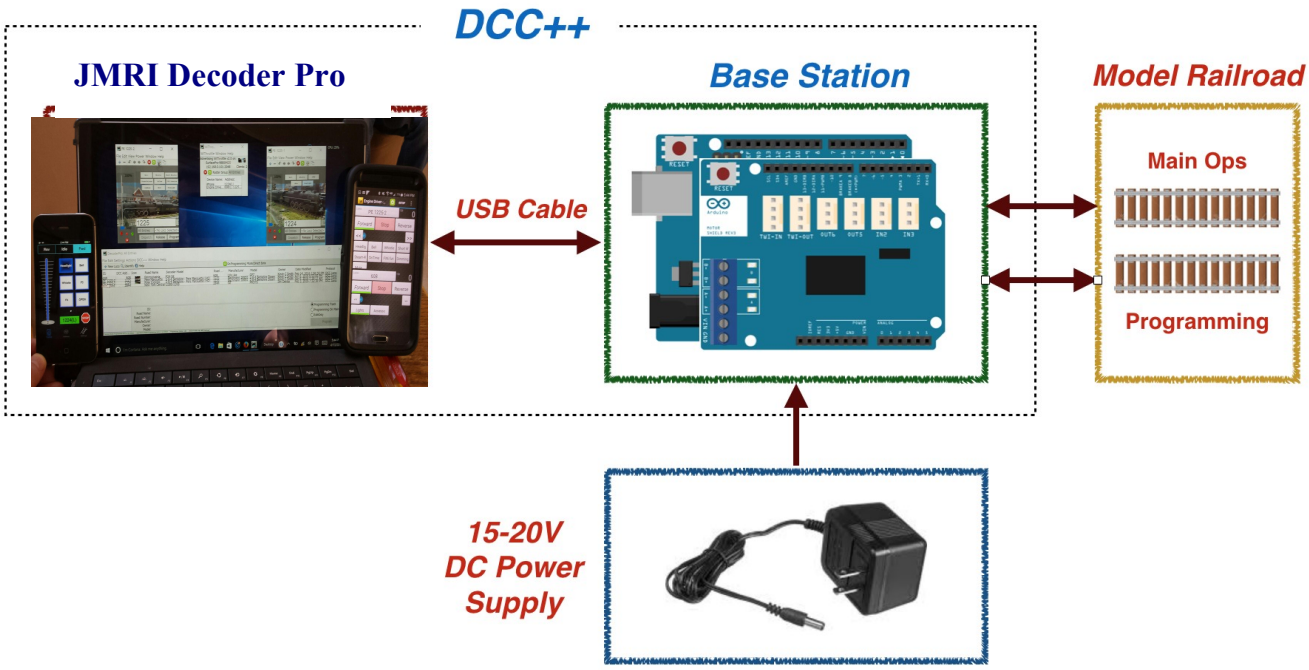
DCC++ A Complete Open-Source DCC Command Station and Interface for Operating Model Railroads

DCC++ Base Station
Arduino Micro Controller
with
Java Model Railroad Interface
Decoder Pro

'Getting Started'



DCC++ Base Station 1.2.1 and DecoderPro 4.6 & 4.7.1 - Getting Started
Version 1.2 April 5, 2017 Kevin C Smith

Introduction
DCC++ is a free open source software solution for the Arduino micro controller & motor shield developed by Gregg Berman. Together it is a full function and complete Digital Command Control DCC Base Station and a Engine Decoder Programming Station, with a connection to the USB port of a personal computer or similar device. DCC++ software is NMRA DCC compliant and supported by Java Model Railroad Interface JMRI DecoderPro and PanelPro. The JMRI software is open shareware available as a free download. DCC++ is also capable of operating a full layout by itself or by using the features of DecoderPro and PanelPro. The DCC++ interface to JMRI was primarily developed by Mark Underwood, (twindad)

DCC++ Features;
- Programs virtually all NMRA compliant DCC decoders
  - 2-byte and 4-byte locomotive addressing
  - 128-step speed throttling
  - Activate/de-activate all accessory function addresses 0-2048
  - Programming on the Main Operations Track
    - write configuration variable bytes
    - set/clear specific configuration variable bits
  - Simultaneous control of multiple locomotives
  - Control of all cab functions F0-F28
- Advanced features; operating turnouts, control your own digital inputs, digital outputs, and even analog inputs --great for reading panel switches, sensors, and occupancy detectors, as well as controlling servos &LEDs
- Programming on the Programming Track
  - write configuration variable bytes
  - set/clear specific configuration variable bits
  - read configuration variable bytes
- Easy to use graphical interface with DecoderPro
- USB interface std or optional Ethernet for easy connection to PC
- USB activity LED shows communication with the PC
- Four LED's turn on when programming & main track power is live
- Free Open Source Software development & support is on going

DCC++ Base Station for Decoder Pro minimum requirements:
- Arduino Uno 328P R3, or a Arduino Mega 2560 R3 micro controller
- Arduino or compatible L298P R3 Motor Shield. Others available
- USB A to USB B cable (prt)
- Regulated DC Power Supply
- JMRI Decoder Pro online or from a USB thumb drive.
- A 3ft length of track for programming and/or test running
- No extra hardware required for programming sound decoders
- Optional 2.1mm Female barrel power (Center Positive) connector for the motor shield vin
- Optional Clear Acrylic case

Specification/Operating Conditions
Arduino Uno ATmega 328P Revision3
Input; 7 - 12vdc Arduino Vin output supply current 3.3v to 5vdc. (**Never** input DC power supply greater >12vdc)

Arduino L298P R3 Motor Shield dual full bridge driver acts as a booster (2amp per channel 4 amp total)
Input; 15 - 18vdc, 2 to 5 amp to the L298P Motor Shield (**note cut Vin trace on bottom if >12Vdc is used**)
Vin output to main and programming track 14 to 16.5vdc
Notes: Input Specifications recommend not to exceed 18vdc

1. Minimum supply voltage depends upon the requirements of the decoder being programmed. In general it is safer to use as low a voltage as possible in case of problems with a newly installed decoder.
2. DCC++ will remove track power if output current exceeds Current Sample Max as measured 100ms after applying power.
3. Arduino Uno is protected against reverse polarity connection but will not work unless the polarity is correct.

Arduino DCC++ Base Station is not protected against track and main power connections being interchanged.

DCC++ Base Station & Software Notes
It is **assumed** the DCC++ Base Station is built and that the Arduino Interactive Development Environment IDE 1.8 software editor or newer is already on the PC & the DCC++ Base Station Software v1.2.1 [DCCpp_Uno.ino] or later has been uploaded from the PC into the Arduino Uno Or Mega micro processor using the Arduino Interactive Development Environment IDE 1.8.

Recommend that the DCCpp_Uno.ino program tab 'CurrentMonitor.h' have the current_sample_max set as follows;
```
#define  CURRENT_SAMPLE_MAX    600   // 300 for N scale 1amp (800ma),
                                     // 600 for HO scale 2amp (1600ma)  shut down protection
```

IF Needed, Please See **Arduino DCC++ Base Station 1.2 - 'Build & Setup' PDF documentation for further details to proceed.**
also additional information at https://sites.google.com/site/dccppsite/home & https://github.com/DccPlusPlus/BaseStation/wiki

JAVA Model Railroad Installation steps

This document gives brief instruction instructions for the Windows Operating System 8 or 10. For further instructions and for instructions to install DecoderPro on Mac OS & Linux operating systems, please refer to the Install Guides on the JMRI website at http://jmri.sourceforge.net/download/index.shtml

The following steps are required to install on your PC before you can use JMRI for the first time:

- Install Java Runtime 1.8 or newer and the JMRI DecoderPro 4.6 or newer software on the PC
- Connect USB A to USB B Cable from PC to Arduino DCC++ Base Station USB port
- Plug 15-18vDC power supply to the Top right Motor Shield vin, gnd. "*NOT to the lower Arduino processor board"*
- Connect wiring to the Programming track & Main track.
- Start Decoder Pro from the PC desktop Icon.
- Edit DecoderPro Preferences & Connections i.e. pick manufacturer DCC++ and your Com Port.

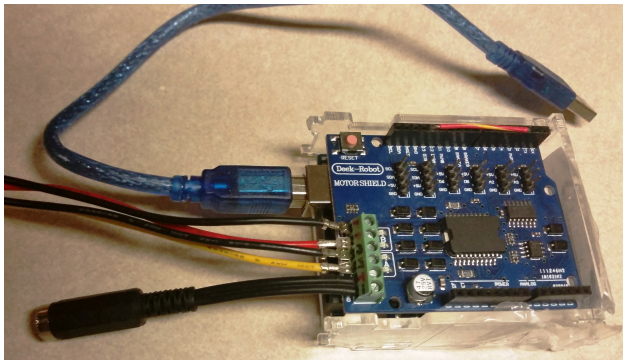
Installing Java & JMRI Decoder Pro Software

Java and DecoderPro can be installed from on-line from a USB thumb drive.
Download and install Java Runtime 1.8.1 or newer on the PC, https://java.com/en/download/windows_offline.jsp
Download and install JMRI 4.7.1 or newer onto the PC http://jmri.sourceforge.net/download/index.shtml
JMRI install guide Windows Help http://jmri.sourceforge.net/install/WindowsNew.shtml

Browse to the download directory specific to your operating system to find the Java 1.8 and JMRI 4.7.x and Install them..

To install from the USB thumb drive such as E: on Windows,
E: \DCC++ & JMRI Programs\Java Releases\ jre-8u121-windows-i586.exe or the latest release
E: \DCC++ & JMRI Programs \JMRI Releases\ JMRI.4.7.1-R1b7dd65.exe or the latest release.
Double click on the file and follow the easy standard installation instructions.
After Decoder Pro is installed, in the future you can check for updates and download newer releases while in Decoder Pro through the Help 'Check for Updates' screen.

DCC++ Base Station + Arduino Uno with a Motor Shield on top



If additional help is needed, step by step visual aids http://trainelectronics.com/DCC_Arduino/JMRI_DCC++_Setup/index.htm

Connect the USB cable to PC & plug in External DC Power Supply to the Motor Shield

Power is supplied to the Arduino Uno processor board via the USB B cable connection.
The Separate DC Power connects to the Motor Shield to the right two terminal block connecters labeled +vin & -gnd
The Programming & Main track power connect ions are now AC current so the -+ leads are negligible as described below.

- Programming Track wire connects to the left two terminal block screws labeled B - & +
- Main Track wire connect to the middle two terminal block screws labeled A - & +
- The DCC++ system has track shorting protection and will automatically turn power off when shorted
- Connect the 15-18vdc power (or connect a 2.1mm barrel plug 'Center positive') in the right two +vin & -gnd terminals on the Motor Shield and tighten the screw. The Arduino motor shield is protected against reverse polarity connection of the power supply but will not work unless the polarity is correct.
- The four LED will only illuminate steadily when the Decoder Pro software power button is turned on and the USB is connected to the host computer (next step).

Use the supplied USB cable to connect DCC++ Base Station to the host computer. The Arduino Uno power LED should be lit. Windows systems, v7, 8 & 10 should find and connect to the DCC++ and assign a COM port. Piece of cake...

IF needed - how to Identify the connection Port

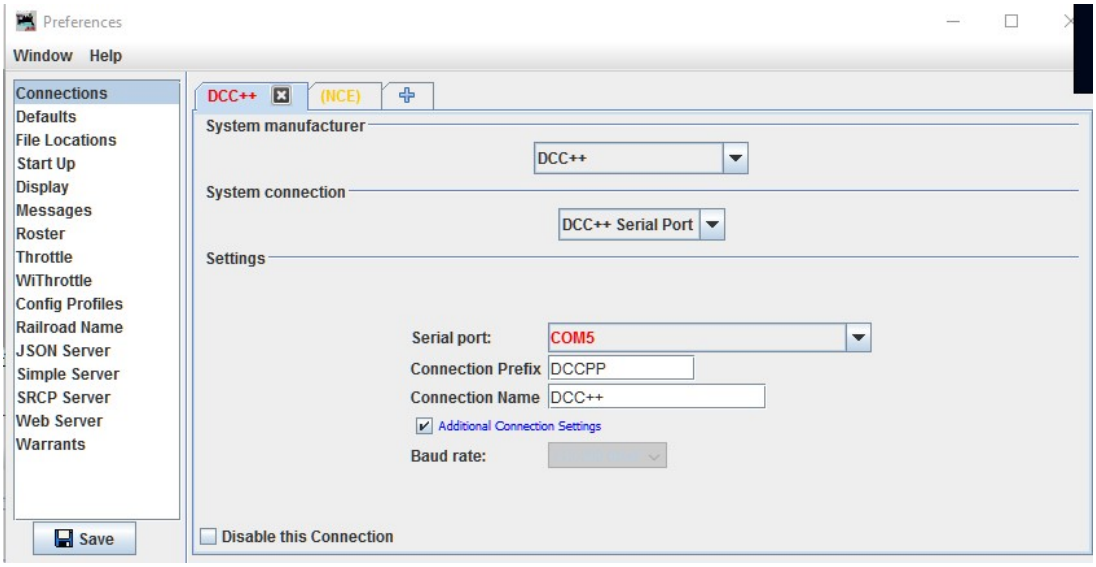
If there is a issue you can check Windows Device Manager to verify which communications Port was assigned.
Open the System Properties from the Windows Control Panel (you may need to switch to classic view in Control Panel).
Alternatively, right click on the Desktop "My Computer" Icon and select Properties. Select the Hardware tab:
Click "Device Manager"
Click the "+" to open the Ports (COM & LPT) category and note the COM port assignment for the Arduino Uno {USB Serial Device COMx} Where x is you com port number.

Starting JMRI Decoder Pro
Now start DecoderPro by double clicking the icon that was placed on the Windows desktop during the installation.

Assuming this is the first time that you have started DecoderPro, you will see a small startup window in the center of your screen, and then several other windows will be opened. For Default Owner enter Your Name. Click next,
We will continue to the Preferences window and Connections

If the Preferences window is not there, Open the Edit -> Preferences Connections dialog from the menu; below is the Preferences window from JMRI 4.6 onwards:

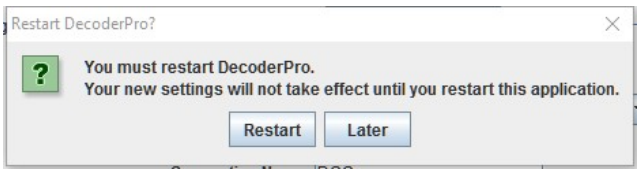
In the following example "Connected via DCC++ on COM5":



Click the arrow in the System manufacturer field and scroll down to select **DCC++.**
Then in the System connection field, select **DCC++ Serial Port**

Click the arrow in the Serial Port field and select the COM port noted during driver installation i.e. **COM4, COM5**.
If you do not see a COM offered, Try unplugging and re-plugging in the USB Cable to the Arduino.

Do not select any other fields at this time. Click "**Save"** at the lowest left of the window.

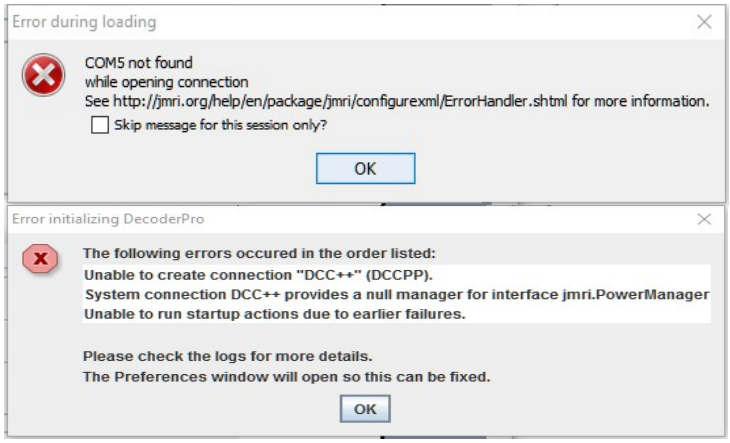


Click "**Restart**". DecoderPro will save the new settings and restart

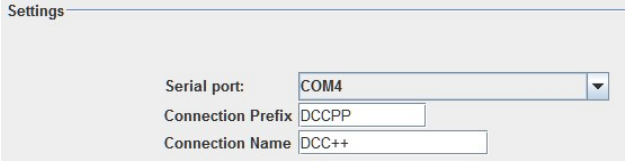
.

Possible Error not found

IF at Any Time in the future when starting up Decoder Pro you get a connection error like the following, it means its expecting the DCC++ system to have COM5 as a connection but your Device Manager assigned another USB COM port .
Click OK on both widows below and the Decoder Pro Preferences Connection window will reopen.



In Preferences just click on the Setting Serial Port tab and choose the new assigned port # offered, for example COM4



Note of Caution; Connect the Programming Track

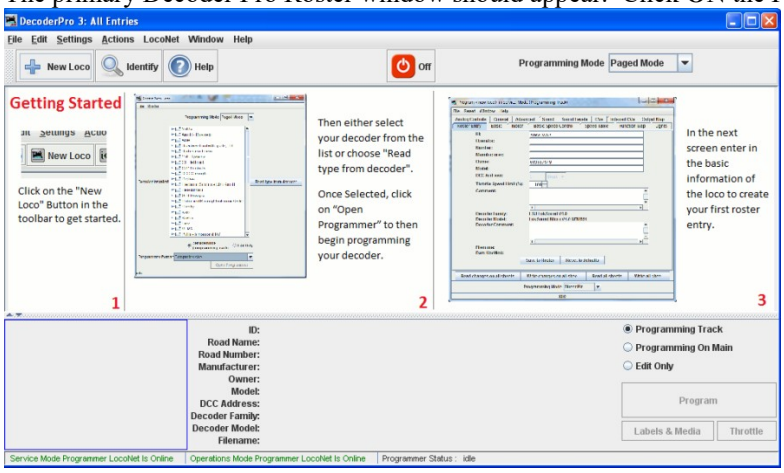
The programming track MUST be isolated from all other DC or DCC control systems and connected only to the DCC++. Damage may result to the Arduino Uno DCC++ Base Station or other equipment if this rule is not followed.

Connect the DCC++ to the programming track using the “B” terminals of the pluggable terminal block.
There is no requirement to observe any particular polarity when connecting the programming track. The DCC output voltage to the track will be approximately 1V or less below the power supply voltage, typically 15V with the standard power supply.

During programming the track current is sensed 100 milliseconds after the programming track is powered up. If the current exceeds 250 milli Amps then the programming track power is removed to avoid potential damage to an incorrectly installed decoder.

Decoder Pro - Roster

The primary Decoder Pro Roster window should appear. Click **ON** the Power Button . You can now add new engines.



This help window only shows up the first time you use decoder pro.
From now on you will see the Decoder Pro All Entries Roster Screen.
Tip: If you want to Read & Write when Programming a engine be sure you choose the (') Programming Track button.

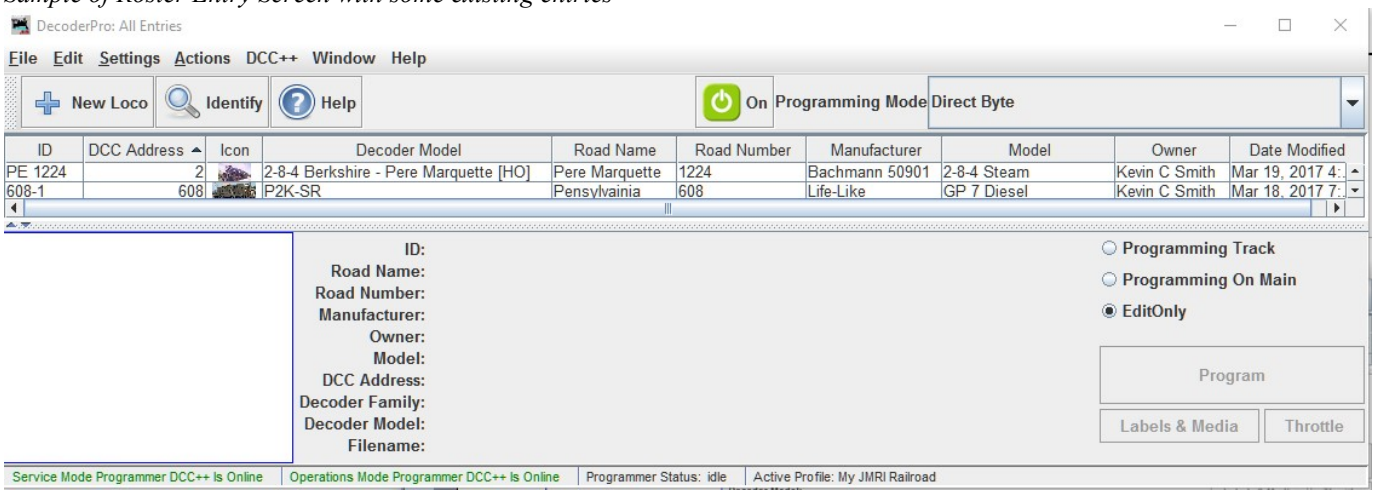
Decoder and Programmer Selection
You'll get a new window to configure the programmer with information about a specific type of decoder, or an existing locomotive. This will become your locomotive roster as you program your decoders

It is Very helpful if you have a good Idea of what manufacturers' DCC decoder you have in your engine before you get started. Before changing the current programmed CV's in a decoder one should always "read" the decoder to ensure the values for the CV's are correct for the decoder being programmed rather just writing the "default" values.

To start working with a newly-installed decoder You have two choices, by clicking the "Identify" button the programmer reads the decoder and attempt to identify it against the list of your existing engines in the roster.

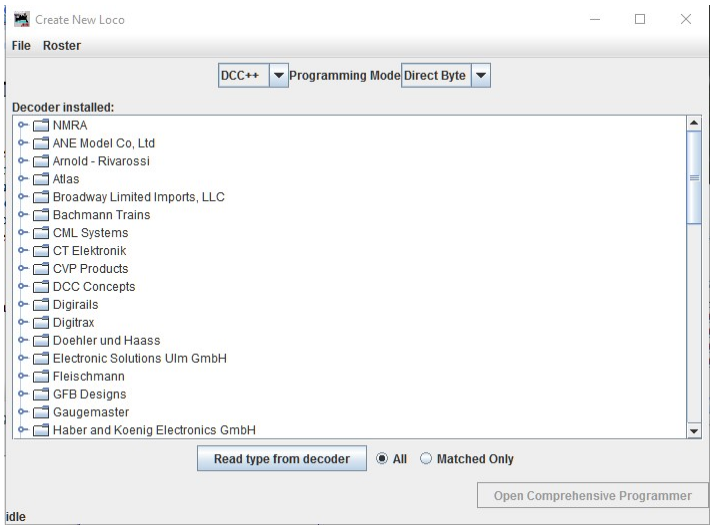
The second is add a new locomotive by clicking the "+New Loco" button

Sample of Roster Entry Screen with some existing entries



You have other 3 choices on this screen either Programming Track, Programming on the Main or Editing Only
Click on the (') Programming Track radial button then
Click " **+ New Loco**" button to open the Programmer Setup window.
A new window will pop up which shows a list of decoder manufacturers.

Create a New Loco

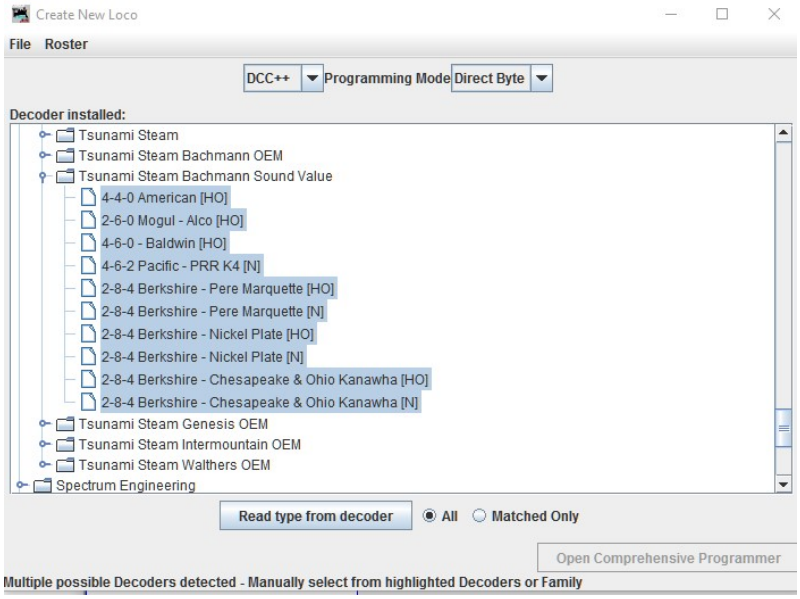


Clicking the icon or double clicking the folder icon next to a manufacturer name will open a list of decoder types from that manufacturer. You can always manually select a decoder type in this way.

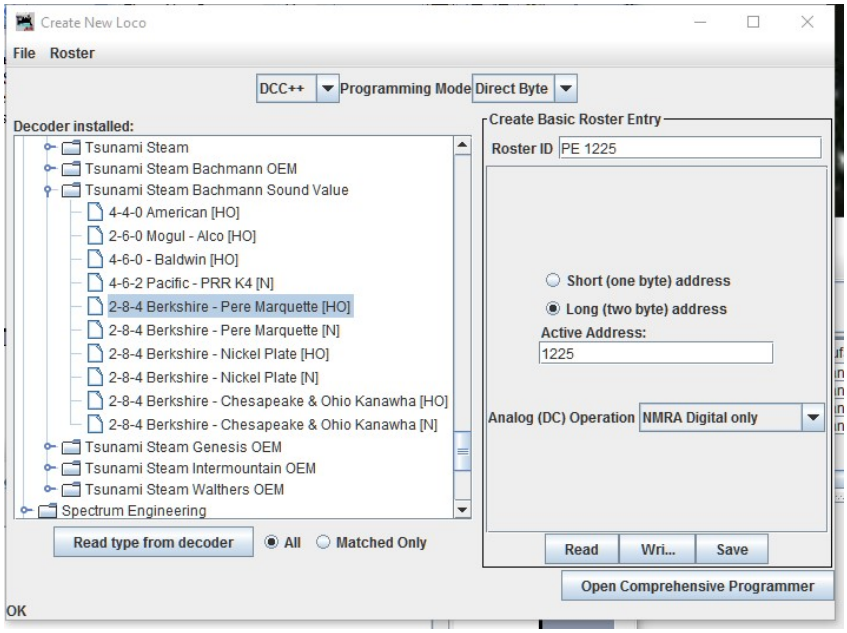
In most cases, DecoderPro can determine the manufacturer and decoder type automatically. Place a decoder equipped loco on the programming track and click on "**Read type from decoder**".

In the example below, DecoderPro has identified a Tsunami Steam Bachmann Sound Value decoders. Sometimes DecoderPro can identify the manufacturer but not exact model of decoder fitted. This is because the vendors often use the same readable ID version for multiple similar decoders. Often the difference is physical such as a steam 2-6-0 is a different from an 2-8-4, or N vs HO scale or they have specific sounds for a different railroad; these do not affect the programming, and so they use a common number for that whole series of models.

Create a New Loco



In these cases it will be necessary to select the decoder type manually, often from several highlighted possibilities.
While the programmer is talking to the decoder, status will be displayed in the bottom of the window; "Idle" or "OK" means that things are working. If it succeeds, it will select the decoder model in the selection box. Usually it will only be able to narrow the selection down to a few choices. Check that the right model is selected in the "Decoder Installed" box; update the selection if desired.

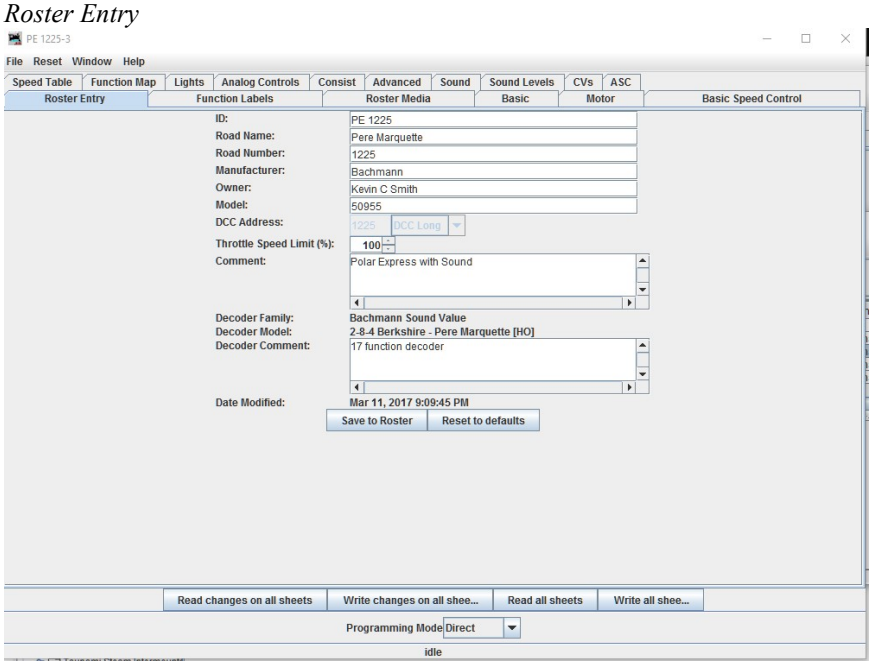


After selecting the nearest description you need to enter a unique name in the top box (initially marked <new loco>) as this will be the name of a file in the Roster with all the settings for this loco. The road number is often a good DCC choice.
Type a new "Unique" Roster ID: PE 1225 and assign a unique Active Address:1225 then click "**Write**" button.
Once the decoder type has been selected, click "**Open Comprehensive Programmer**".

The service mode programmer window opens with a selection of "tabs" grouped in rows below the menu bar, Clicking on a tab selects a group of CVs to be programmed, related to the title of the tab. The selection of tabs available will vary depending upon the features supported by your decoder and the CVs available. On the "Roster Entry" tab you may enter arbitrary details of the loco to be saved in Decoder Pro on the PC for future reference.

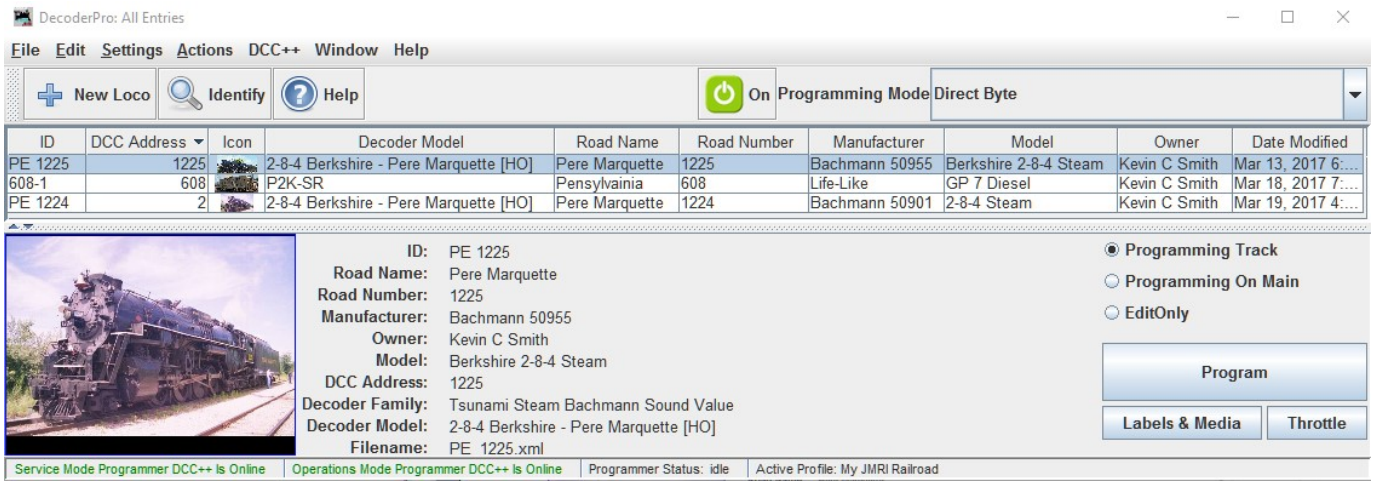
Comprehensive Programmer

Roster Entry



Everything else on this pane is optional, and for your convenience. The DCC address field will be filled in by DecoderPro once the decoder has been read or programmed. If you wish to use the Roster then you should make sure to click "**Save to Roster**" on the Roster Entry tab when you have finished programming the decoder. The Roster is especially useful if the decoder loses its settings as seems to happen occasionally with some decoders. Alternatively, you may wish to program a second loco with similar CV settings.

Note on 'ID:' naming When you click Save, the ID will become the file name of the roster entry created for this loco, and also the entry in the list for you to be able to select a previously-programmed and saved loco in future, by using the Identify button. Name these entries wisely!. Many people use the loco type and number, or Operator/road name and number, but choose a system that will be meaningful to you. Notice the image of Pere Marquette 1225 this is entered on the Roster Media a few pages down.



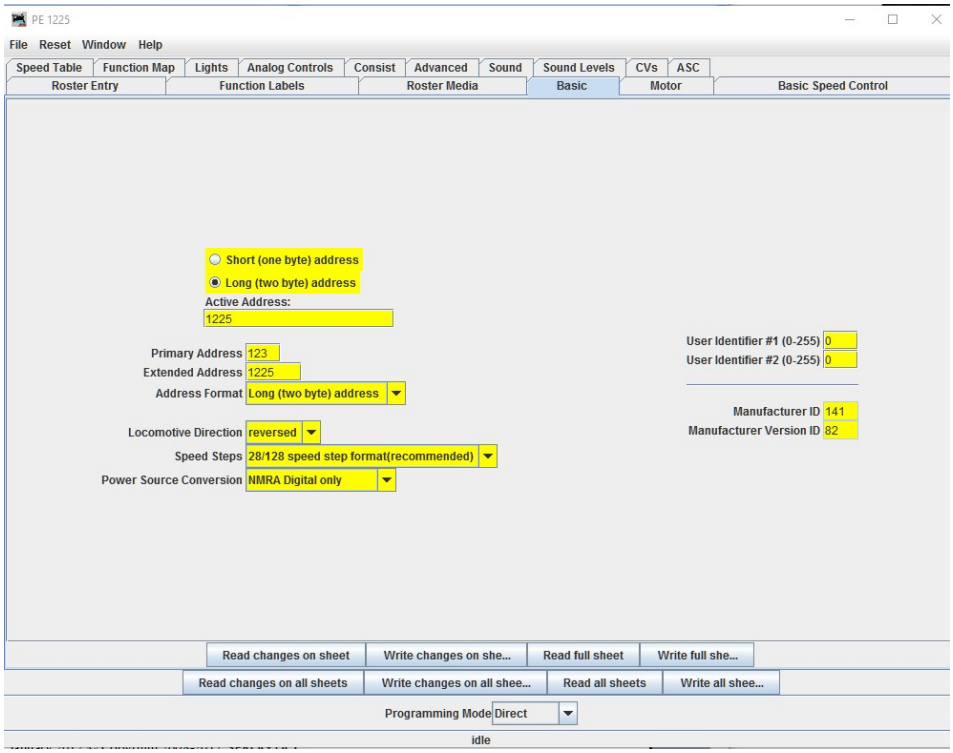
Note: Roster Program Functions (if an engine already exists you may right mouse click on a roster engine for the following choices)

Programmer Type -->	( ) Programming Track
	( ) Programming on Main
	( ) Edit Only (Allows you to edit roster without making changes to program)
Labels & Media	Open programming page to select either Label Tab or Roster Media Tab
Throttle	Open JMRI Throttle
Duplicate	Duplicate the selected roster entry to use for same type of locomotive.
Delete	Delete roster entry

Highlight the engine you wish to program click (") Programming Track then the Program button.

Programmer Detail

Then click on the "**Basic**" tab and you will see the most essential settings for the decoder including the address. Initially, all of the data fields are colored red or yellow to show that no data has been read from or written to the decoder. There are eight read and write buttons at the bottom of the window. Click "**Read full sheet**" to read the data for the current tab from the decoder. That will read CVs 1, 29, 17, 18, 19, 7, 8, 105 and 106 and fill in these values from the loco. *Careful clicking* "Read all sheets" this will read every CV from the decoder and may take a considerable time to complete. Notice the engine pulse forward on the track.



Each of the tabs selects a screen you can use to configure different aspects of the decoder. The "Basic" pane handles the address other settings you'll usually need. Using the buttons -- On the bottom of each screen are buttons that will read values from or write values to the decoder.

For example, to do a basic configuration, enter values in the fields and click "Write changes on sheet"
The variable fields are color coded to represent their states:

Edited

You've changed this value, or it contains default values from the decoder file. This is shown as orange, since the value differs from what's in the decoder.

From File

The field contains values read from a locomotive file. This is shown as yellow, since we're not certain that the file agrees with the decoder contents.

Read

The value shown has been read from the decoder. This is shown as white to indicate that the value is trustworthy.

Stored

The value show has been written to the decoder. This is also shown as white.

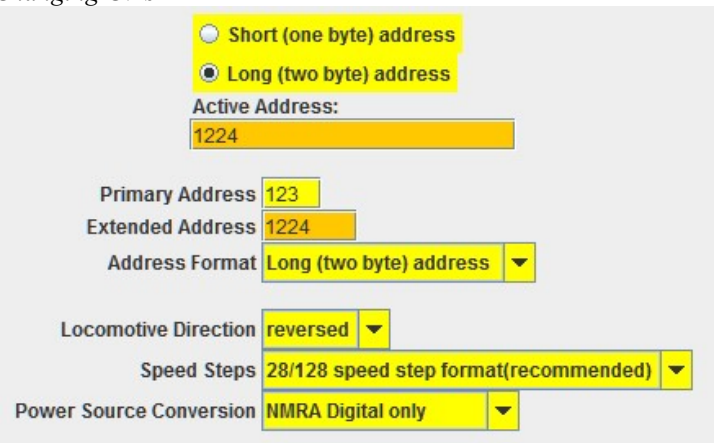
Unknown

If something goes wrong with the read or write process, we might have a completely untrustworthy result. This results in the variable's state being marked as UNKNOWN and shown as red.
Status error 306 Timeout Talking to Command Station. Check your wire connection and whether your in the correct mode.
Status error 303 may mean you are attempting to "read" while your in programming on the Main

Each data field should return to white or the background color of the window. Check that you see what you expected, especially if you are reading a decoder that was already programmed elsewhere! The exact look and layout of these programming panes may vary as versions of DecoderPro or the specific decoder types are updated, but the essential information and capabilities remain consistent. To change the locos address, Select the type of address to be set (short or extended) type a new address in the Active Address followed by carriage return. The address field will turn orange, indicating that the address has been changed but not yet written to the decoder.

Tip: There is a nice DCC Decoder Short Cuts Card link at the end of this document under Other Useful Links.

Changing CV's



Click "Write changes on sheet" to write the new address to the decoder. The address will turn red and then white as it is written to the decoder. You can set other basic properties such as the direction and DC operation on this tab. To Save your data to the Roster record on your computer, either pick from the menu bar File>Save at the top of the screen, or go back to the first pane where there is a "Save to Roster" button. The other tabs work in a very similar way. You may find it useful to have the decoder documentation available when setting more complex or manufacturer specific features.

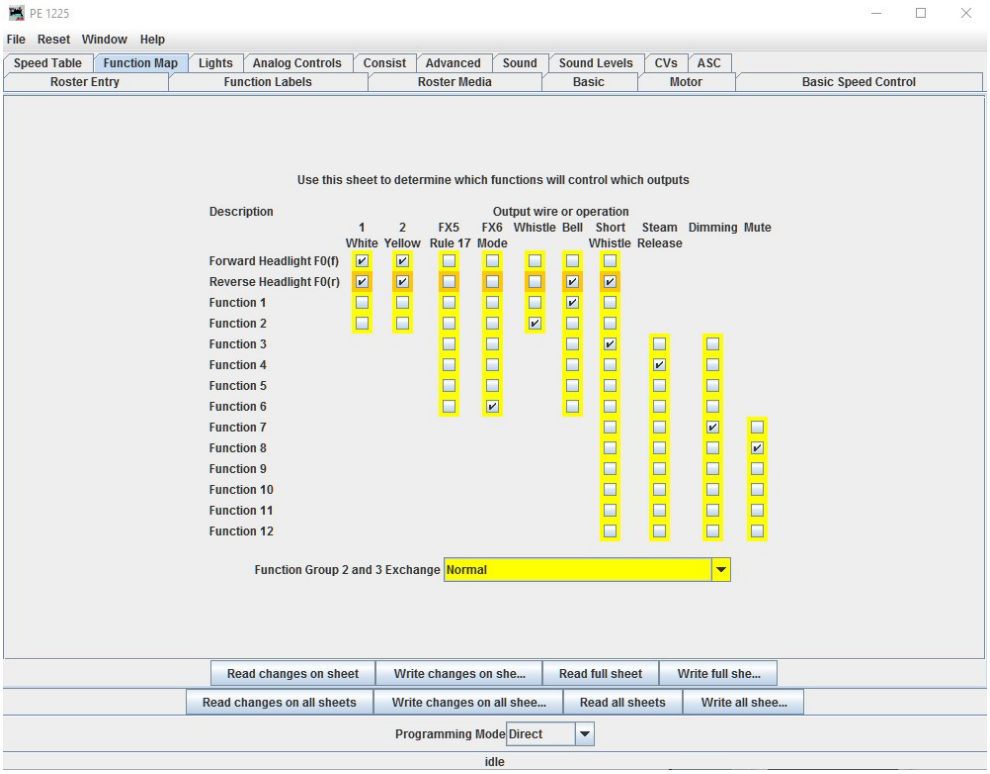
Speed Table

Next we'll look at the Speed Control tab that allows you to fine tune the way the loco responds to the throttle. Click on the "Speed Control" tab.

The decoder in this example has a choice between "Use Speed Table" or default to what is on the Motor tab. The "look and feel" of this tab will vary between decoders but most recent decoders support the use of a speed table. It is left as an exercise for the reader to experiment with the sliders for setting the speed table and the buttons just below. The "Force Straight" option will give a linear relationship between the throttle position and the speed of the loco. "Constant ratio curve" will give little change in speed at low throttle settings, greater change at higher throttle. First click 'Read full sheet' to see how the decoder is set then make your desired changes. Remember to Write Changes on each sheet before moving to a new sheet.

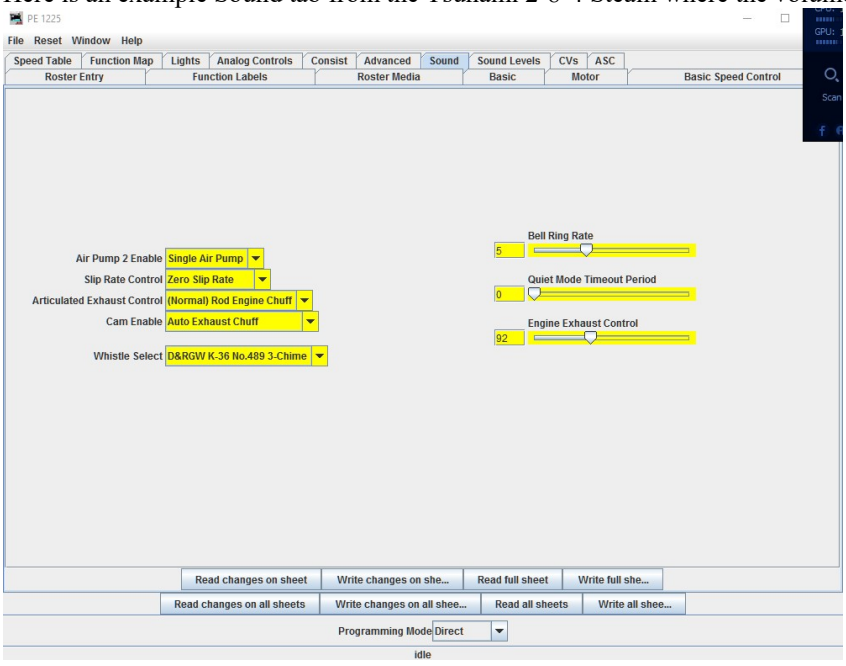
Next, click on the “Function Map” tab. The Function Map allows you (in those decoders that support it) to control which throttle function key is mapped to each output wire or operation (e.g. sound effect) of the decoder.

Function Map



You may like to add a automatic bell when backing up and a automatic short whistle when changing direction as seen above in the check marks added to Reverse Headlight line, and the orange status waiting to be written. click "Write Changes on Sheet" button to only save those items in orange. The Headlights F0 functions key must be pressed in order for the sounds to activate

DecoderPro comes into its own for programming the many and varied options in a sound decoder.
Here is an example Sound tab from the Tsunami 2-8-4 Steam where the volume of individual effects can be set.



Hovering over an item with the cursor will often show a short description for that item, but it is highly recommended that you read the documentation for your decoder to get the best from it with DecoderPro. Again, remember to write your changes.

Tip: There is a nice DCC Decoder CV Support link the end of this document under Other Useful Links.

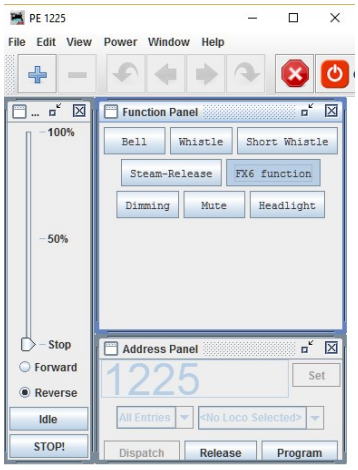
Using the DecoderPro Throttle

When "Programming on Main" track it is possible to test run a loco after writing new CVs, e.g., to test the effect of changing the speed table of function mapping, or changing and immediately listening to the different whistles sounds available.
Using a long piece of track for programming on the main is recommended for test running. More than one loco may be present on the main track layout but only one may be selected on the throttle and driven at one time.
If you Program on the Main track, Care should be taken when using the service mode programmer that only one loco is on the layout during programming, or that the layout is isolated except for a shorter section of track or spur for programming.

On the main DecoderPro window, click on the "Actions" menu item and then select "New Throttle"
Or on the Main Roster page click on the locomotive line you'd like and then click the "Throttle" button at the bottom right hand of the page and engine# will pre fill-in throttle.

When the throttle window opens, enter the loco address and click "Set". With DCC++, loco addresses less than 127 are assumed to be short (2digit) addresses. Addresses greater than 127 are assumed to be extended (4 digit) addresses.
Setting the loco address will enable the function buttons and speed slider.

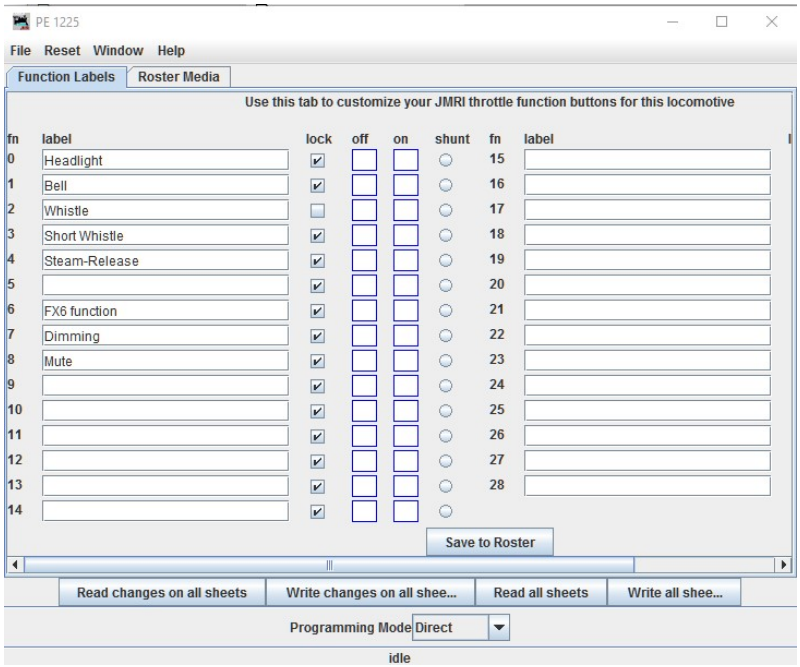
JMRI Throttle



The Power icon button on the right of the menu bar is the track power control and must be clicked until it is green to turn on the track power. The DCC++ base stations four track power LED's will turn on when the power is live. The function keys mostly have a latching operation (or lock). Click once to turn a function on, click again to turn the function off. To control the loco speed, click and drag the slider caret. For fine control of speed, click on the slider just above or below the caret. The "Idle" button allows the engine to come to a slow stop. The "STOP!" button will stop the loco but does not turn off the track power.

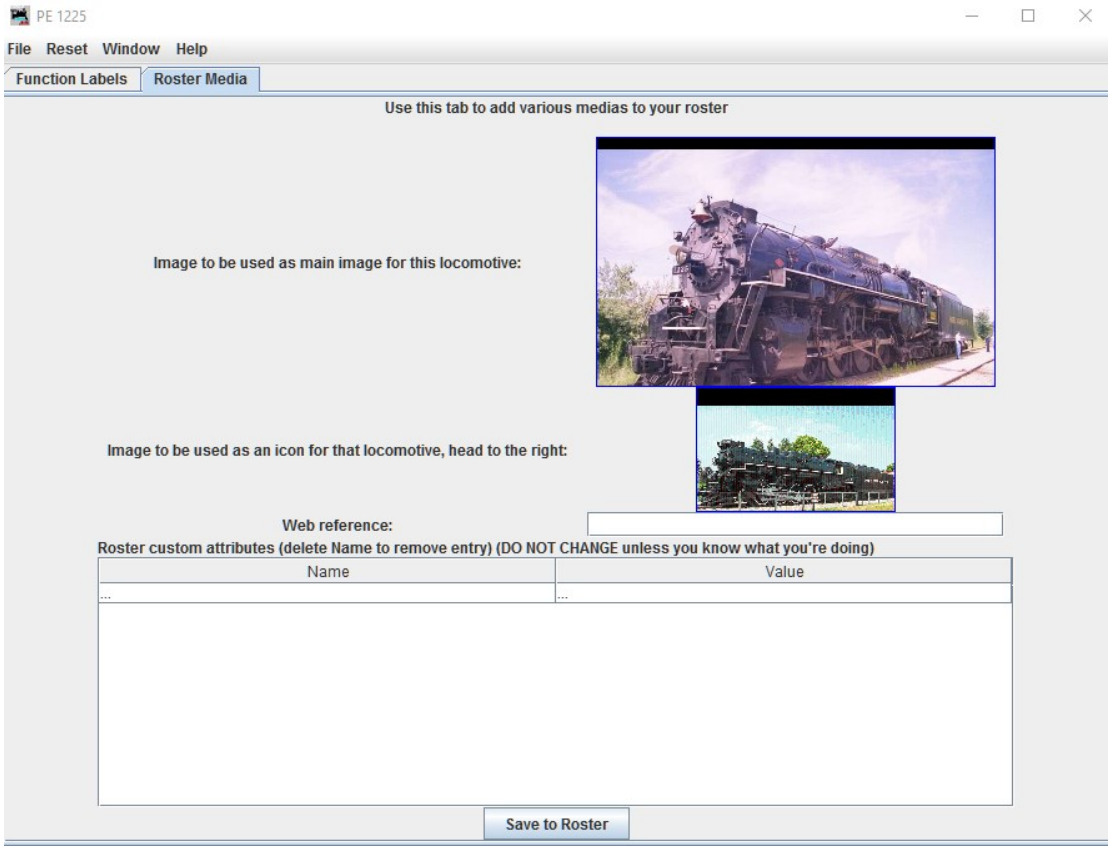
Click "Actions' tab then "Labels & Media' to label the buttons .

Labels & Media

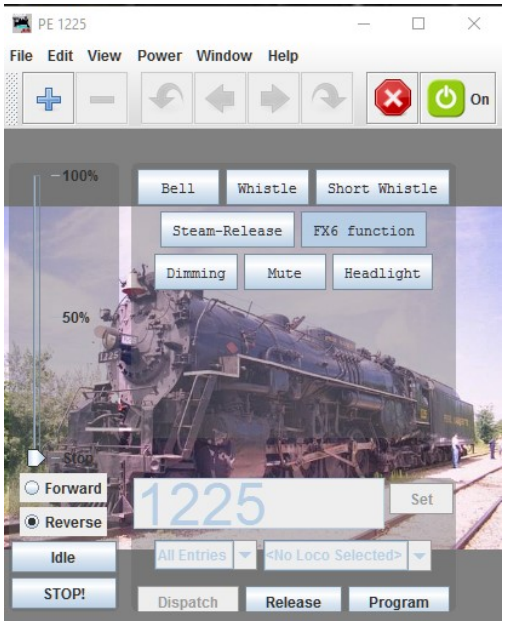


Roster Media

If you want to attach a photo of your locomotive on your engine roster and throttle, click on the Roster Media tab , find photos on your PC and drag it to the two boxes as shown. The top picture displays on the throttle, the lower picture on the roster line. You can directly access this tab by right clicking on the locomotive line in the Roster and click 'Label & Media'

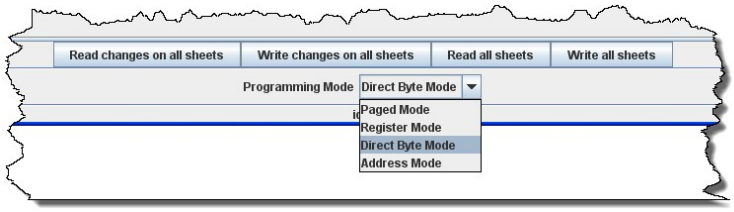


Click the Icon  to the Right of the On/Off button to display your engine on your throttle.



To set a different loco address, click the “Release” button.
enter a new engine number and click 'Set' .

Some brief comments on Programming Modes



DCC++ supports the newer "Direct" programming mode choice for DCC systems that use both the Direct Bit and Direct Byte operations when programming. Direct Bit and Direct Byte (when available) mean that the command station is told to use only one or the other, and will automatically make the choice which might be useful with very old decoders.

Address Mode is an outdated programming method that is included here for the sake of full compliance with the NMRA DCC standard.

Register Mode is an expanded form of Address Mode, and is still used by some older and/or lower end decoders, particularly some from MRC and Wangrow. It is inherently limited in its ability to access all CVs in a decoder.

Paged Mode is an expansion of Register mode that gives full access to all decoder CVs.

If you experience difficulty programming a decoder in try a thicker gage wire and a higher amperage power supply.

Measuring Loco Current

Loco current on the Main Track may be measured by using DCC++ in command station mode.
Click on DCC++ tab then the 'Track Current Meter' line. This is the Percentage of the Amp rating of your Motor Shield ie. a 50% on a 2amp shield is 1amp draw.

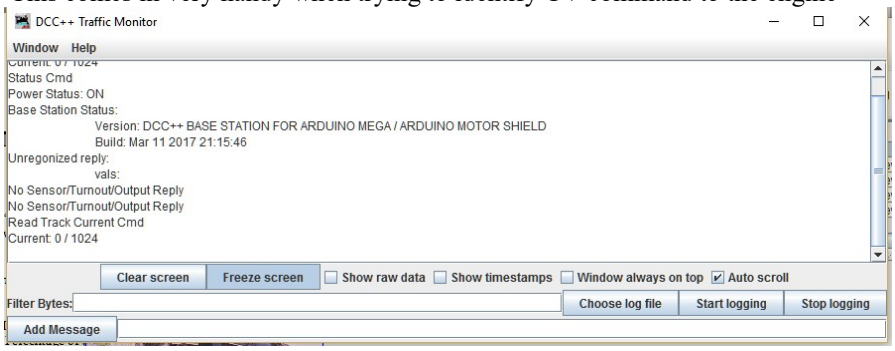
Send DCC++ Command

Click DCC++ tab, then Send DCC++ Command

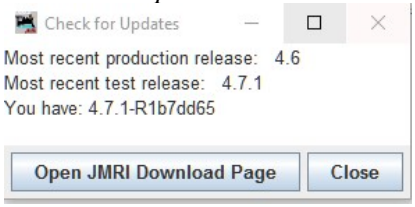
You can see the throttle commands and function buttons display on the monitor window below as well as send one character commands through the screen above. The command <s> displays the status of DCC++ base station. Tip: See DCC++ github site for complete list. https://github.com/DccPlusPlus/BaseStation/wiki/Commands-for-DCCpp-BaseStation

DCC++ Traffic Monitor

Click DCC++ tab, then DCC++ Traffic Monitor
This comes in very handy when trying to identify CV command to the engine



Check For Updates



<--- From Roster screen click *Help* tab then *Check for Updates*

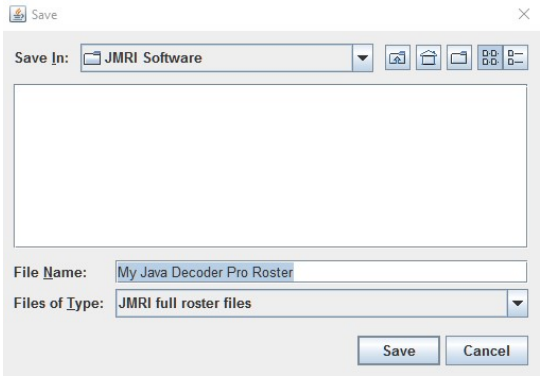
Print Previews and Roster Summary
You can print Roster Entries and/or Summaries by clicking File then Print Preview, then Summary

Back up "Export Roster" & Import Roster File Procedures

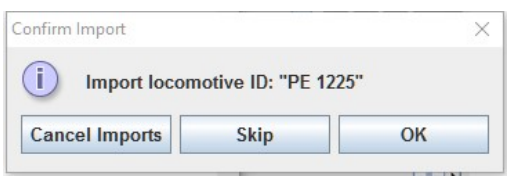
Click FILE, then

Export Roster Entry (for a single roster line, "ID")
or
Export Roster (for the Full Roster all entries.)

Give it a recognizable file name " My JMRI Decoder Pro Roster" or "Kevin's Polar Express RR" then press Save.



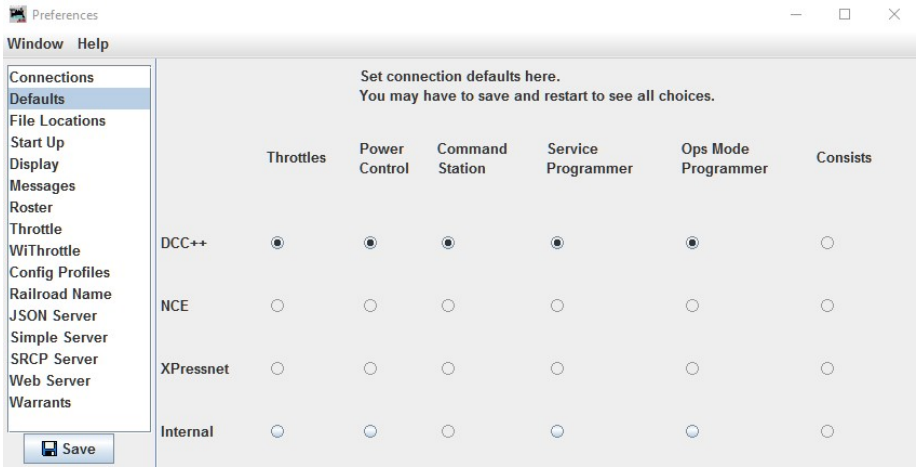
Import Roster
When you FILE> Import a Full Roster it will prompt you to either Cancel Import, Skip this engine, or OK import this engine.



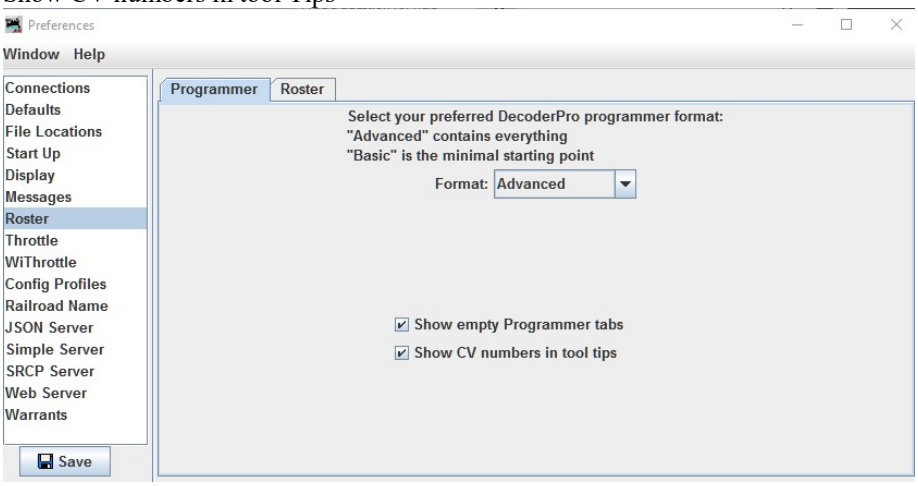
Options Settings for Preferences.

Your DCC++ Decoder Pro system has choices in the Preferences screen were we've already setup Connections, however, there are many additional things from Defaults, File Location, Roster etc. Here are a few suggestions on how to initially set it up

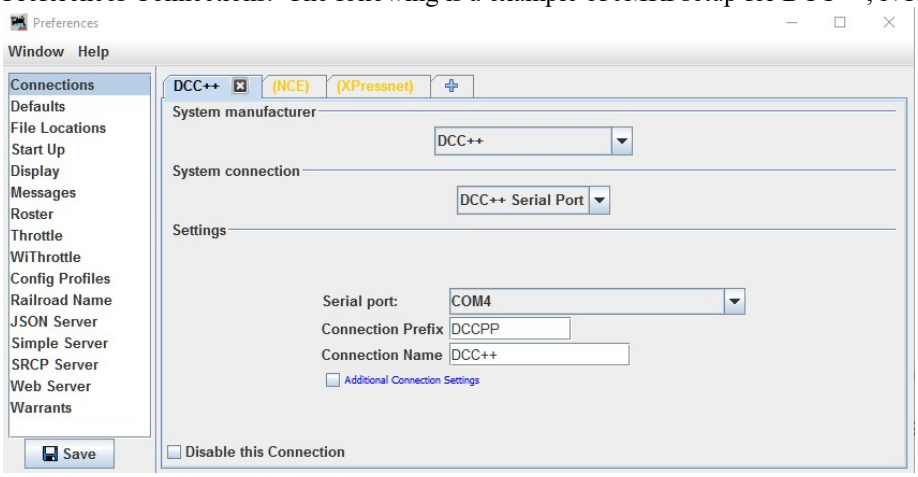
Preferences >Defaults
If in the future you set up additional connections be sure to come back to this screen and set the connection Defaults as shown to the DCC system your connected live to In this case there all set to DCC++. Defaults set to one system or the other not mixed between them, Otherwise you'll be playing ping pong on the Programmer All Enter screen and Tab screens.



Preferences >Programmer
Initially Set the Preferences on Programmer as shown. Format: >Advanced is the most robust, and check mark, turn on the Show CV numbers in tool Tips

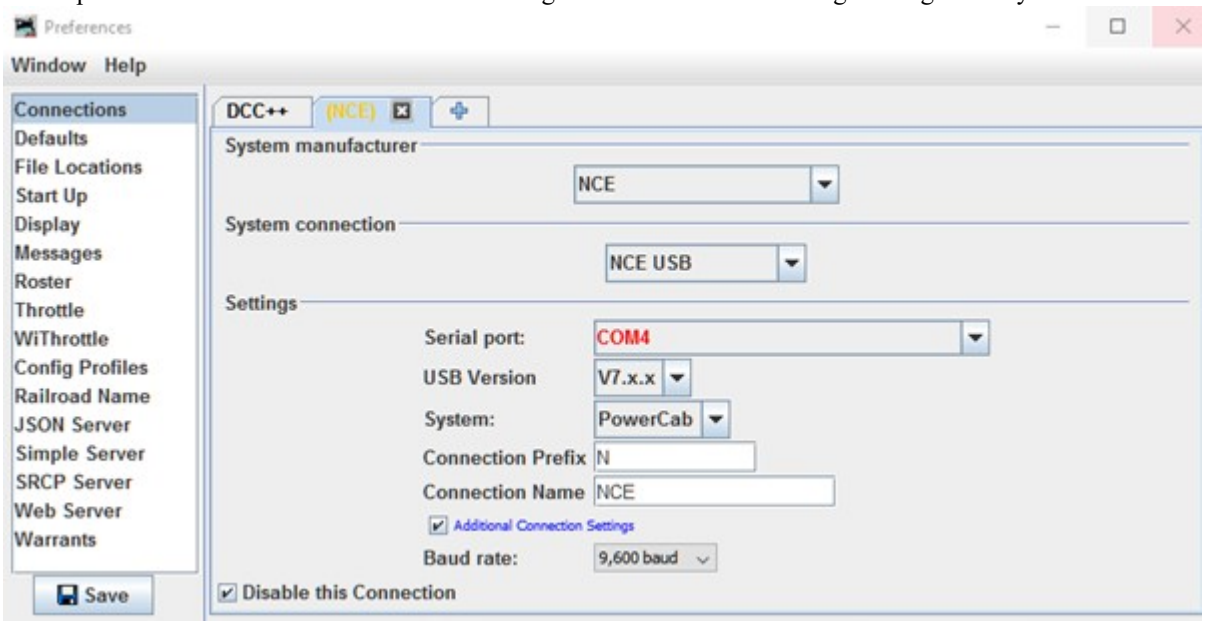


Preferences Connections. The following is a example of JMRI setup for DCC++, NCE and Lenz DCC Systems.

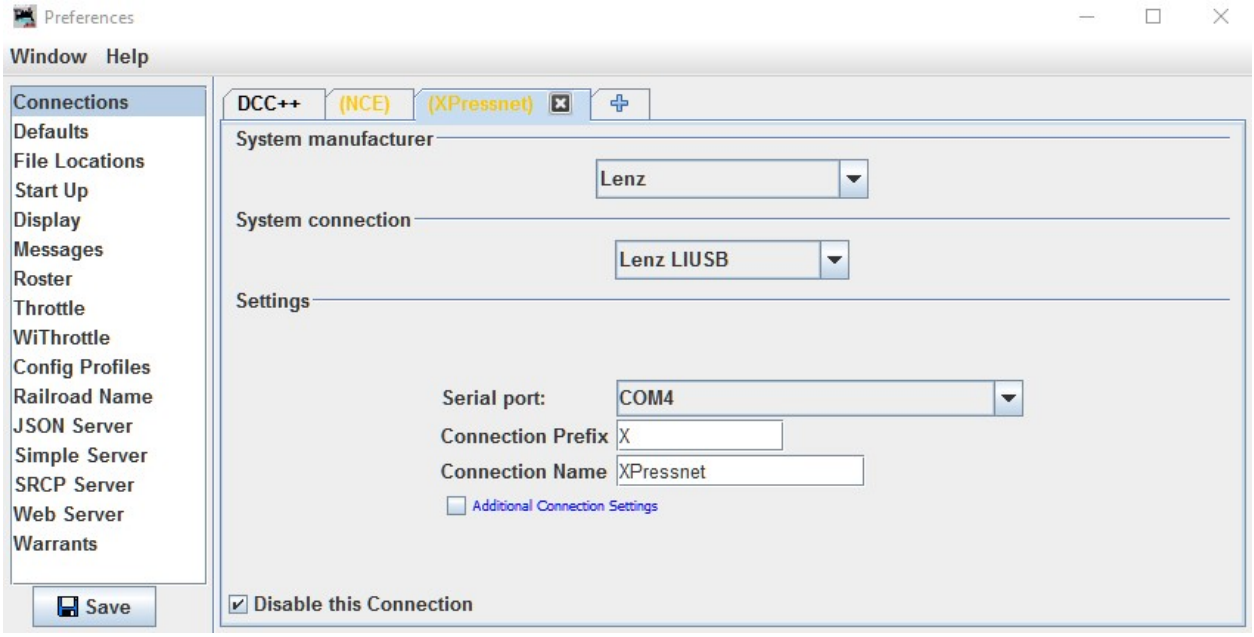


Preferences > Connections
These two sample connections show DCC++ as running and the NCE and LENZ connections Disabled

Sample Screen of JMRI Decoder Pro Connecting to a NCE North Coast Engineering DCC System



Sample Screen of JMRI Decoder Pro Connecting to a LENZ DCC System



So that's a quick setup & introduction of the DCC++ Base Station and JMRI Decoder Pro software.

For more information on decoder pro 4.6 and later see the JMRI web site and all the *Other Useful Links* below.

As of the date of this "Getting Started" document the latest and most helpful official User Guide available in PDF format from JMRI is the **JMRI DP_man_3-4.pdf** release which I've place in the folder of your thumb drive: \DCC++ & JMRI Programs\JMRI Documents\

The JMRI_DP_man_3.4 is a older release of an official manual and is much more in-depth on Decoder Pro and is meant to be used after you complete the DCC++ Getting Started doc. http://jmri.org/manual/pdf/DP_man_3-4.pdf

Troubleshooting
Arduino issue;
Fail to compile or upload, check your DCCpp_Uno software edits and make sure you use // to comment out a line.
Try a different USB cable as yours may be faulty.
If you are experiencing intermittent faults with your DCC++ base station, please ensure that you are using a good quality regulated 12vdc 500miliamp for Z scale, 12-15vdc for N scale and 15-18vdc for HO scale =>2 to 5amp power supply.
Recommend for Programming you use a 2 to 5 amp regulated power supply for all scales to insure sound decoder read/writing.

DCC++ issue:
There is a DCC support page which will be updated to reflect the most common questions people have about DCC++.
DCC++ Development & Support http://www.trainboard.com/highball/index.php?forums/dcc.177/
Testing the Arduino and Base Station code https://github.com/DccPlusPlus/BaseStation/wiki/Diagnostics---D---Command

JMRI issues;
One common problem is the configuration of the "Virtual COM Port" for USB. Check Device Manager on your PC to make sure your COMx port in is set to 115200 baud to match the baud rate in the DCC++ system.
First try unplugging the USB cable from the Arduino and plugging it back in to see if that reconnects.

Getting Started With DecoderPro See the DecoderPro website http://jmri.sourceforge.net for any updates and latest information. Join the JMRI Yahoo group http://groups.yahoo.com/group/jmriusers for help from other DecoderPro users and the team who develop it. Ensure that all software and drivers are installed and that DecoderPro preferences have been setup in accordance with the previous steps. Start DecoderPro. The main window will open and show the current connection method. This merely reflects the preferences setting and does not actually indicate a physical connection.

Please review the information in the section "Edit DecoderPro Preferences". If this is not successful, it is often effective to delete the file containing the Preferences settings. This file is in the user's data area on the computer and is named DecoderProConfig2.xml. On any computer, this is stored in the user's data area, in the JMRI folder. Be sure that DecoderPro is not running, locate and delete this file, and then start DecoderPro once more. Set preferences as before, and restart DecoderPro.

***Other Useful Links*:** **The links throughout this PDF document are clickable, Please use them**

Java Model Railroad Interface JMRI
- Java Model Railroad Interface. http://jmri.sourceforge.net/help/en/html/apps/DecoderPro/index.shtml
- Java Supported on DCC++ http://jmri.sourceforge.net/help/en/html/hardware/dccpp/index.shtml
- Download & install Java Runtime 1.8 or newer required on the PC, https://java.com/en/download/windows_offline.jsp
- Download & install JMRI 4.71 or newer onto the PC http://jmri.sourceforge.net/download/index.shtml
- Initial JMRI & DCC++ Controller Setup. http://trainelectronics.com/DCC_Arduino/JMRI_DCC++_Setup/index.htm
- JMRI Yahoo group for latest news and discussion of DecoderPro. https://groups.yahoo.com/neo/groups/jmriusers/info
- JMRI Clinics, Follow the links inside the clinics http://jmri.sourceforge.net/community/clinics/
- JMRI Decoder Pro 3.4 Users Guide PDF http://jmri.org/manual/pdf/DP_man_3-4.pdf
- DCC CV Calculator http://www.digitrax.com/support/cv/calculators/

- DCC CV Support http://www.digitrax.com/support/cv/
- JMRI Engine Driver for Android Smart Phone https://enginedriver.mstevetodd.com/
- WiThrottle for Apple Smart Phone http://jmri.org/help/en/package/jmri/jmrit/withrottle/UserInterface.shtml

DCC++ Base Station & Arduino Micro Controllers
- DCC++ Home Page Author/Developer Gregg E Berman https://sites.google.com/site/dccppsite/home
- DCC++ You Tube Channel. https://www.youtube.com/channel/UCJmvQx-fe0OMAIH-_g-_rZw
- DCC++ Base Station Wiki page https://github.com/DccPlusPlus/BaseStation/wiki
- DCC++ New Products Links. http://www.trainboard.com/highball/index.php?threads/links-for-dcc.95220/
- DCC++ Development and Support. http://www.trainboard.com/highball/index.php?forums/dcc.177/
- My Experiments with DCC++ Dave Bodnar, developer http://model-railroad-hobbyist.com/node/25429
- DCC++ Facebook page https://www.facebook.com/groups/1406785379394934/1444943065579165/

Arduino Micro Controllers
- Arduino Micro Controllers https://www.arduino.cc/en/Main/ArduinoBoardUno
- Down load & install Arduino Interactive Development Environment IDE (editor) https://www.arduino.cc/en/Main/Software
- Arduino IDE Installation Instructions http://arduino.cc/en/Guide/Windows
- Arduino Online Frequently Asked Questions FAQ's https://www.arduino.cc/en/Main/FAQ

Articles Model Railroading
MRH Magazine
- Dr Geoff Bunza Arduino Micro Controllers, December 2016 http://mrhpub.com/2016-12-dec/online/
- Dr Geoff Bunza DCC++, March 2017 http://mrhpub.com/2017-03-mar/online/html5/?page=204

DCC Decoder Short Cut Card http:// http://00200530.pdl.pscdn.net/002/00530/MRH04/DCC%20Shortcuts%20Card.pdf

Want more fun?

Using Java Engine Driver 2.13 and a Android smart phone Throttle to operate Engines, throw switches and run accessories.

Java Model Railroad Interface Software JMRI Pro bundle
Free: Runs on a Windows or MAC computers.

1) JMRI 4.6 release or higher: **http://jmri.org/install/WindowsNew.shtml**
http://jmri.org/help/en/html/hardware/dccpp/index.shtml

When you start JMRI DecoderPro it will display a PC based GUI throttle and the engine roster list.
You can operate trains and program mobile decoders from here.

To use Wireless Smart phone Throttles You must down load one of these two smart phone Apps

Engine Driver or WiThrottle Smart phone software - Free: Runs on a Android or iPhone**.**

2) Android use JMRI Engine Driver 2.13 or higher from Google Play store
3) iPhone use WiThrottle Lite 2.1 or higher from Apple iTunes store

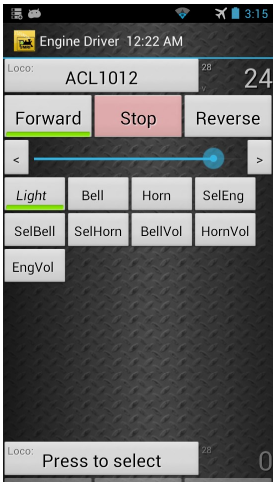
JMRI Engine Driver for Android Smart Phone https://enginedriver.mstevetodd.com/
WiThrottle for Apple Smart Phone https://itunes.apple.com/us/app/withrottle-lite/id344190130?mt=8

Then on your PC from Decoder Pro click Action tab, then WiThrottle Server to start the WiThrottle Server window..
Note the IP address and Port number, look for this 192.168.x.xxx : xxxx. displayed in the WiThrottle Server window.
Enter this number on your Smart phone throttle application.

After downloading the App to the smart phone open it up and in enter the IP number and Port number that was displayed in the JMRI DecoderPro - WiThrottle Server window. example IP 192.168.1.112 port 2048 then press the Connect button.
Pick engine from your JMRI Roster displayed on your phone, and have fun!.

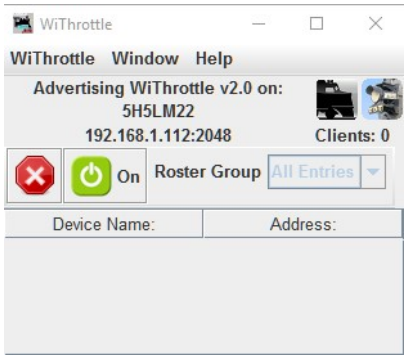
JMRI WiThrottle Server Help http://jmri.org/help/en/package/jmri/jmrit/withrottle/UserInterface.shtml

Engine Driver for Android



WiThrottle for iPhone

WiThrottle Server



from Decoder Pro on your PC

Notes:

Arduino
Reference to Arduino DCC++ 1.2 Building & Setup PDF

JMRI Decoder Pro
Preferences
Connections - set to DCC++, then DCC++ Serial Port, then COMx Port
Defaults - set to one system or the other not mixed between different DCC systems
Display set - throttle to metal or windows
Roster - set to Advanced

Main Track Programming can only 'write' CV's but you can change sounds and speed settings then run &listen to it live.
Programming Track you can 'read & write' CV's but you can not run or hear the sound changes.

PC
Device Manager: set COM Port >properties to 115,200 baud for DCC++

Please make notes here for update ideas or context issues;