Pit Station Maintenance Manual:

The purpose of the Pit Station is to control and monitor the battery pack charging process. After all the QA tests, the Pit Station does satisfy most of the requirements, but only under very specific circumstances. It is essential for any users of the Pit Station to carefully read through the user manual before charging. If any exceptions/unexpected events occur, please refer to this manual to find any possible solutions.

Foreseeable Problems:

1. Having trouble deploy the software(cannot run the Charger Program on a new desktop/laptop):
2. Having trouble view/edit/compile/run the code for the Pit Station Software?
   1. I highly recommend you to download Java JDK with Netbeans from this site: <http://www.oracle.com/technetwork/java/javase/downloads/jdk-7-netbeans-download-432126.html>
   2. Afterwards, I suggest you download Git Bash from [here](http://git-scm.com/downloads), and go to a new folder your created. Now enter “git clone <https://github.com/MrAlwaysRight/SeniorDesignProject.git>” to clone the project to your local directory.
   3. Now open Netbeans, go to “open project” in the tool bar and open this project from the directory you clone the code into.
   4. Now you should have the source code, and you can either port the code into a different repository or request the write access so you can also push the code into Github. I will gladly grant you write access if I receive an email about the request.
3. “Cannot communicate with Arduino” warning keep showing up after starting up the charging program.
   1. There is an easy fix. Shut off the program by clicking “Quit” under “File” menu. Wait for 12 seconds.
   2. Now unplug the USB cable between the laptop and the Pit Station Box.
   3. Wait for 5 seconds.
   4. Reconnect the USB cable between laptop and the Pit Station Box
   5. Re-launch the program.
   6. Hover over “Mode” menu and “Charging Mode” submenu, check if “With Arduino” is selected
   7. If it still not connected, check the USB cable inside box (including the USB hub) and see if any of them are compromised. As long as the LED on the Arduino board is still on, that means the Arduino board is alive. The LED should also blink on start up.
4. “Cannot communicate with BMS” or similar warnings keep showing up.
   1. It is possible that the BMS boards are corrupted. Please shut off the program by clicking “Quit” under “File” menu. Wait for 12 seconds.
   2. Now unplug the I2C wire and plug it into SCADA box. Turn on SCADA, if the BMS green LED light is still on, that is good news. You can now connect the I2C back to the Pit Station Box and start the Charging program again. You should now expect now more of the warnings to appear.
   3. If “Cannot communicate with BMS” or similar warnings still keep showing up, then shut off the program by clicking “Quit” under “File” menu. Wait for 12 seconds.
   4. Now unplug the USB cable between the laptop and the Pit Station Box.
   5. Wait for 5 seconds.
   6. Reconnect the USB cable between laptop and the Pit Station Box
   7. Re-launch the program.
   8. If it still not connected, check the USB cable inside box (including the USB hub) and see if any of them are compromised.
   9. That’s as far as I can help you. If those warnings still show up, it is probably a bad day for you to buy lottery tickets.
5. Any reading on the Charging Monitor is not expected.
   1. Open the EventLog.txt file on the desktop
   2. Press “Ctrl+F” to search for keyword “Error”. Go through the log items with “Error” printed at the front and check for the time stamp printed at the end of the corresponding line to find the error that might help explaining the invalid reading on the GUI.
6. Cannot close the charging relay to start charging process
   1. First of all, you can monitor the charging relay’s status by outputting constant current and observe the actual current readings on the power supply. It should be a non-zero value once the relay is closed, assuming the power supply is outputting.
   2. When the Charging Program starts up, check the “Mode” menu and the “Charging Mode” submenu, if the selected item is “Without Arduino”, then it is a connection problem with the Arduino board. Check solution for problem #1.
   3. It is also that the Relay malfunctions and needs to be replaced. What you can do is to disconnect the relay from the box. Powering the input side with a power supply and check the conductance on the output side (The multimeter should beep, showing there is a connection). With the current Relay closed, the output resistance should be zero. If it still have close to infinite resistance, then that means it is a broken relay and it needs to be replaced
   4. If the Relay seems to be fine. What might have happened is that in the charging circuit, some connection is loose. You will need to use a multimeter and the flukes to probe around the charging circuit inside the box (from the Anderson connector to the charging supply, to the relay). Rewire and reinforce the wiring if any loose connection is found.
7. What if the BMS boards are corrupted(how to reprogram the BMS board):
   1. **If you plug the I2C cable into SCADA box and the LED for BMS on SCADA is still red, it probably means that the BMS boards are corrupted.**
   2. Please follow the following instructions to reprogram BMS boards:
      1. Following the BMS subsystem instructions to download firmware to the board
      2. Connect the BMS board to the I2C adaptor
      3. Connect the I2C adaptor to PC
      4. If the device driver cannot be automatically installed, please install it at <http://www.ftdichip.com/Drivers/VCP.html>
      5. Now launch RealTerm, which should be available in most lab machines. If not, install Realterm at <http://www.i2cchip.com/realterm>
      6. Now follow instructions here <http://sites.lafayette.edu/ece492-sp13/files/2013/03/Instructions-for-I2C-Operation1.pdf> to configure the serial port
      7. Select the bus number based on the connection between your BMS and the I2C adaptor. It is most likely to be 3, so click on 3 on the “bus number” on the I2C section. Now write to the board with address 0x04 with command 0x0100+address (7 bits/ 2 hex). For example 0x010002 for the first board
8. The Arduino board is corrupted/smashed into pieces/starting to rust:
   1. You can try to reprogram using the software IDE you can download [here](http://arduino.cc/en/Main/Software)
   2. You can then download the firmware code [here](https://github.com/MrAlwaysRight/SeniorDesignProject/tree/master/Arduino%20Software).
   3. You can then open the firmware and click on “Compile and Upload” button to program a new board
   4. To test if the code is actually new code you just download, open the Serial Monitor under “Tools” in the toolbar in the IDE. Now connect a voltmeter across pin 13 and once of the GND pins.
   5. Type in “a”, the voltage should jump to a digital high
   6. Type in “b”, the voltage should drop to 0
   7. Type in “d”, the serial monitor should spits out a “Ardui” message
9. The software is displaying the wrong number of cells, but no warnings are showing.
   1. First wait for several minutes. It is possible that the software just failed to pin some boards for certain polling cycle (this is common to happen when bypass is occurring). If this issue still occurs after several minutes, keep reading this.
   2. It is possible for you to have some loose connection in one of the bms boards. Check if the I2C connection is strong on all bms boards
   3. The only plausible explanation is that the second bms board(0x04) or even the third bms board is corrupted. Please follow Problem 5’s solution to reprogram and reassign the address of the boards.