Arduino DCC++ Base Station v1.0

The DCC++ Base Station software is a C++ *sketch* designed for an Arduino Uno micro-controller fitted with an Arduino Motor Shield. This sketch enables the Uno to serve as a full-function DCC base station that can be connected directly to the tracks of a model railroad to drive and control DCC-equipped locomotives and DCC stationary decoders.

For a complete overview of the system, as well as detailed instruction videos, please see my DCC++ YouTube channel: https://www.youtube.com/channel/UCJmvQx-fe0OMAIH-_g-_rZw

Command and Communication

Communication to and from the Uno is through standard serial interface protocols via a USB cable connected to a Windows PC, Mac, or Linux system. A Bluetooth transceiver can optionally be connected to the Uno's serial pins to enable wireless serial communication with no modification to the sketch.

Commands and data sent to and from the DCC++ sketch are of the following form:

```
< C P1 P2 ... PN >
```

where *C* is a single alphanumeric command character and *P1* through *PN* are optional alphanumeric parameters.

White space between the initial opening bracket < and the command character *C* is permitted but not required. White space between the command character *C* and the first parameter *P1* (if there is one) is permitted but not required. White space between the last parameter *PN* (if there is one) and the closing bracket > is permitted but not required. White space between parameters (e.g. *P1* and *P2)* is required.

Characters sent to the sketch that are not enclosed in brackets are ignored. Only the first 20 characters after an opening bracket is received by the sketch are recorded and parsed. Any additional characters sent before the closing bracket is received by the sketch are ignored to prevent an inadvertent overflow of the parsing buffer. None of the current commands should require more than 20 total characters.

Users can manually send commands to, and receive data from, the sketch running on an Uno using the Serial Monitor that is part of the Arduino IDE software. Generic Serial Window utility programs can also be used instead of the monitor that comes with the Arduino IDE. This sketch sets the baud rate of the Uno's serial connection to its maximum value of 115200. You may change it to a lower value if desired but always ensure it matches the baud rate set for whatever program you use to communicate with the Uno.

Operating the sketch by manually typing commands into a serial window is a very good way of learning how the sketch works, understanding the data it sends back, testing new commands

and parameters, and verifying proper connectivity to the tracks and trains. However, using the DCC++ Base Station in this fashion is not very practical for actually running a full model railroad.

Instead, the sketch should ideally be driven by a separate control/interface program that provides users with an intuitive GUI that constructs Base Station commands behind the scenes and transmits them to the Uno as needed. This same GUI should also process and act on any output sent back from the Base Station. DCC++ Controller, written in Java using the Processing IDE and graphics library (https://www.processing.org), is one such GUI. It is separately available under an open-source license as part of the overall DCC++ system.

Installing and Using DCC++ Base Station

The entire DCC++ Base Station sketch is stored in single folder named DCCpp_Uno, and consists of 6 C++ files, and 6 associated header (*.h) file as follows:

DCCpp_Uno.ino:	Main entry point for the sketch. Declares global variables, configures timers and interrupts, and contains the required setup() and loop() subroutines.
SerialCommand.cpp:	Parses and processes text commands received on the serial port.
CurrentMonitor.cpp:	Measures the current drawn from both the main operations track and the programming track and monitors for short circuits.
Accessories.cpp:	Allows for the optional creation of a list of turnouts controlled by stationary DCC decoders. The direction of turnouts specified in this list are updated and stored in the Uno's EEPROM for retention after power is shut down.
Sensor.cpp:	Allows for the optional creation of a list of train-detection sensors that are directly connected to unused pins on the Uno.
PacketRegister.cpp:	Contains all the code that creates NMRA DCC bit packets based on the commands parsed by SerialCommand.cpp.

To open the sketch within the Arduino IDE, open the main entry point file DCCpp_Uno.ino. The Arduino IDE will automatically open all the other files found in the same directory.

Follow the IDE instructions to compile and upload the sketch into an Arduino Uno.

The DCC++ Base Station sketch has been fully tested with Arduino's latest IDE version 1.6.5.

Each of the 12 sketch files are heavily commented. In particular, SerialCommand.cpp contains complete descriptions of every DCC++ Base Station command, including all input parameters and return parameters. A summary of these commands (without listing the parameters) is as follows:

<t>: sets the throttle for a mobile engine decoder using 128-step speeds
<f>: controls mobile engine decoder functions F0-F28
<a>: controls stationary accessory decoders
<T>: controls turnouts connected to stationary accessory decoders
<w>: writes a configuration variable byte to an engine decoder on the main ops track
<b>: sets/clear a configuration variable bit in an engine decoder on the main operations track
<W>: writes a configuration variable byte to an engine decoder on the programming track
<B>: sets/clear a configuration variable bit in an engine decoder on the programming track
<R>: reads a configuration variable byte from an engine decoder on the programming track
<1>: turns on track power
<0>: turns off track power
<c>: reads current draw from main operations track
<s>: returns status messages, including power state, turnout states, and sketch version

One Step at a Time

If you don't have an Arduino Uno, you can still download the Arduino IDE for free from

http://arduino.cc

and open/explore the DCC++ Base Station sketch code and read through the comments. You can even compile the sketch though without an Uno you cannot load it, and therefore can not run it.

If you have an Uno but do not have an Arduino Motor Shield you can still load and run the software on the Uno, and test out the serial connectivity and command data flow, but you cannot connect the Uno directly to the tracks of a model railroad.

If you have an Uno AND a Motor Shield, but do not have a power supply, go buy a power supply, at which point you will then have a complete DCC++ Base Station ready to connect to a model railroad. As mentioned above, you'll probably want to download and modify the DCC++ Controller GUI for your own use, since manually typing commands into the Serial Monitor of the Arduino IDE gets tiring very fast.

Enjoy!