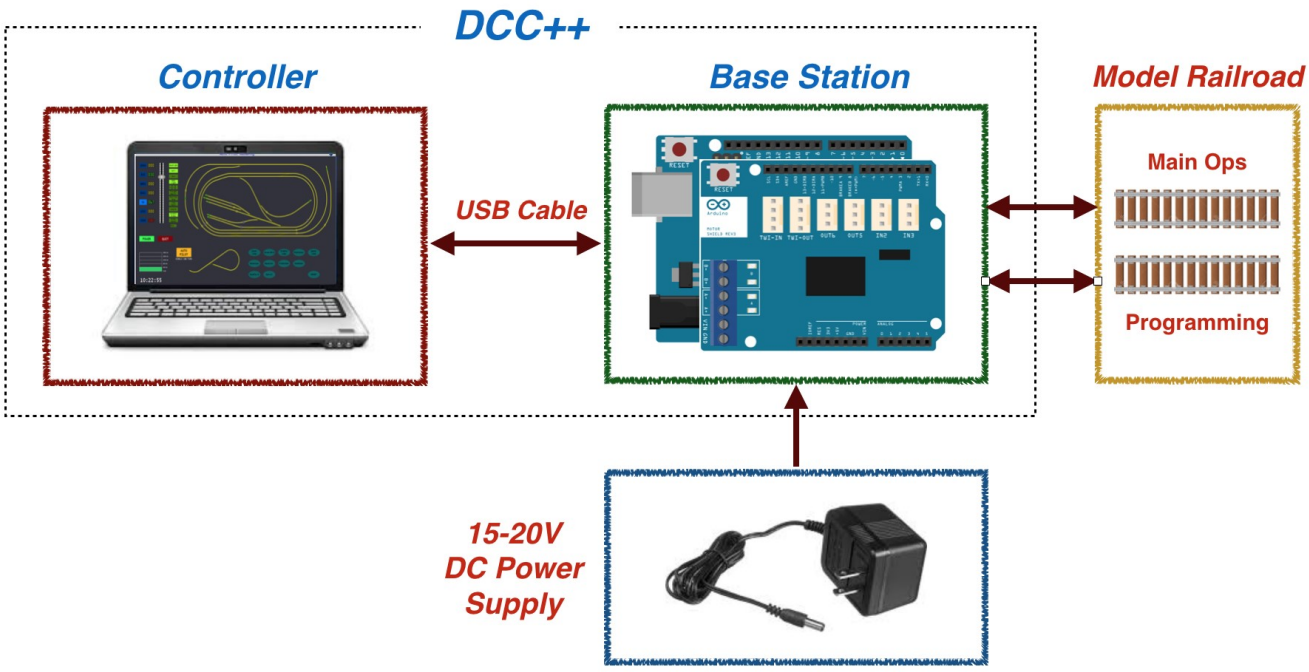
DCC++ A Complete Open-Source DCC Command Station and Interface for Operating Model Railroads

DCC++ Base Station
Arduino Micro Controller

'Build & Setup'



Arduino DCC++ Base Station 1.2.1 - Build & Setup
Version 1.2 April 5, 2017 Kevin C Smith

Introduction

DCC++ is a free open source software solution developed 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++ has a optional custom PC based train Controller written in JAVA language but you can use JMRI Decoder Pro and Engine Driver as our train controller software instead. The DCC++ to JMRI interface was primarily developed by Mark Underwood, (twindad)

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, and by using the features of DecoderPro, PanelPro, Operations Pro and Engine Driver smart phone throttles.

DCC++ Features;

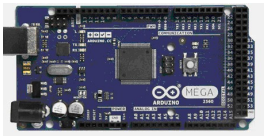
- Programs virtually all NMRA compliant DCC decoders
  - 2-byte and 4-byte locomotive addressing
  - 128-step speed throttling
  - Activate/de-activate all accessory function addresses 0-2048
  - Programming on the Main Operations Track
    - write configuration variable bytes
    - set/clear specific 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 JMRI software
- USB interface 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

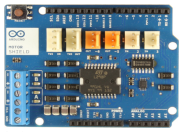
DCC++ Base Station Requirements & Choices:

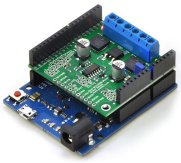
- Arduino Uno 328P R3, or a Arduino Mega 2560 R3 micro controller
- Arduino L298P R3 Motor Shield or a Pololu MC33926 Motor Shield (act as boosters)
- USB A to USB B cable
- Regulated DC Power Supply
- JMRI Decoder Pro from http://jmri.sourceforge.net/ or the USB thumb drive.
- A 3ft length of track for programming and/or test running
- No extra hardware required for programming sound decoders
- Optional 2.1mm Female barrel power connector (Center Positive)for the motor shield
- Optional Clear Acrylic case

The same DCC++ software system is being used by model railroaders worldwide from Z, N, HO and G scales in a variety of configurations. However, there are some choices you can make as to what Arduino hardware and power supply you could use. A over simplified example, for entry level system or just a programming station an Arduino Uno with a Arduino Motor Shield. For a larger layout and more interest in advanced input/output accessory connections servos, switches, sensors etc., a Arduino Mega and a Pololu Motor Shield would fit the bill. Although both configurations could work well in both environments .


Arduino 328P R3


Arduino Mega 2560R3


Arduino L298P R3 Motor Shield


Pololu MC33926 Motor Shield on a Arduino

Once you've chosen the hardware decide which regulated DC power supply you want for Z, N, HO, G scale operations. Suggestion, a 12vdc 500mil amp for Z scale, 12-15vdc 1>amp for N scale and 15-18vdc 2-5amp for HO scale for operations. Recommend for Programming you use a 2 to 5 amp regulated power supply for all scales to insure sound decoder read/writing.

Once you have the DCC++ DCCpp_Uni.ino software loaded onto the Arduino Uno or Mega you can use a variety of DCC controller software like Java Model Railroad Interface, *JMRI to program and operate your locomotives and layout Or, alternatively you can use Gregg Berman's Java Process Controller software as well as other Train Control software to operate trains.

***Please See DCC++ Base Station 1.2.1 and DecoderPro 4.7.1 "Getting Started" guide for more information on Decoder Pro**

Assembling a DCC++ Base Station

DCC++ software was design and developed by Gregg Berman and first released in October 2015. http://sites.google.com/site/dccppsite/
I recommend watching the five DCC++ Base Station videos and also his operation Controller videos for a great visual tutorial.
We will use these components http://**sites.google.com/site/dccppsite/videos/dcc-base-station**
Gregg's custom Controller, http://**sites.google.com/site/dccppsite/videos/dcc-controller** we'll be using JMRI software instead.

Components:
Laptop PC with Windows 7, 8 or 10 and a USB 2.0 connection. Apple OSx and Linux operating systems are also supported.

Hardware:

1) Arduino Uno ATmega328P R3 processor board, *Or* a Arduino Mega 2560 R3 processor board.
2) Arduino L298P Motor Shield 4amp*, (2amps per Main & Programming tracks)
   Or a Pololu MC33926 Motor Shield 6amp* (3 amps per Main and Program tracks)
3) USB A-Male to USB B-Male Printer type cable
4) A 15-18vDC 2 to 5 amp power supply (Center Positive) to power the Arduino Motor Shield which act as the DCC++ "booster".

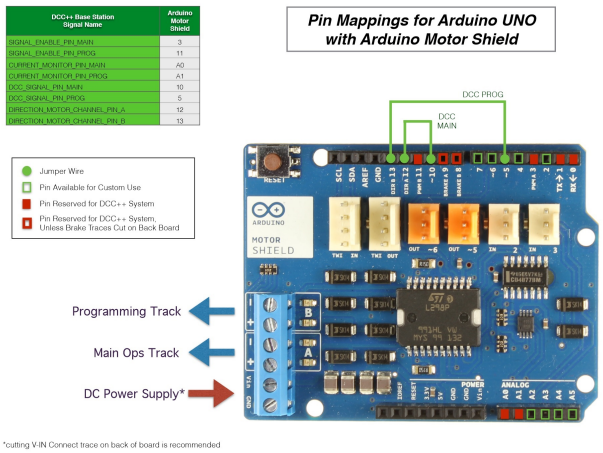
Optional DCC Wifi wireless smart phone Throttles;

5) Free Android (Google) and or iPhone iO (Apple) smart phone for throttles.
6) A Home Wifi network Or, a dedicated one like the TP-Link 702N Personal Access Point Router for Wifi connection of Smart phones to any DCC & JMRI ready System.

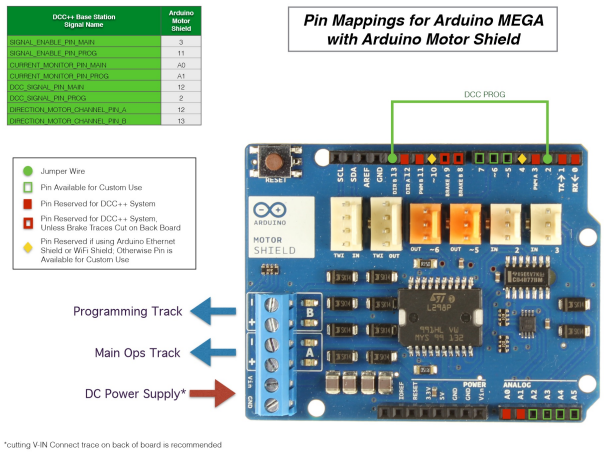
After watching the Videos above , review the Arduino & Motor Shield Documentation pdf revision Jan 17, 2016
https://github.com/DccPlusPlus/Documentation/blob/master/Motor%20Shield%20Pin%20Mappings.pdf
& The DCC++ Wiki home page https://github.com/DccPlusPlus/BaseStation/wiki

Wiring pins for the different Arduino Microprocessors and two different Motor Shields

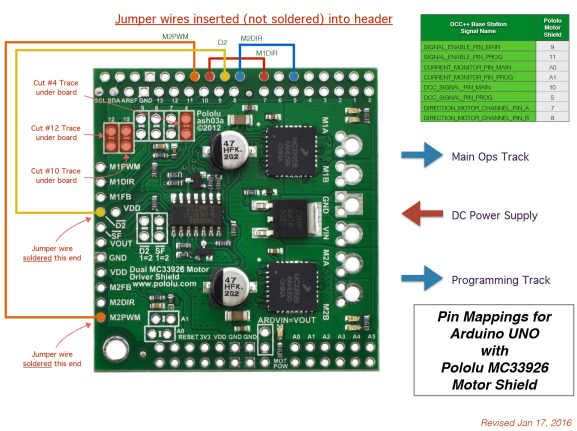
Arduino Uno pin connections

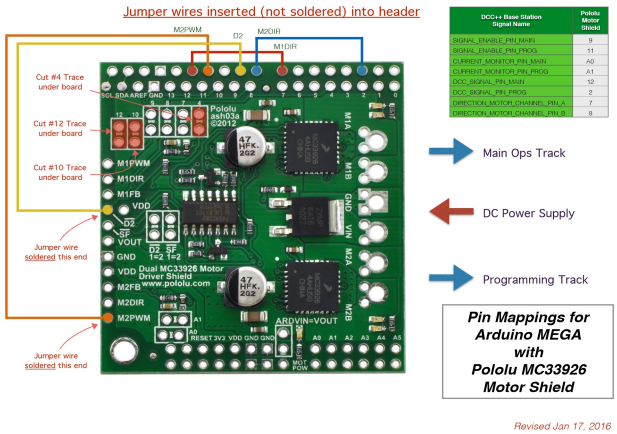


Arduino Mega pin connection



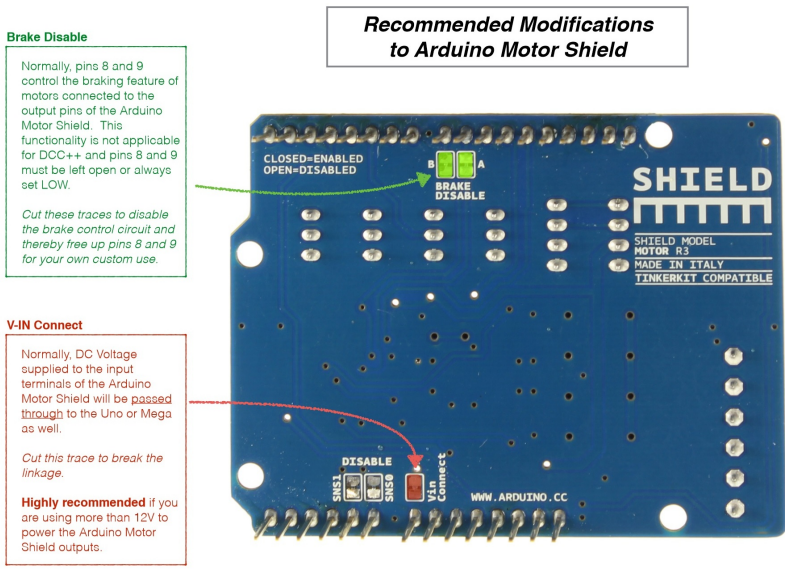
OR If you use a Pololu Motor Shield





It is recommended you use a15-18vdc 2 -5amp regulated power supply for N and HO Scale layouts.
If you are using a DC power supply >12vdc then you need to scratch cut the Vin Trace, voltage line trace on the bottom of the Arduino or compatible L298P R3 Motor Shield board to eliminate any power passing from the top motor shield board back down into the Arduino processor board. If you're using regulated 12vdc or Less to power a Z scale layout then you do not need to do this.

The bottom view of the Arduino L298P R3 Motor Shield



After you've cut the Vin-trace, Carefully place the Arduino Motor Shield on top of the Arduino Uno or Mega and align the pins and press them together. Follow the directions to wire the pin outs depending on which Arduino processor board you are using

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 (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

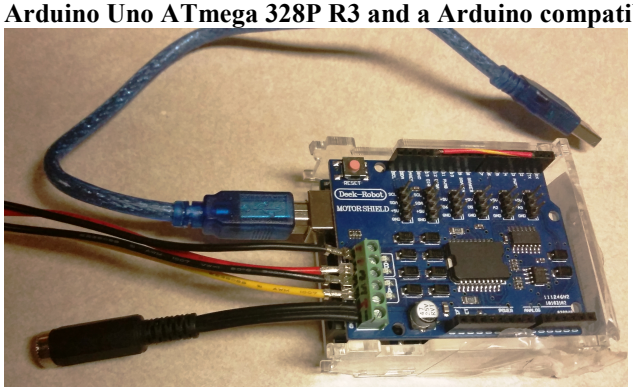
Input Specifications recommend not to exceed 18vdc

Notes:

1. Minimum supply voltage depends upon the requirements of the decoder being programmed and operated. 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 of the power supply but will not work unless the polarity is correct.

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

Arduino DCC++ Base Station assembled view
Arduino Uno ATmega 328P R3 and a Arduino compatible L298P R3 Motor Shield on top.



Notice the blue USB A to USB B cable for the PC connection
a black & red wire in the first two screw blocks B for the Programming Track
a black and yellow wire in the middle two screw blocks A for the Main Track
and a black 2.1mm Female barrel power plug for the separate 15-18vdc (Center Positive) power supply to the Motor Shield.

Also notice there are two smaller gage red/yellow wire in the top of digital pins D5 to D13 & pins D10 to D12 for the DCC pwm pulse width modulation signal. Now that the hardware is assembled we need to load a couple of pieces of software onto a PC.

First the Arduino Interactive Development Environment IDE programmer, it is the Arduino software "editor and compiler" which is used to edit and upload programs called "sketches" like DCC++ from the laptop onto the Arduino micro controller.
Arduino based software, all free: (these procedures may vary because there are so many different web browsers)

First download Arduino IDE from the internet and install it on your PC to dir: \Documents\Arduino folder
Second download DCC++ Base Station Software to your PC dir: \Documents\Arduino folder
Third start Arduino IDE icon and File>Open sketch DCCpp.ino then uploaded DCC++ to the Arduinos via the USB cable.

1) Arduino Interactive Development Environment IDE https://www.arduino.cc/en/Main/Software download and install it.
On your PC click on the Arduino icon and open the IDE programmer.
Then click on Tools and choose the Board: your using, Arduino Uno 328P, or Mega 2560.
Then click on Tools and choose the USB port you are connected to i.e. Port: 4 or 5 or 6 etc.

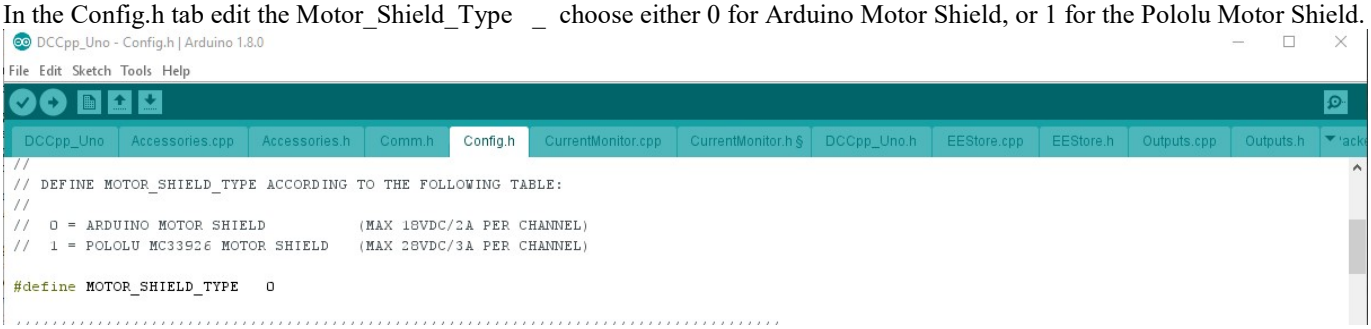
2) DCC++ Base Station software 1.2.1 or higher in the BaseStation-1.2.1.zip file. http://**sites.google.com/site/dccppsite/**
download it to your PC & Unzip folder then copy the DCCpp_Uno folder into the folder 'User\Documents\Arduino
Or an alternate example => **http://github.com/DccPlusPlus/BaseStation/wiki/Getting-Started-With-DCC---Hardware**

Editing then Uploading the DCCpp_Uno sketch from the PC to the Arduino
Now we'll edit, compile and upload it to the assembled Arduino Uno & Motor Shield base station hardware.

3) On your PC click on the Arduino icon which opens the IDE programmer, then click on Files, scroll down to Sketchbook, open the DCC++ Base Station software sketch **DCCpp_Uno.**ino which is the Production version..
Note: If you want the latest DCCpp "Development" version it is in a BaseStation-master.zip file. Unzip & install it instead.

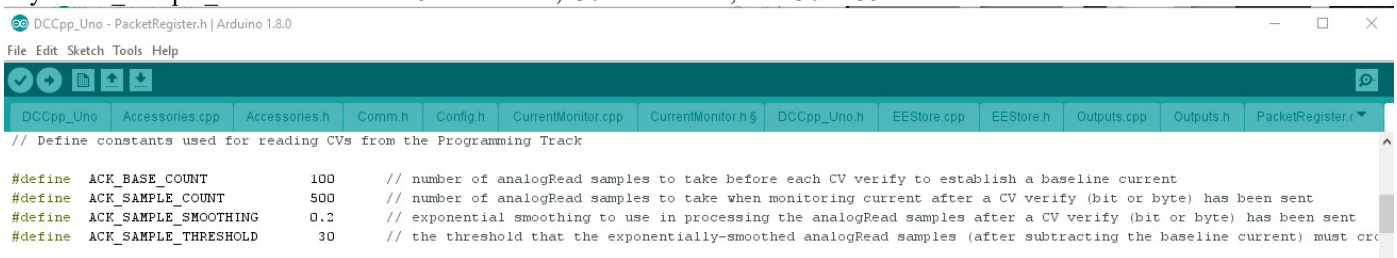
Once open the DCCpp_Uno sketch has many tabs across the top of the program file which allow you to edit features for your specific layout.
When open click on and edit the following file tabs located across the top. Follow the directions to choose whether your using the Arduino L298P Motor Shield or a Pololu MC33926 Motor Shield. Also choose how high you want the Current_Sample_Max to draw until it shuts down if there is an overload or short on the tracks.

In the Config.h tab edit the Motor_Shield_Type _ choose either 0 for Arduino Motor Shield, or 1 for the Pololu Motor Shield.



In the CurrentMonitor.h tab edit the Current_Sample_Max leave it at 300 for N scale or change it to 600 for HO scale

If later you find a particular DCC decoder is hard to read/write to you can try playing with this Acknowledge Sample Threshold.
In the PacketRegister.h tab change Ack_Sample_Threshold from 30 to 35 to try improving Program track response.
Try a Ack_Sample_Threshold set to 15 for Z Scale, 30 for N Scale, and 30 to 35 for HO Scale decoders.



Your done, now click on the button to Compile and Upload it to the Arduino. Then save as the file name DCCpp_Uno
You may be prompted to change the name like DCCpp_Uno2 then save. This is your run & backup file.

Testing the Motor Shield

If you feel you may have an issue you may test the signals to the motor shield by following these directions.

Testing the Arduino and Base Station code **https://github.com/DccPlusPlus/BaseStation/wiki/Diagnostics---D---Command**

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

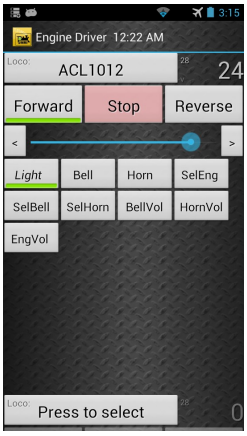
JMRI Engine Driver for Android Smart Phone https://enginedriver.mstevetodd.com/
WiThrottle for Apple Smart Phone https://itunes.apple.com/us/app/withrottle-lite/id344190130?mt=8

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 type 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!.

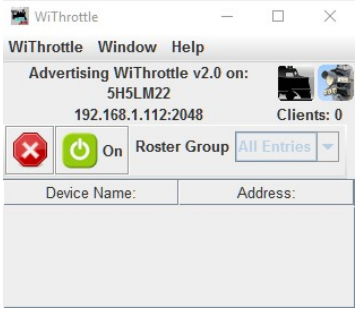
JMRI WiThrottle Server Help http://jmri.org/help/en/package/jmri/jmrit/withrottle/UserInterface.shtml

Engine Driver for Android



WiThrottle for iPhone

WiThrottle Server



from Decoder Pro on your PC

Troubleshooting

Arduino issue;
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.

Fail to compile or upload, check your DCCpp_Uno software edits and make sure you use double slash // to comment out a line.
Check Tools then Board to make sure your uploading to the correct type Arduino, a Uno Or Mega.

Try a different USB cable as yours may be faulty.

DCC++ issue:
There is a DCC++ support page which will be updated to reflect the most common questions people have about DCC++.
DCC++ Development and Support. http://www.trainboard.com/highball/index.php?forums/dcc.177/

One common problem is the configuration of the "Virtual COM Port" for USB. Check 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.

Testing the Arduino and Base Station code https://github.com/DccPlusPlus/BaseStation/wiki/Diagnostics---D---Command

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

Arduino Micro Controllers
Learn Arduinos in about 15 min https://www.youtube.com/watch?v=nL34zDTPkcs
Arduino Micro Controllers https://www.arduino.cc/en/Main/ArduinoBoardUno
Arduino Interactive Development Environment IDE (program 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/

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/
DCC prototype & development projects with Arduino by Dave Bodnar http://trainelectronics.com/miscellaneous_projects.htm
My Experiments with DCC++ http://model-railroad-hobbyist.com/node/25429
DCC++ Facebook page. https://www.facebook.com/groups/1406785379394934/1444943065579165/

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 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

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
Other uses of micro controllers in Model Railroading http://modeltraincontrols.com/
DCC Decoder Short Cut Card Http:// http://00200530.pdl.pscdn.net/002/00530/MRH04/DCC%20Shortcuts%20Card.pdf

DRAFT #1 Arduino DCC++ Base Station 1.2.1 - Build & Setup
Version 1.2 April 5, 2017 Kevin C Smith

Notes:
Arduino

JMRI Decoder Pro
Reference DCC++ Base Station 1.2.1 and DecoderPro 4.6 & 4.7.1 - Getting Started

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;