Conductors and many Tinkerers will prefer to follow these steps to install our DCC-EX Library and the DCC-EX Command Station. There is more information at the end for advanced Tinkerers or Engineers who want to see the software and perhaps help with the development or make changes for themselves. Links to the GitHub repositories are there.

## Conductors and Tinkerers

1. Make sure you download and install the Arduino IDE if you don't already have it.

NOTE: This is a 192Mb download.

<https://www.arduino.cc/en/Guide>

2. Download the "Arduino Timers" library here:

<https://github.com/davidcutting42/ArduinoTimers/archive/master.zip>

3. Download the DCC-EX library file here:

<https://github.com/DCC-EX/CommandStation/archive/master.zip>

4. Download the CommandStation Sketch here:

<https://github.com/DCC-EX/CommandStation-DCC/archive/master.zip>

5. Unzip the two library files (ArduinoTimers-master.zip and CommandStation-Master.zip) one at a time to your desktop or downloads folder. You should then have these two folders after unzipping:

ArduinoTimers-Master  
CommandStation-Master

6. Rename these two folders to remove the “-master”. In other words the folders should now look like this:

ArduinoTimers  
CommandStation

7. Select both the above mentioned folders and move them to your Arduino libraries folder, usually:

\Documents\Arduino\libraries\

You will see other folders there which are other libraries for the Arduino. Your new library folders will live happily next to the others.

8. Unzip the CommandStation-DCC.zip file to your download or desktop folder and then move it to your Arduino projects folder, which should be:

/documents/Arduino

You may notice other project folders here, and should see the libraries folder where we just copied your libraries to.

Rename this folder like you did with the library folders to remove the “-master” so that the folder name is simply, “CommandStation-DCC”.

9. Open the Arduino IDE program from the operating system menu on your computer. Then click on “File -> Open” and navigate to your projects folder (/documents/Arduino) and find the CommandStation-DCC folder. Click on that folder to open it and find the main project file. All Arduino main files end in .ino (as in Ardu-INO) and should have the Arduino logo on them to advertise their association with the Arduino IDE. Click on “CommandStation-DCC” to open the project in the IDE>

The project will open either in a new window or the same window depending on your settings. If there are two windows, you can close the first window if you like because we won’t need it anymore. You will see tabs for the 3 files used by CommandStation-EX running across the top of the Arduino IDE.

10. Connect a serial cable from your computer to your Arduino Command Station. You should see lights blink on the Command Station board and get an acknowledgement from your computer that you just plugged something in.

11. Select “Tools -> Board” from the Arduino IDE menu and find your board. You should be using one of the following:

* Arduino Mega or Mega 2560
* Aduino Uno
* Arduno Nano

12. Choose the serial port your operating system assigned to your Arduino board under “Tools -> Port”. The IDE should automatically detect the correct port, but you may have to select it if you have more than one Arduino board plugged in or have more than one you plug in at different times. Once you select a board, the system will remember that port for that Arduino.

13. The last step is to upload the sketch to your Arduino. Click the “upload” button at the top left that looks like a right arrow. It is next to the checkmark button that lets you compile and verify the program separately. This test is conducted before uploading and will flag any errors anyway. Besides, those of us on the DCC++ EX team never make mistakes, so it should compile and upload perfectly!

The compile and upload process should take between 30 seconds and 2 minutes. Once complete, you should be able to run you DCC++ EX Command Station!

## Testing

The Arduino IDE has what is called a “serial monitor” built in. It allows us to monitor things connected to the serial port. DCC++ EX has messages and debugging included. With the Command Station still connected to your computer via the USB cable and the correct port selected, choose “Tools -> Serial Monitor” from the menu. A new window will open and you should see a line of text giving you status information that looks like this:

DCC++ EX Command Station V2.1.1<NO:SERIAL>

This lets you know everything is working. As another quick test, after making certain that your connections to the track are correct and that there are no shorts, you can type:

<1>

Into the serial monitor command box at the top, to the left of the “send” button and press send. If your motor controller has LED indicators, they should light and you will have power to the track. If you have a locomotive with a sound decoder on the track, it should start to make idle sounds. To turn off the power, simply enter:

<0>

And hit the send button. You are finished. To run trains, you simply need to connect a controller like a throttle or JMRI. Happy Trails! Or Tracks, in this case.

## For the Engineers

Browse the code or use these links to clone the DCC-EX Command Station files. Create repositories:

The command station library file:

https://github.com/DCC-EX/CommandStation.git#master

The Arduino Timers libary file:

https://github.com/davidcutting42/ArduinoTimers.git#master

The main Command Station Sketch:

https://github.com/DCC-EX/CommandStation-DCC.git#master

To clone, use GitHub Desktop or Git Bash. In git bash, navigate to where you want to clone the repository and from the "$" prompt, type "git clone https://github.com/DCC-EX/CommandStation.git" and <https://github.com/DCC-EX/CommandStation-DCC.git>

## Troubleshooting:

The Com port under “Tools -> Port” is grayed out.

You don’t see anything from the serial monitor when you open it

## For the Dev team

Question: Can we zip up all the files into one zip file?

Question: Are we going to need detailed instructions on how to use an unzip program? Pictures? It will be pages long!

Question: What are we going to call these files? How do we refer to the library? DCC-EX Library? DCC-EX CommandStation Library?

Question: Is number 7 correct for Mac and Linux? What are their folder paths?

NOTE: We need to rename the repos. the library has 'master' appended to the end of it. In the Arduino IDE, we unzip and have folders with –master tacked on and that looks terrible. Step number 6 is confusing and we should not have to do it.

Question: How many definitions do we need? Do we need to say what a library is? What the Arduino IDE is? Etc.

My Mega build shows:

Sketch uses 22656 bytes (8%) of program storage space. Maximum is 253952 bytes.

Global variables use 450 bytes (5%) of dynamic memory, leaving 7742 bytes for local variables. Maximum is 8192 bytes.

My Uno Build Shows:

ketch uses 21564 bytes (66%) of program storage space. Maximum is 32256 bytes.

Global variables use 416 bytes (20%) of dynamic memory, leaving 1632 bytes for local variables. Maximum is 2048 bytes.