



Serial Port Utilities

April 2023

© 2023 Dilithium Design

Contents

Overview	2
Windows - PuTTY App Installation.....	2
Mac OSX - Coolterm App Installation.....	7
Android – FTDI UART Installation	12
Firmware Upgrade – 8-bit devices (EVCC, BMSC, VCU, GFM).....	15
Firmware Upgrade – MCU.....	17
Warrantee and Support	19
Document History.....	19

Overview

Dilithium Design devices such as the MCU, EVCC and BMSC use a serial port for configuration and for firmware upgrade. A 5-volt USB to serial cable connects a host computer USB port to the device using a 3.5mm TRS jack (like a headphone jack). The same cable can be used for all Dilithium Devices. The FTDI TTL-232R-5V-AJ cable is recommended, and can be purchased from the Thunderstruck website (<https://www.thunderstruck-ev.com/serial-cable.html>).

A terminal application (or “telnet client”) is used on a Windows, Mac, or Android host computer, to create a simple text interface to Dilithium devices. Firmware upgrades require a separate application – see *Firmware Upgrade* below.

The steps to install the necessary supporting software are similar for all host computers. Terminal application installation instructions for Windows, Mac OSX and Android are included below. In the case of a Windows host computer, it may also be necessary to install Virtual Com Port (VCP) Drivers.

There are good tutorials available on the Internet on how to install terminal application software and the VCP Drivers. Search for “FTDI installation”, “putty installation”, etc. For example, there is a YouTube tutorial covering “Putty” installation on a Windows PC (link below).

<https://www.youtube.com/watch?v=j4K6ApxwYXY>

Windows - PuTTY App Installation

PuTTY is a terminal application that provides a text user interface for external serial-connected devices. The following steps describe installation and configuration of PuTTY for connection to devices using a Windows computer. PuTTY is just one application of this type and there are other suitable telnet clients that may be used.

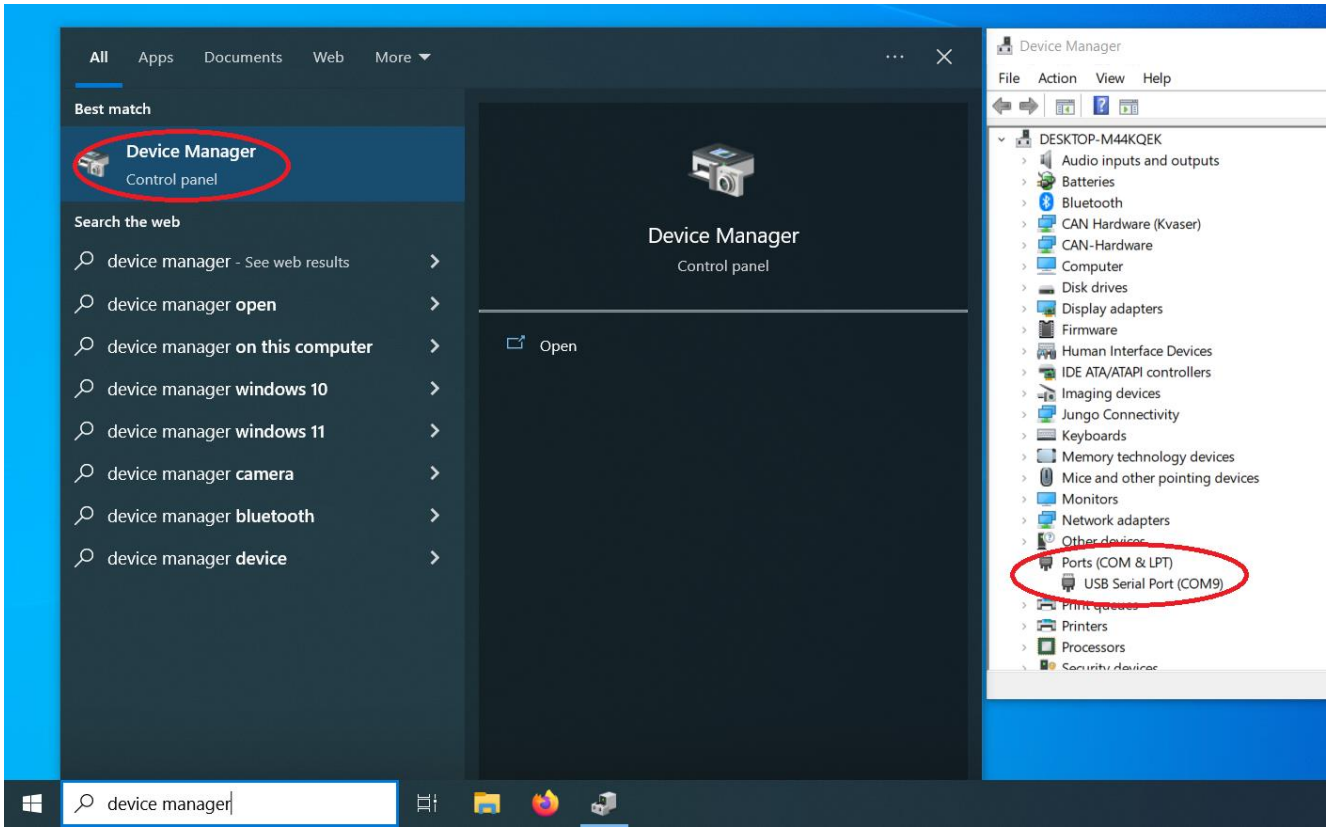
1. Install PuTTY on the host computer from <http://www.putty.org/>.
2. Plug in the serial cable to a USB port on the computer. The serial port cable has a built-in USB to serial converter and a unique serial number. Windows assigns a COM port name which identifies the serial cable. VCP Drivers are required: they may have already been installed or may be installed automatically when the serial cable is connected. If the VCP drivers are not installed, then there will be an “unrecognized device” error when the cable is plugged in, and it will be necessary to install the VCP drivers manually.

VCP drivers are available at: <https://ftdichip.com/drivers/vcp-drivers/>
Installation documentation is available at <https://ftdichip.com/document/installation-guides/>

3. Determine the Windows COM port name which identifies the cable. If this is the first time that the serial cable has been connected to the computer, Windows will assign a new virtual “COM Port” name. The COM port name is of the form “COM<n>”, where n is a small number.¹

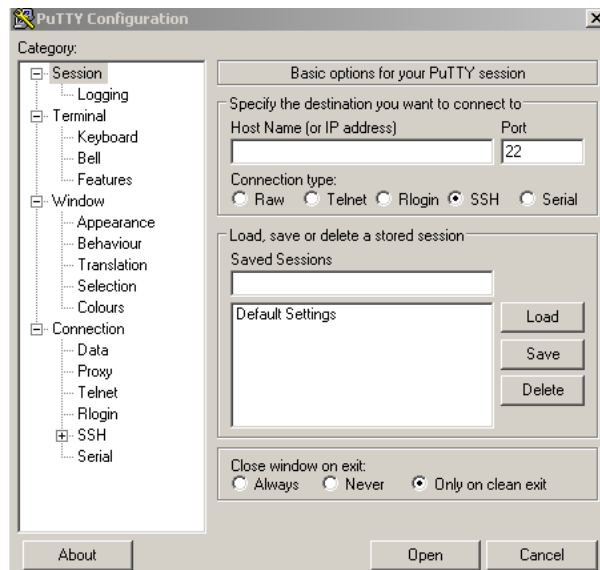
The COM port name assignment can be determined by going into the Windows “device manager” application and inspecting the list of serial port devices present on the computer. Use Windows search to find the Device Manager (or search in the Control Panel). Open the “Ports (COM & LPT)” to determine the port number assigned to the serial cable (COM9 in the screenshot below).

¹ Because each cable has its own identifier, it is possible to use multiple cables at once. Each cable is plugged into a different USB ports or USB hub. Multiple PuTTY sessions can be used at the same time.

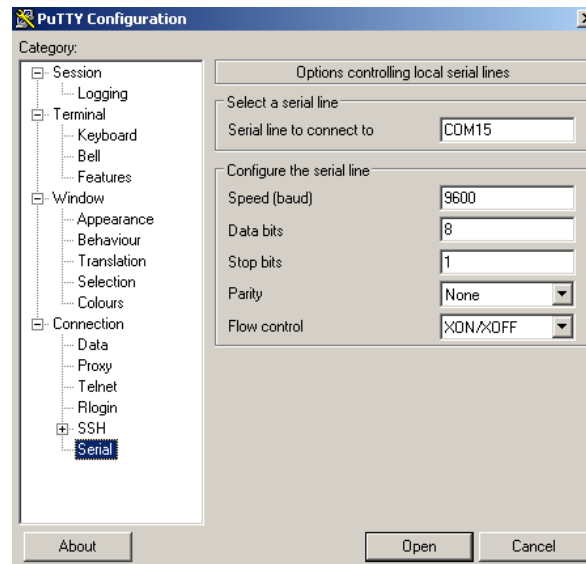


Windows releases vary on how they organize the hardware devices and so the screen above may not apply. If further help is needed, do an internet search “how to identify Windows com port.”

4. Launch the PuTTY application. The port name “COM15” is used in the example below. The initial screen will display:



5. Configure the Serial cable. Select “Serial” in the Category column. The following screen will appear:

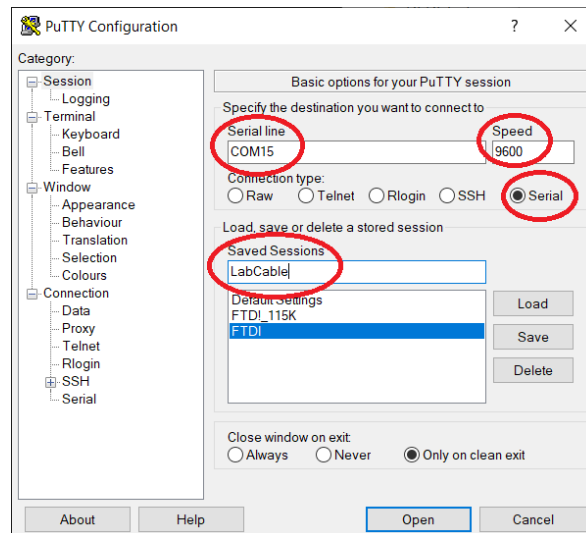


Set **Select a line to connect to** to the COM port that Windows assigned to the cable in the field, e.g., COM15.

Set **Speed** to the default serial baud rate. A value of 9600 is suggested here. This default value may be overridden on a per-session basis.

Leave **Data bits**, **Stop bits**, **Parity**, and **Flow Control** to the default settings (but verify they match the example above).

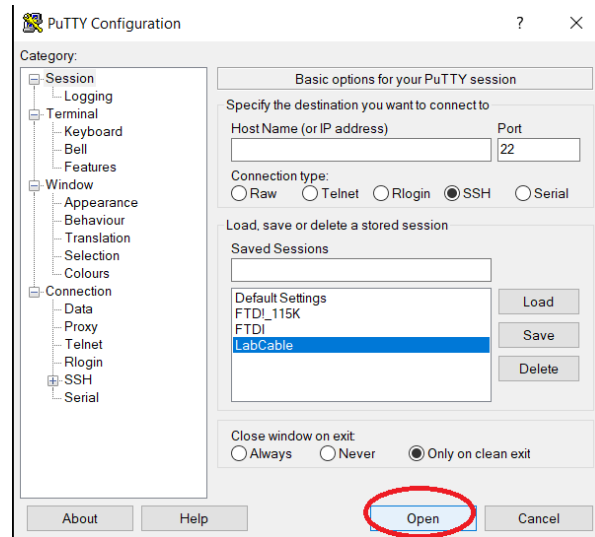
7. Create a PuTTY session. Return to “Session” by clicking the word **Session** in the left side Category window. The following screen will appear:



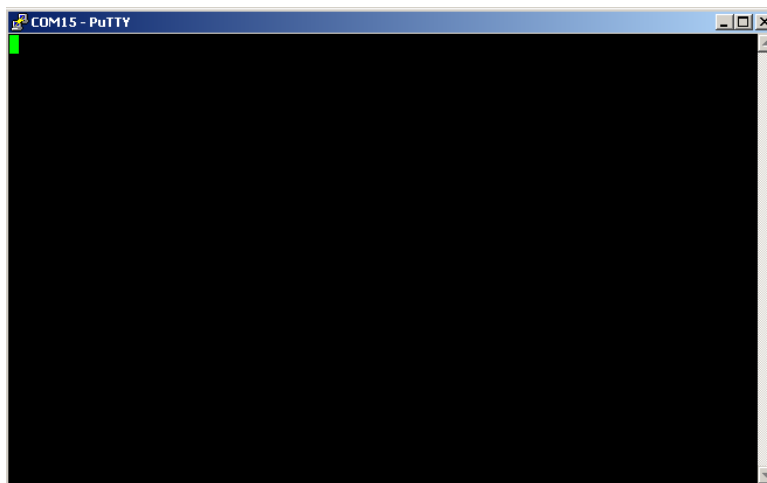
Verify the following settings:

- Set **Connection type** to “Serial.”
- Set **Serial line** to the COM port number (COM15, in the example).
- Set **Speed** to either 9600 or 115200. The EVCC, BMSC, and VCU support a baud rate of 9600, and the MCU supports a baud rate of 115200
- Set **Saved Sessions** to a name. In this case “LabCable” is used.
- Press **Save** to save the session. Once saved, PuTTY is ready for use.

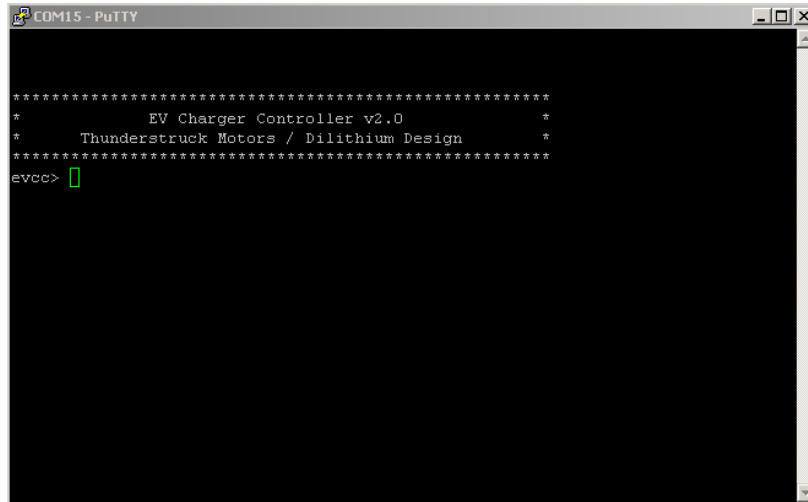
5. Select a saved session and then press **Open**.



A screen like the following should appear. Click the computer cursor into the PuTTY screen, and the green cursor in the corner will fill in solid.



6. Plug the serial cable jack into the device. In this example, the device is the EVCC. When the device is powered up, a startup banner similar to the following is displayed. Note the device firmware version is shown in the banner.




```
COM15 - PuTTY
*****
*           EV Charger Controller v2.0           *
*   Thunderstruck Motors / Dilithium Design   *
*****
evcc> █
```

The device user interface is now available for changing settings and running reports. Type “set” for a list of commands. See the Thunderstruck website for links to specific device manuals which include information about using commands.

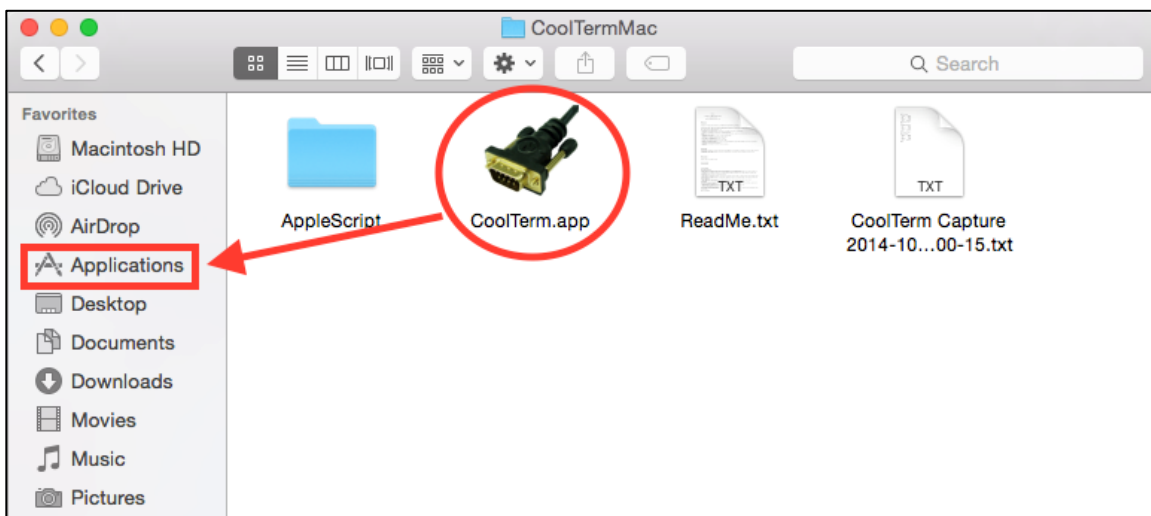
Mac OSX - Coolterm App Installation

When using Mac OSX, the CoolTerm application is recommended.

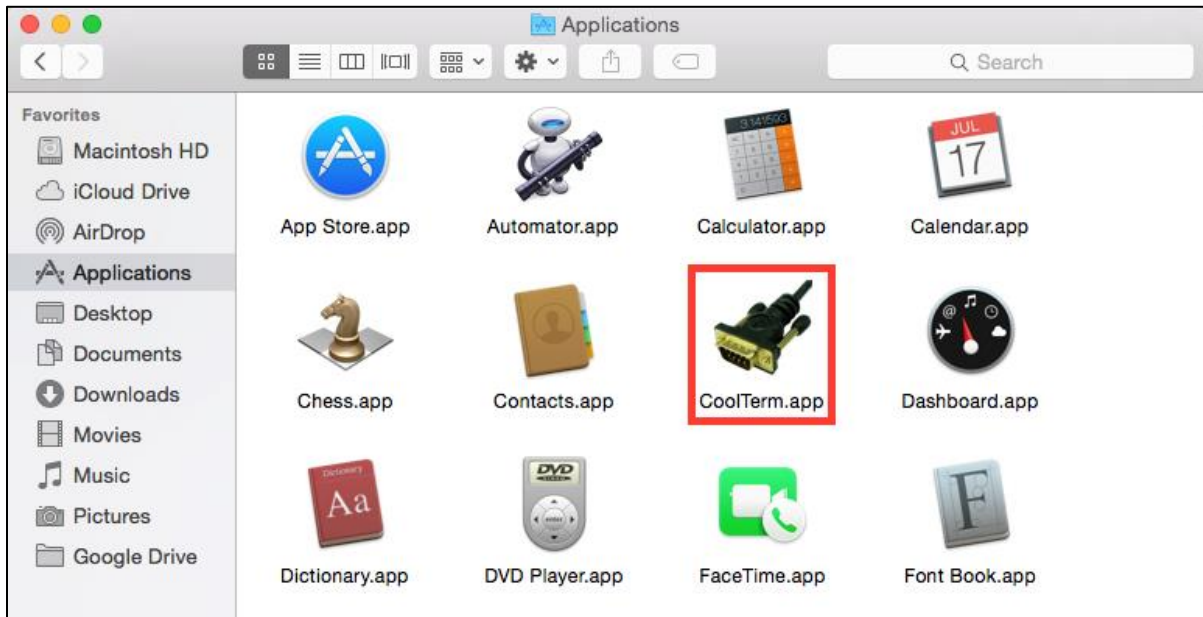
1. Download CoolTerm from <http://freeware.the-meiers.org>
2. Select the applicable download for the appropriate processor for the host computer.

Application	Version	Description
<p>CoolTerm</p>  <p>macOS: <input checked="" type="radio"/> Intel/ARM</p> <p>Win: <input checked="" type="radio"/> Intel 64Bit <input checked="" type="radio"/> Intel 32Bit <input checked="" type="radio"/> ARM 64Bit</p> <p>Linux: <input checked="" type="radio"/> Intel 64Bit <input checked="" type="radio"/> Intel 32Bit</p>	<p>2.0.1 12/20/2022</p>	<p>CoolTerm is a simple serial port terminal application (no terminal emulation) that is geared towards hobbyists and professionals with a need to exchange data with hardware connected to serial ports such as servo controllers, robotic kits, GPS receivers, microcontrollers, etc. Written in Xojo.</p> <p>32-Bit Builds: Starting with version 1.6.0, the default for all platforms (except Raspberry Pi) is 64-bit. Note that version 1.7.0 is the last 32-bit build for macOS. All newer versions will only be available as 64-bit builds.</p> <p>LINUX and Raspberry Pi: The LINUX and Raspberry Pi versions are not "officially" (meaning: "not well") supported. While almost everything is expected to work as expected, only minimal testing using virtual machines has been performed to confirm that all the features work properly. The LINUX and Raspberry Pi builds have been posted here as a courtesy to the users that asked for it. Please use these builds at your own risk. Please use the forums to share your experiences with other users.</p> <p>OS X Universal Binary (PPC/Intel): v1.4.7 is the last version of CoolTerm available as a universal binary supporting OS X 10.6 or older, click here to download.</p> <p>Windows XP: Starting with v1.4.5, the Windows build will only support Windows 7 and newer. v1.4.4 is the last build that supports Windows XP. It can be downloaded here.</p> <p>Older Versions: Older versions of CoolTerm can be found here.</p> <p>Books that mention CoolTerm (AUTHORS: If you would like make a contribution to the "CoolTerm Library" by donating a signed copy of your book, it would be greatly appreciated. :-)):</p> <ul style="list-style-type: none"> • Building Wireless Sensor Networks by Robert Faludi • Making Things Talk, 2nd Edition by Tom Igoe • Arduino Cookbook, 2nd Edition by Michael Margolis

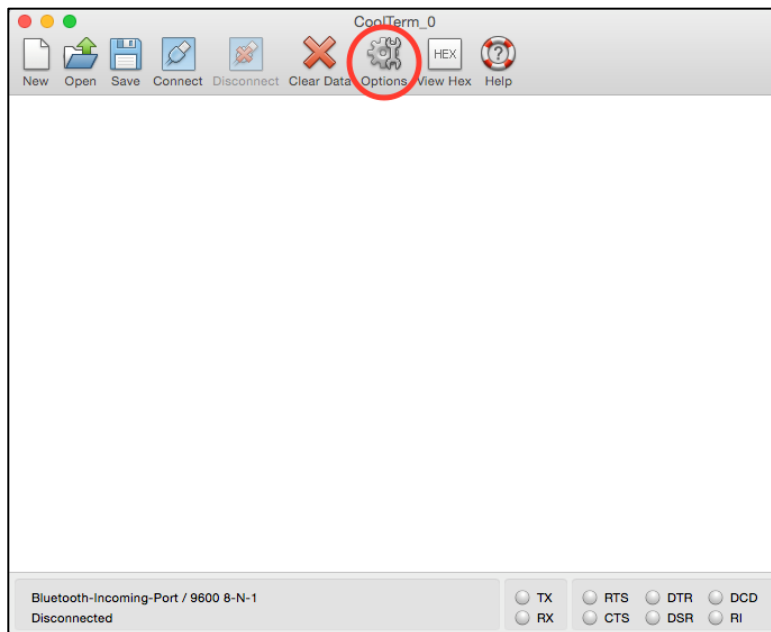
3. Extract the .zip file, open the CoolTermMac folder and drag the CoolTerm app into the applications folder.



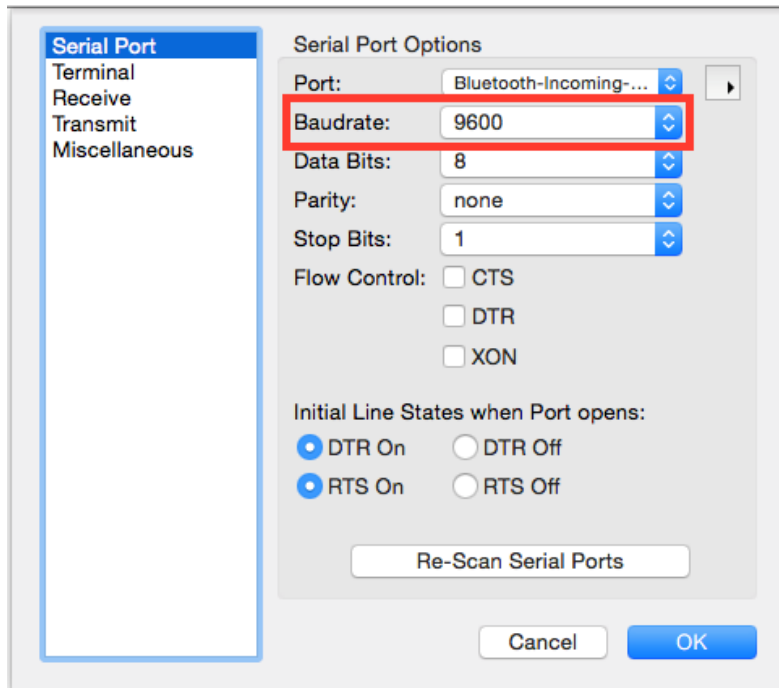
4. Plug in the USB to serial cable. Open the applications folder and double click CoolTerm.app



5. Click Options

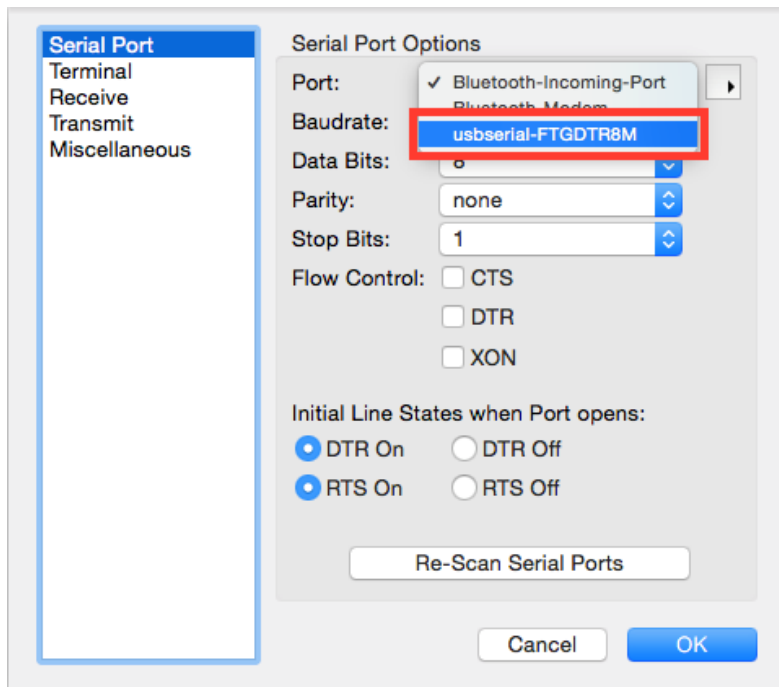


6. Set the **Baudrate** for the device being used. Use 9600 for EVCC, VCU and BMSC; **115200 for MCU**.

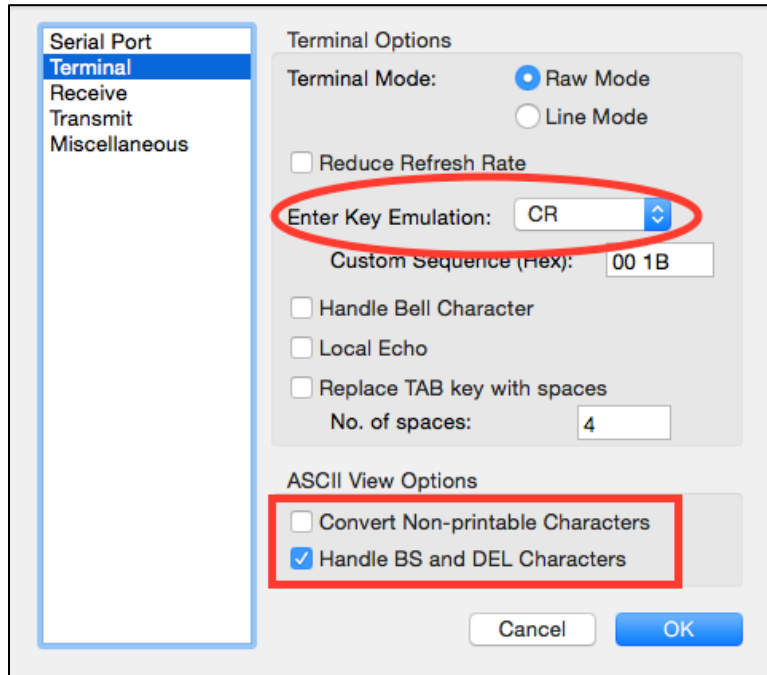


7. Click the drop-down menu and select “usbserial-<sn>” where <sn> is the specific number for the serial connection as discussed earlier. (For MAC OS X, the virtual serial port device name is of the form “usbserial-<sn>” where <sn> is the serial number of the USB to serial connection. For example: **usbserial-FTGDTR8M**).

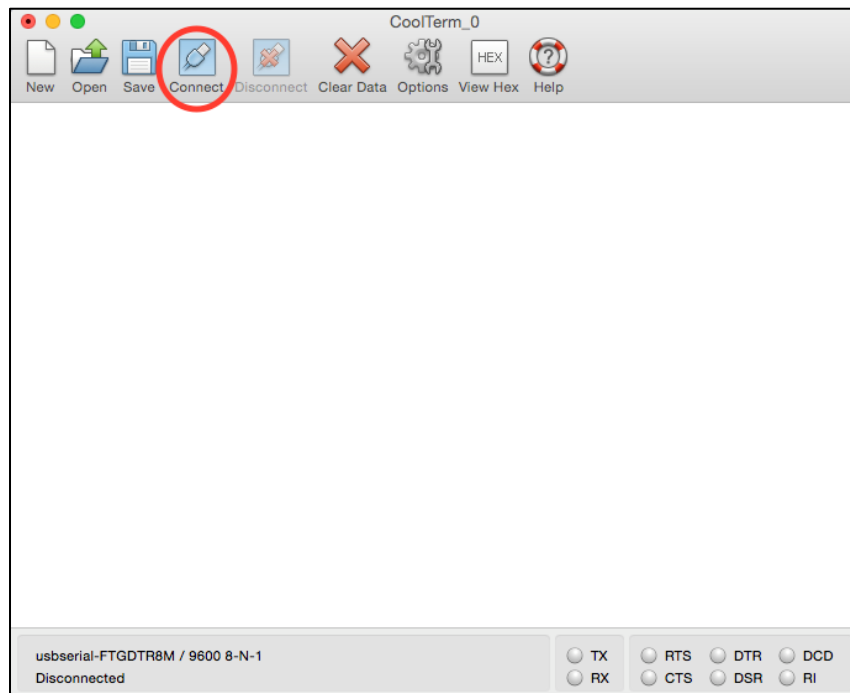
Note that the usbserial-<sn> will not show up in the dropdown menu if the USB is unplugged prior to starting the program. If this occurs, exit CoolTerm, plug in the USB cable and restart CoolTerm.



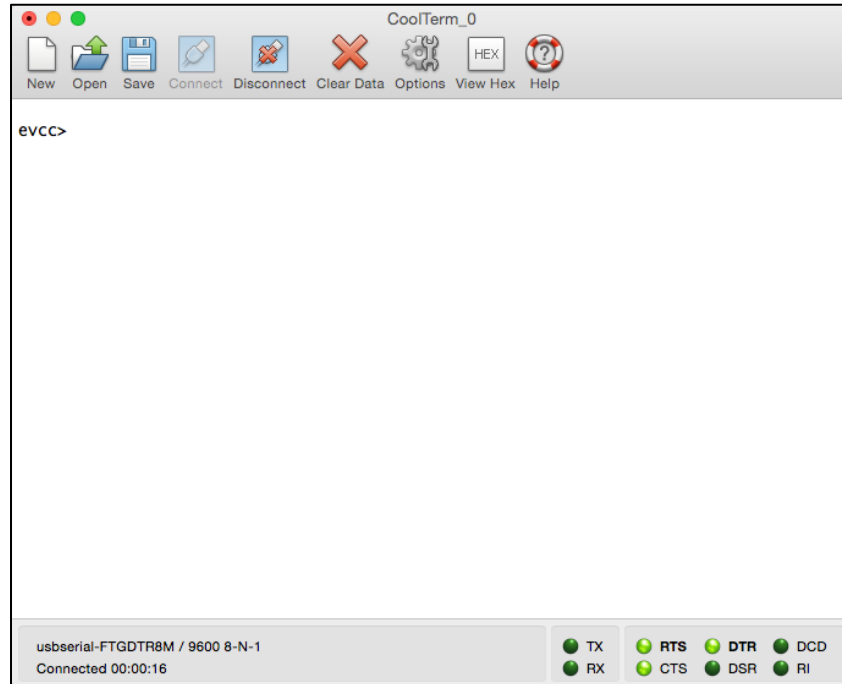
8. Go to the left-hand column and click **Terminal**. Then change the window to match the settings below.



9. Click **Connect**



10. Press the “return” key, the device command prompt should come up (EVCC Shown). Ensure that 12V power is hooked up to the device (MCU, EVCC, BMSC), and that the USB to serial cable is plugged into the device.



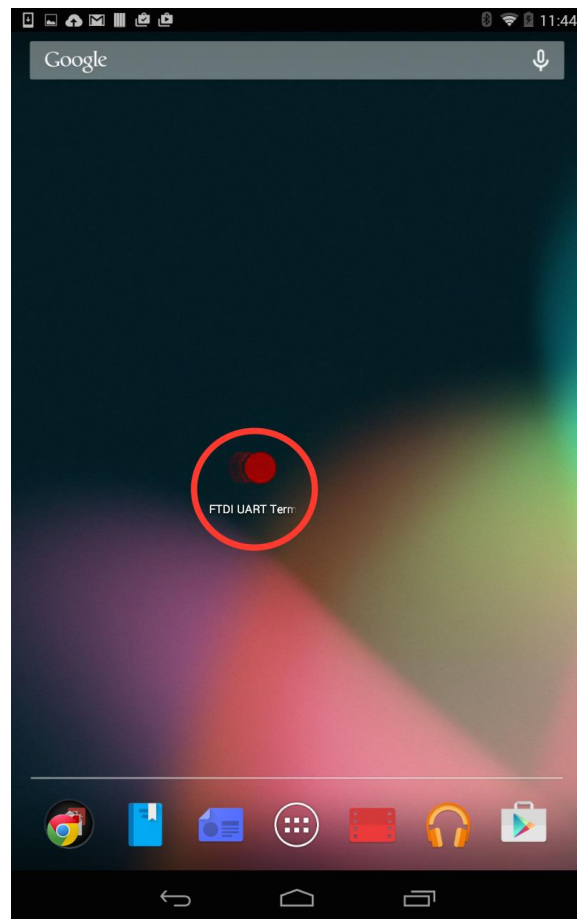
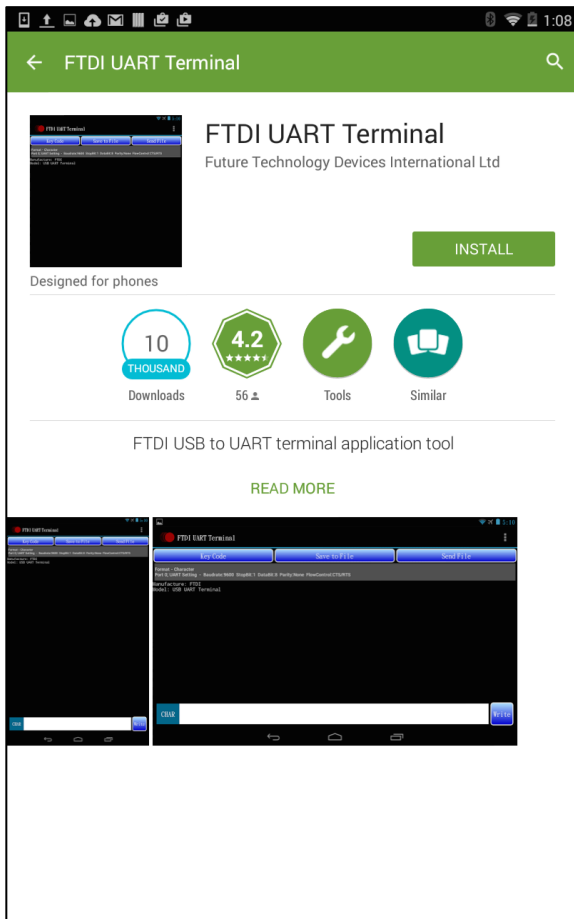
Note: Although the operation of the serial port is very similar to the Windows examples, above, there is one important difference. Windows keyboards generate an ASCII “DEL” character when a “delete” is pressed. MAC keyboards generate an ASCII “BS” character. Current EVCC firmware only interprets the DEL key and the MAC “delete” key may not work as expected. However, the ASCII “DEL” character can usually be generated by MAC keyboards (look for another “delete” key with an “x” or try pressing FN-DEL).

Android – FTDI UART Installation

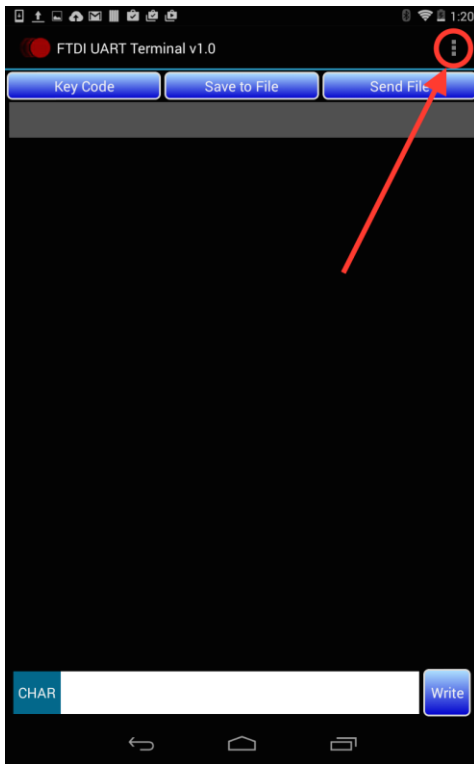
In this example, an ASUS Nexus 7 Tablet is used. The same (or very similar) process should apply to other Android tablets and smart phones.

In addition to the FTDI USB to serial cable, a USB cable to the On-The-Go (OTG) cable is needed. The OTG USB female to USB micro adapter can be found here: <http://www.amazon.com/IVSO%C2%AE-Nexus-Tablet-Micro-Cable/dp/B00932N46S>

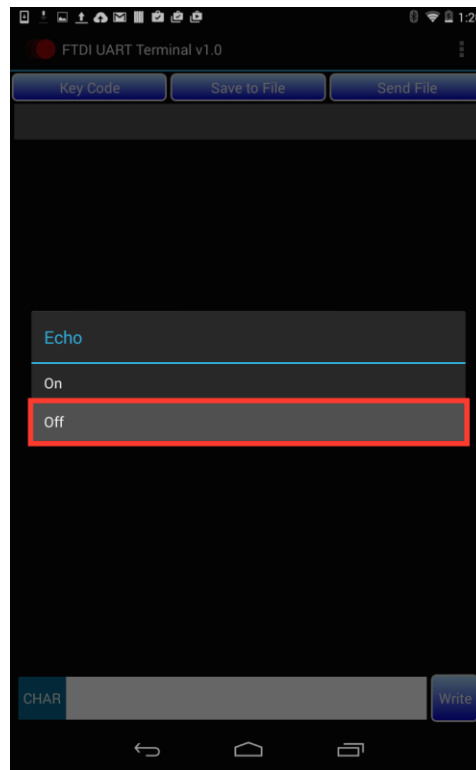
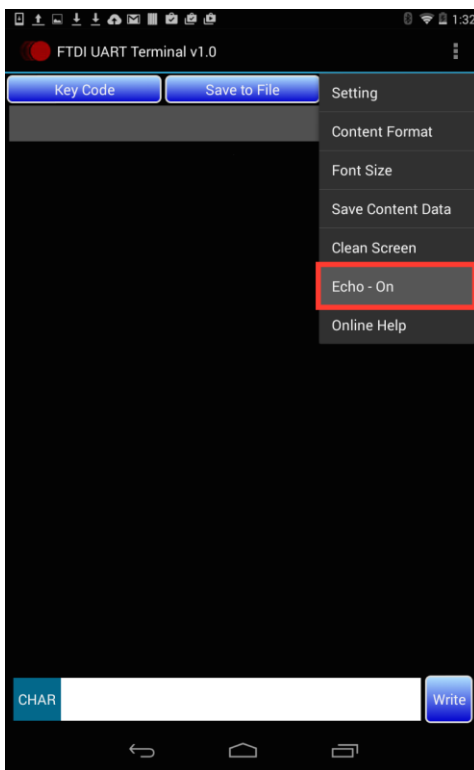
1. Ensure that the Android is running 3.2 or later.
2. Download and install the FTDI UART Terminal app from the app store (below left).
3. It should appear on the home screen and look like the image below right.



4. Connect 12V power to the device (MCU, EVCC or BMSC).
5. Open the app, or plug in the device. It should auto-detect and give a prompt to open it with a program from the list. Select FTDI UART Terminal from the list. (or check the box to always open with selected program). Once open, push the menu button (this differs from android to android).

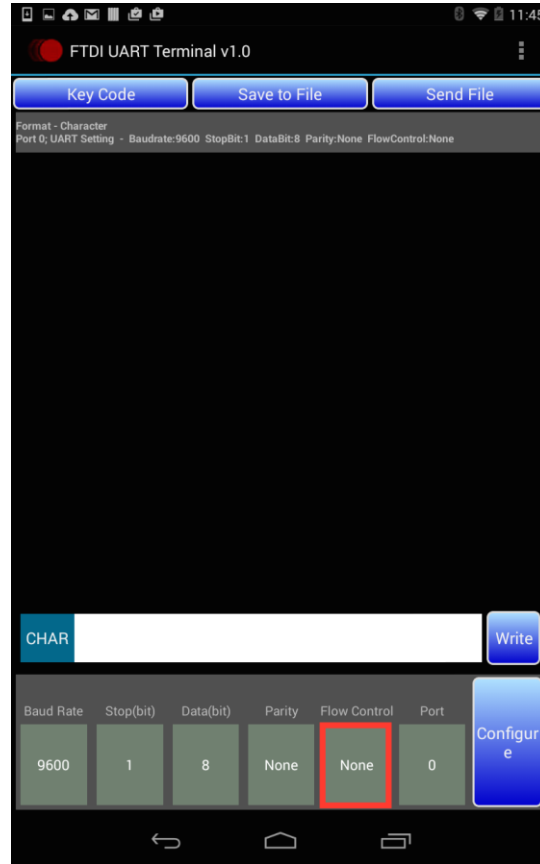
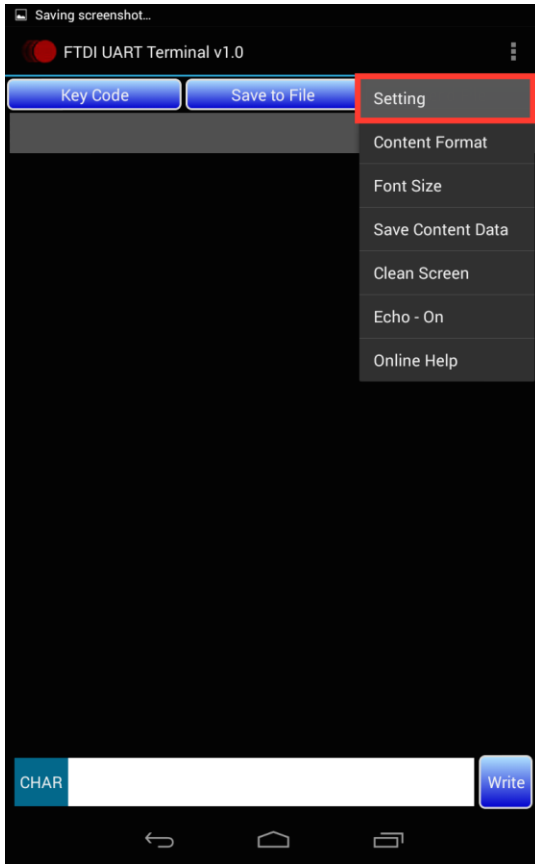


6. Turn Echo Off



7. Go to the menu again and this time go to settings. Change “Flow Control” to “None” then press configure.

Note: Settings cannot be configured unless the device is connected beforehand.



8. The Android Device should now be set up to program the device.

Note: because of the way the text data is sent to the device, it is necessary to place a “return” after each command before writing the command. This does what “enter” would do on the PC.

Firmware Upgrade – 8-bit devices (EVCC, BMSC, VCU, GFM)

The firmware upgrade procedure for Dilithium Design 8-bit devices uses the ds30 bootloader. This bootloader is distributed under the GNU GPL license. It is free and the application runs on Windows, MAC and Linux servers. See the next section *Firmware Upgrade – MCU* for MCU upgrades.

Documentation, screenshots and downloads are available at: <https://picbootloader.com/forum/>

Or, you may download the ds30loader application directly from ThunderStruck using this link:

<https://drive.google.com/file/d/1eBcBX2mSRB7UAzuq99fihCwV1Nm0roWl/view?usp=sharing>

The following example instructions are written for the EVCC, however, the process is similar for all 8-bit Dilithium products (not the MCU or Display). Before performing an upgrade, it is recommended that the existing device configuration be saved. Type “show config” and copy the results into a text file for reference after the upgrade.

1: Power on the device and connect it to the computer using the provided USB to Serial cable. In the EVCC terminal application, place the device into “upgrade mode” by entering the **upgrade** command as follows. The terminal response will give instructions.

```
evcc> upgrade
```

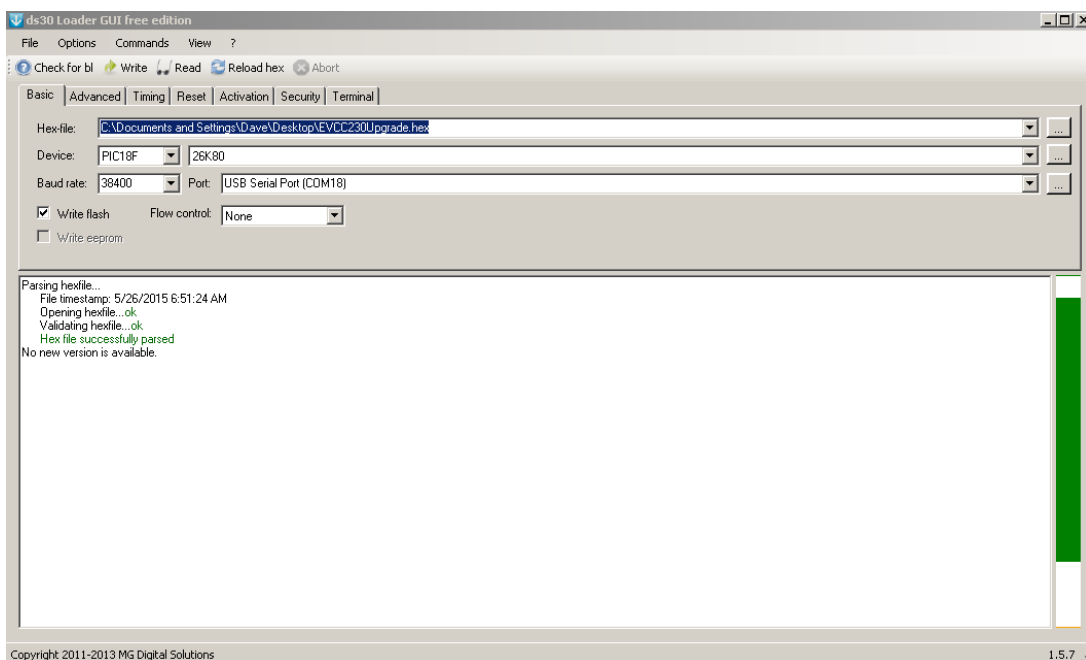
```

***                               Starting EVCC Upgrade                               ***
*** 1) Exit from the terminal application                                         ***
*** 2) Start the bootloader and download a new .hex file                         ***
*** 3) Restart the EVCC                                                         ***

```

2: At this point the EVCC is waiting for the upgrade. The EVCC LED will blink at a slow blink (once every 2 seconds). Close and exit from the terminal application (e.g., PuTTY), in order free up the serial port.

3: Start the ds30Loader application. The following screen should appear:

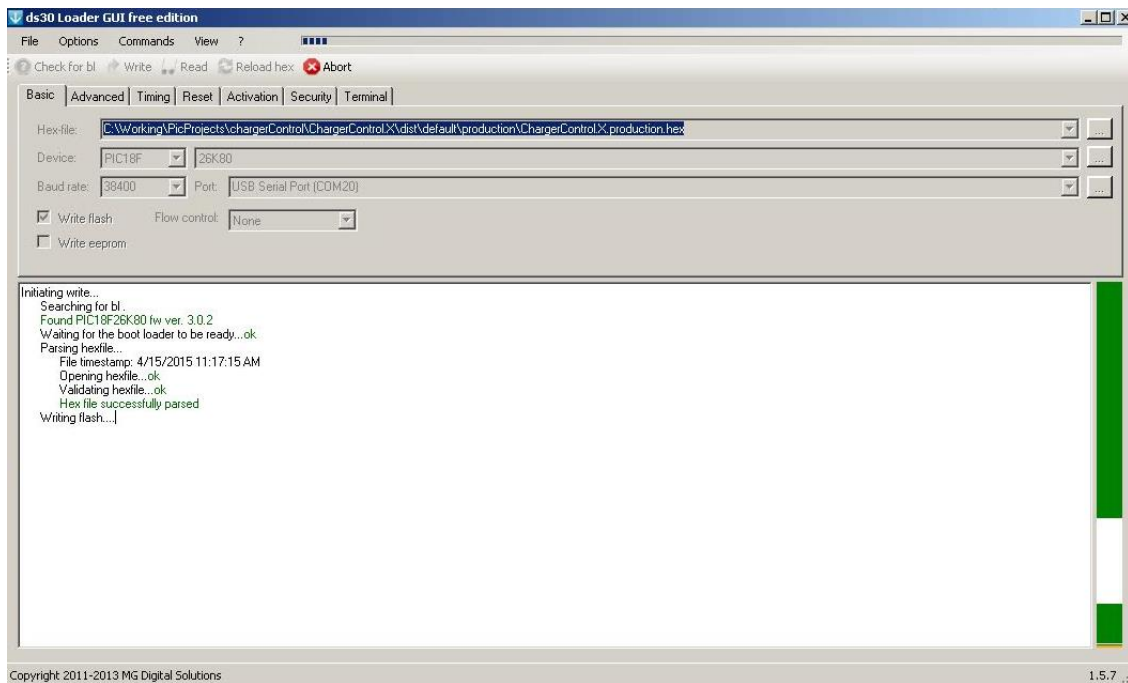


4: Navigate to the **Basic** tab.

5: Set upgrade parameters as follows:

- Set **Device** to PIC18F, and the unnamed parameter to the right of **Device** to 26K80.
- Set **Baud rate** to 38400.
- Set **Port** to the USB Serial Port COM number. The ds30Loader will present these as a pulldown if you type the “v” at the end of the field.
- Set **Hex file** to the hex file provided by Thunderstruck for the upgrade.
- Note that once the parameters have been set, they will be automatically available the next time the bootloader is used. And so, typically, only the Hex file would need to be changed.
- Check the **Write flash** box.

6: Click on the “Write” option (at the top left). The upgrade should start. The ds30 Loader will show a progress bar, and the EVCC LED will blink rapidly to show that the upgrade has started.



7: When the upgrade completes, exit the ds30Loader application, start up a new terminal application session, and cycle power to the EVCC or other device.

Enter the **show version** command in the device interface to verify that the proper version is running.

Firmware Upgrade – MCU

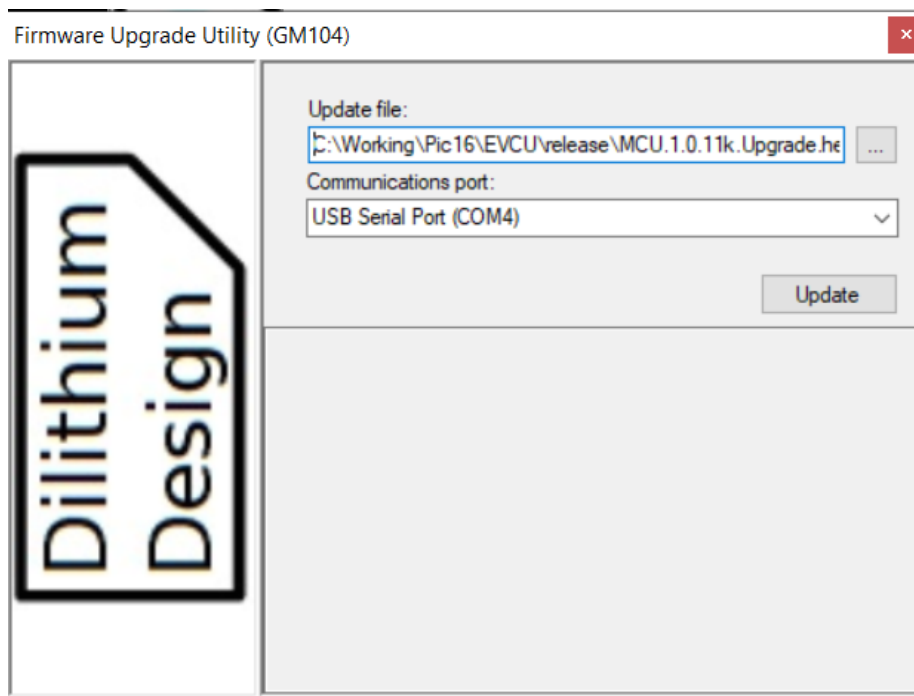
The firmware upgrade procedure for the MCU uses a Firmware Upgrade Utility. This application is only available for Windows machines.

The application is called MCU_FW_Upgrader.exe, and can be downloaded from this Google Drive location:

<https://drive.google.com/file/d/1fKjhtVndfnPnKpnL76Qy1oUvPszoTzsX/view?usp=sharing>

Before performing an upgrade, it is recommended that the existing device configuration be saved. Enter **show config** and copy the results into a text file for reference after the upgrade.

- 1: Move the firmware update file sent by Thunderstruck to your computer for access during this process.
- 2: Connect the MCU to the computer using the provided USB to Serial cable.
- 3: Start the MCU_FW_Upgrader application. The following screen should appear:



4. Set the correct **COM port** (use the pulldown) and navigate to the **Update file** you placed on the computer.
5. Select **Update**. The upgrade should start. The MCU_Firmware_Updater will show a progress bar, and the MCU LED will blink to show that the upgrade has started.
- 6: When the upgrade completes, exit the MCU_Firmware_Updater, start a new terminal application session, and cycle all power to the MCU. For a complete power cycle, the MCU must have all 12v sources disconnected.
7. Enter **show version** to verify that the proper version is running.
8. Enter **show config** to verify the configuration settings still match the values saved before the upgrade.

Warranty and Support

Dilithium Design reserve the right to make changes or improvements in design or manufacturing without assuming any obligation to change or improve products previously manufactured and / or sold.

For general support and warranty issues, contact

connect@thunderstruck-ev.com

For errors in this document, or comments about the product, contact

djmdilithium@gmail.com

Document History

Rev 1.0.0	Apr 17, 2014	Initial Version
Rev 1.1	Oct 25, 2017	Updated, made more generic
Rev 1.2	Aug 2018	Renamed, minor updates
Rev 1.3	Apr 2023	Updates, added MCU support