGIKEY PART NUMBER
1	Relay - 4PDT with 12 volt coil	HJ4-DC12V
1	Miniature SPST switch	M2011SS1W01/UC
1	Rectifier	1N4001-G
2	Resistor 1 K ⅛ watt or larger	CF18JT1K00
1	LED red	C503B-RAS-CY0B0AA1
1	LED green	WP710A10GD

This parts list is not extensive, as it doesn't include the PTB or the wall-wart power supply. I included Digi-Key part numbers, but most, if not all can be purchased from surplus stores, like All Electronics. The relay listed has 5 amp contacts. Larger

systems will need a beefier relay - 10 amps. The LEDs and associated resistors are optional.

When the switch is open, the power is not applied to the relay and the DCC power is applied to all the tracks and the green LED is lit. Closing the switch applies the DC power to the relay and it breaks contact with the DCC signal before it makes contact with the programming track signal, whether directly from the command station or through a PTB. The red LED will receive power when the relay does, so it will light. The 1N4001 diode is necessary to keep voltage spikes from the relay releasing from damaging the power supply. The exact value is not critical: 1 amp minimum at 50 PIV minimum. The LEDs and their attached resistors are optional.

Here is a block diagram for those of you who don't relate to schematics. The PGM TRK input in the upper left connects to the command station programming track output. The two track outputs on the right go to the respective tracks and the DCC bus connects where shown on the right. If you have 12 VDC on your layout, that can be used in lieu of the wall-wart. The relay selected will require less than 100 mA to activate, so virtually any 12 VDC wall-wart will suffice. Not shown in this diagram is the power to the PTB. Follow the manufacturer's recommendations.



9: Block diagram of the circuit shown in figure 8. A power supply and PTB are shown, too.