

Pit Station Maintenance Manual

Siyuan Wang

05/11/13

Contents

| | |
|--|----------|
| Introduction | 2 |
| 1. Having trouble deploy the software(cannot run the Charger Program on a new desktop/laptop): | 2 |
| 2. Having trouble view/edit/compile/run the code for the Pit Station Software? | 3 |
| 3. “Cannot communicate with Arduino” warning keep showing up after starting up the charging program. | 3 |
| 4. “Cannot communicate with BMS” or similar warnings keep showing up. | 4 |
| 5. Why do I keep breaking the BMS?..... | 4 |
| 6. Any reading on the Charging Monitor is not expected..... | 4 |
| 7. Cannot close the charging relay to start charging process | 5 |
| 8. What if the BMS boards are corrupted(how to reprogram the BMS board):..... | 5 |
| 9. The Arduino board is corrupted/smashed into pieces/starting to rust:..... | 5 |
| 10. Everything works fine. It is just that when I click on the “Start” button when I want to start charging, the GUI does nothing. | 6 |
| 11. The software is displaying the wrong number of cells, but no warnings are showing. | 6 |
| 12. What if the fan doesn’t spin once the USB is connected..... | 6 |
| 13. What is left to do:..... | 6 |

Introduction

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):**
 - a. Go to [here](#) to download the dist folder in a zip file called dist.rar.
 - b. Unzip it and put it on your desktop.
 - c. Go to [here](#) to install the correct version of Java jdk based on the OS you are using
 - d. For a 64-bit OS, go to [here](#) to download the zip folder and extract rtxSerial.dll and rtxParallel.dll to C:\Program Files\Java\Jre6(or 7)\bin
 - e. For a 32-bit OS, go to [here](#) to download the zip folder and extract rtxSerial.dll and rtxParallel.dll to C:\Program Files\Java\Jre6(or 7)\bin
 - f. Now is the most challenging part: Hover over my Computer and click on “properties”. Find “Advanced System Settings”(this is different for different windows OS). Then click on “Environment Variables”. On the lower scrollarea, scroll down to “Path” and click on it, and then click on “Edit”. Now append your jdk and jre library to the end of the line starting with a semicolon to separate the previous one(i.e. ;C:\Program Files\Java\jre6\bin; C:\Program Files\Java\jdk1.7.0_15\bin; C:\Program Files\Java\jre7\bin;)
 - g. Now click on “OK”, “OK”...
 - h. Before you run the charging program by double clicking on the jar file in the dist folder, try downloading it and run it in the terminal in case more error messages occur.
 - i. Navigate to the dist folder using Command window (I assume you know how to do this with you bash skills from CS classes). Use “java -jar ChargingController.jar” to run the jar executable.
 - j. If it worked(GUI opened), it is very likely that you can run the jar file by double clicking it. If it still doesn't work, copy down the exception printed in the terminal and search that exception in [StackOverflow](#).
 - k. If you haven't install the Arduino driver, you can find the file you need to download from [here](#) and you can follow the instruction to install the driver from [here](#)(It is the instruction for a different type of Arduino board, but you should be following the same

procedure). The Arduino software IDE zipfile is also in the deployment folder in the repository. When driver is installed, right click on My Computer and click on “Manage” or “Device Manager” to check if the adaptor does show up as a new COMM Port.

- I. It is possible that you have not yet installed the driver for the I2C to USB adaptor. Click <http://www.ftdichip.com/Drivers/VCP.htm>. Make sure you disconnect the USB and connect it back again after the driver is installed so that it actually shows up as a new Comm port. Right click on My Computer and click on “Manage” or “Device Manager” to check if the adaptor and Arduino show up as two new COMM Ports.

2. Having trouble view/edit/compile/run the code for the Pit Station Software?

- a. 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>
- b. Afterwards, I suggest you download Git Bash from [here](#), 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.
- c. Now open Netbeans, go to “open project” in the tool bar and open this project from the directory you clone the code into.
- d. 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.

- a. If you haven’t install the Arduino driver, that is probably why. You can find the file you need to download from [here](#) and you can follow the instruction to install the driver from [here](#)(It is the instruction for a different type of Arduino board, but you should be following the same procedure). The Arduino software IDE zipfile is also in the deployment folder in the repository. When driver is installed, right click on My Computer and click on “Manage” or “Device Manager” to check if the adaptor does show up as a new COMM Port.
- b. There is an easy fix. Shut off the program by clicking “Quit” under “File” menu. Wait for 12 seconds.
- c. Now unplug the USB cable between the laptop and the Pit Station Box.
- d. Wait for 5 seconds.
- e. Reconnect the USB cable between laptop and the Pit Station Box
- f. Re-launch the program.
- g. Hover over “Mode” menu and “Charging Mode” submenu, check if “With Arduino” is selected

- h. 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.

- a. It is possible that you have not yet installed the driver for the I2C to USB adaptor. Click <http://www.ftdichip.com/Drivers/VCP.htm>. Make sure you disconnect the USB and connect it back again after the driver is installed so that it actually shows up as a new Comm port. Right click on My Computer and click on “Manage” or “Device Manager” to check if the adaptor does show up as a new COMM Port.
- b. It is possible that the BMS boards are corrupted. Please shut off the program by clicking “Quit” under “File” menu. Wait for 12 seconds.
- c. 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.
- d. 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.
- e. Now unplug the USB cable between the laptop and the Pit Station Box.
- f. Wait for 5 seconds.
- g. Reconnect the USB cable between laptop and the Pit Station Box
- h. Re-launch the program.
- i. If it still not connected, check the USB cable inside box (including the USB hub) and see if any of them are compromised.
- j. 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. Why do I keep breaking the BMS?

- a. Once you have connected the I2C cable and started the program, you may only unplug the I2C cable under one condition: after a safe shutdown
- b. A safe shutdown is
 - i. Go to “File” menu and click on “Quit”.
 - ii. Wait for 15 seconds

6. Any reading on the Charging Monitor is not expected.

- a. Open the EventLog.txt file on the desktop
- b. 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.

7. Cannot close the charging relay to start charging process

- a. 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.
- b. 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 [this solution](#).
- c. 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
- d. 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.

8. What if the BMS boards are corrupted(how to reprogram the BMS board):

- a. **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.**
- b. Please follow the following instructions to reprogram BMS boards:
 - i. Following the BMS subsystem instructions to download firmware to the board
 - ii. Connect the BMS board to the I2C adaptor
 - iii. Connect the I2C adaptor to PC
 - iv. If the device driver cannot be automatically installed, please install it at <http://www.ftdichip.com/Drivers/VCP.html>
 - v. Now launch RealTerm, which should be available in most lab machines. If not, install Realterm at <http://www.i2cchip.com/realterm>
 - vi. Now follow instructions here <http://sites.lafayette.edu/ece492-sp13/files/2013/03/Instructions-for-I2C-Operation1.pdf> to configure the serial port
 - vii. 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

9. The Arduino board is corrupted/smashed into pieces/starting to rust:

- a. You can try to reprogram using the software IDE you can download [here](#)
- b. You can then download the firmware code [here](#) or [here](#).

- c. You can then open the firmware and click on “Compile and Upload” button to program a new board
- d. 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.
- e. Type in “a”, the voltage should jump to a digital high
- f. Type in “b”, the voltage should drop to 0
- g. Type in “d”, the serial monitor should spits out a “Ardui” message

10. Everything works fine. It is just that when I click on the “Start” button when I want to start charging, the GUI does nothing.

- a. Safely shut down the program first by hovering over “File” and click on “Quit”.
- b. Now Wait for 15 seconds for the program to completely shut down
- c. Now disconnect the USB, wait for several seconds, and reconnect the USB back again.
- d. Try re-launching the program and see if that fixed it. If not, repeat from step a.

11. The software is displaying the wrong number of cells, but no warnings are showing.

- a. 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.
- b. 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
- c. The only plausible explanation is that the second BMS board (0x04) or even the third BMS board is corrupted. Please follow [this](#) solution to reprogram and reassign the address of the boards.

12. What if the fan doesn't spin once the USB is connected.

- a. Open the lid
- b. Replace the fan with a new fan(specifications can be found on BOM)
- c. Try connecting the USB cable and see if that helped to get the fan started.
- d. If that still doesn't work, start from step 'a'.

13. What is left to do:

- a. Control the power supply using the charging program
 - i. Using the spare USB port on the USB hub to connect to the RS232 input of the power supply via a USB-to-RS232 adaptor
 - ii. Learn how to interface RS232
 - iii. Add code to control the power supply
- b. Having a non-BMS mode.

- i. In addition to the “without” Arduino charging mode, the charging software also needs to have a “without BMS” mode where it doesn’t check the address of the BMS board, but still acquire cell data from the boards(although taking any data from the BMS boards will require knowing the board address). It is absurd, I know.
- c. Welcome screen, fancier screen about the team.
- d. Being able to save and load the most **recent** parameters.
- e. Solve the issue where one polling cycle has to be at least 12 seconds
 - i. Right now the issue that is keeping the length of the polling cycle from shortening is the switching between two serial port and the limitation of writing consecutive commands to the BMS board
 - ii. One issue to keep in mind that, you may never write to the BMS board “set command” consecutively without pausing. It is very likely that it will break whatever board you are sending this command
 - iii. Maybe pinging the Arduino board is not necessary until charging is in progress. That will shrink the polling cycle of non-charging time.
- f. Conversion equation for current is still off, but is close to be accurate enough.
- g. Having an XML parser to parse the XML data logged in the databaselog.xml file one level above the dist folder. Once the data is parsed, any graph regarding cell voltages, current, cell temperatures can be generated.