

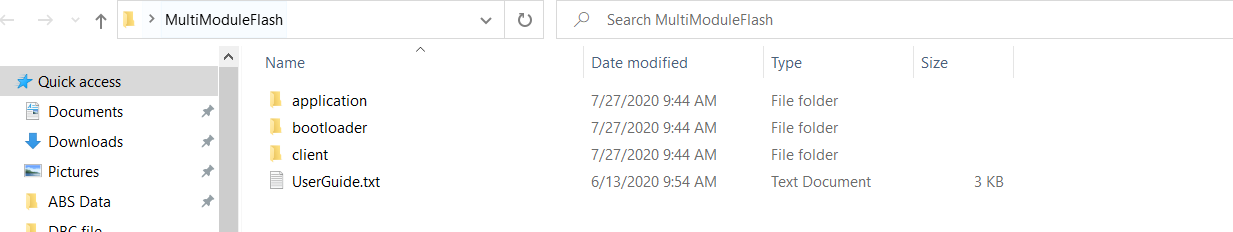
**48V pack core module SW flashing procedures**

**SW flashing hardware needed: PEAK CAN USB dongle.**

**Setup (first time only)**

1. Save the Flashing Environment:

Download and save the flashing environment on the desktop or location of your choice. This should have been provided from ABS via email/flash drive/drop box link



1. Install the Peak USB-CAN drivers:

<https://www.peak-system.com/quick/DrvSetup>

1. Install Bus Master:

(this is optional but allows you to see CAN traffic and read variables with the appropriate .dbf file)

<https://rbei-etas.github.io/busmaster/download.html>

1. Connect the Peak USB-CAN dongle



**Ethernet cable to battery**

**120 ohm Termination**

**RJ45 – DB9 adapter**

**Peak USB-CAN Dongle**

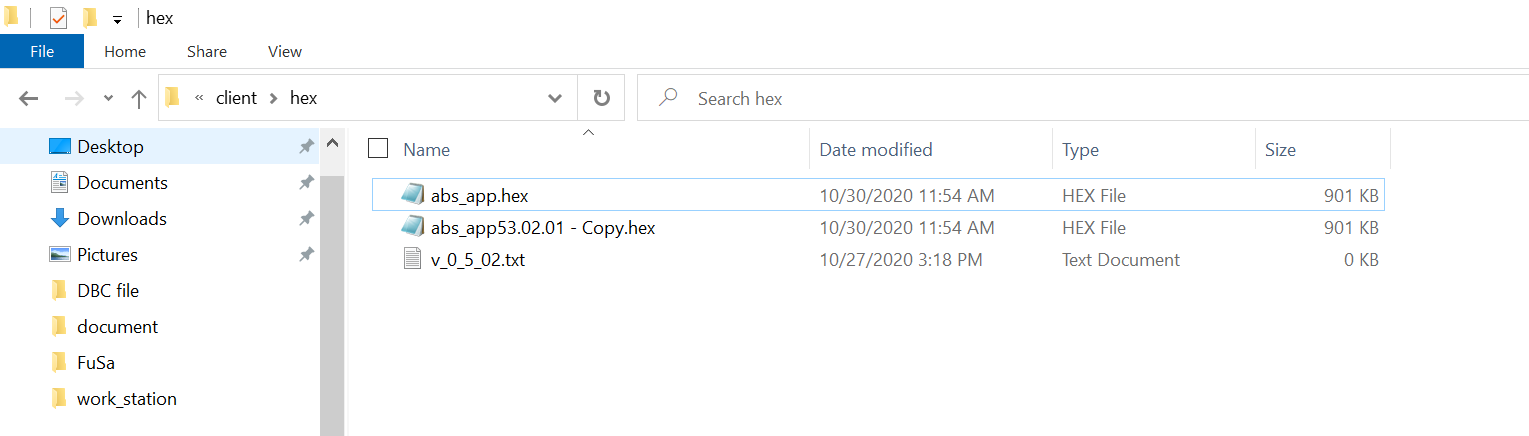
**APP SW Flashing steps:**

**Step 1: Make Connection**

Connect ONE Alliance Module external LV connector to the black or white ethernet splitter with an ethernet cable. Put the gold termination Ethernet connection into the empty space on splitter

**Step 2: Ensure correct Software is flashed**

In the flashing environment from the Setup section above, go to the folder client\hex and find the correct software to flash



The flashing tool will flash whatever software is called abs\_app.hex, if you have a new hex file release, go to this folder, delete the old one, and rename the new one to be abs\_app.hex

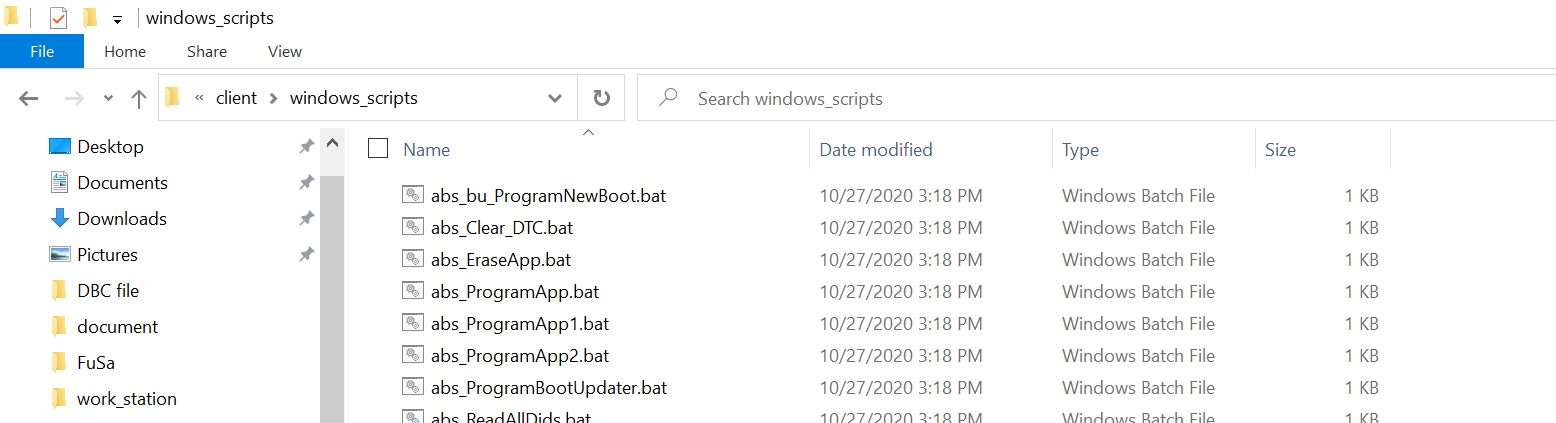
**Step 3: Ensure battery is NOT in Drive or Charge**

Make sure the FETs are open and the pack terminal is not energized

**Step 4: Flash the battery**

If you need to update the bootloader, then go to the “**Bootloader update steps**” procedures to update the bootloader first before flashing the application software, if bootloader update not needed, then go to the folder: \client\windows\_scripts and double click on the “abs\_ProgramApp.bat”. There could be windows warning, just accept them and proceed. Please refer below compatibility matrix table to see if the bootloader needs to be updated for the SW you are going to flash:

|  |  |  |
| --- | --- | --- |
| Bootloader version | App SW version < V4 | App SW version >= V4 |
| 0.0.4 | Yes | No |
| 0.1.2 | No | Yes |
| 0.1.3 | No | Yes |



**Step 5: Flashing**

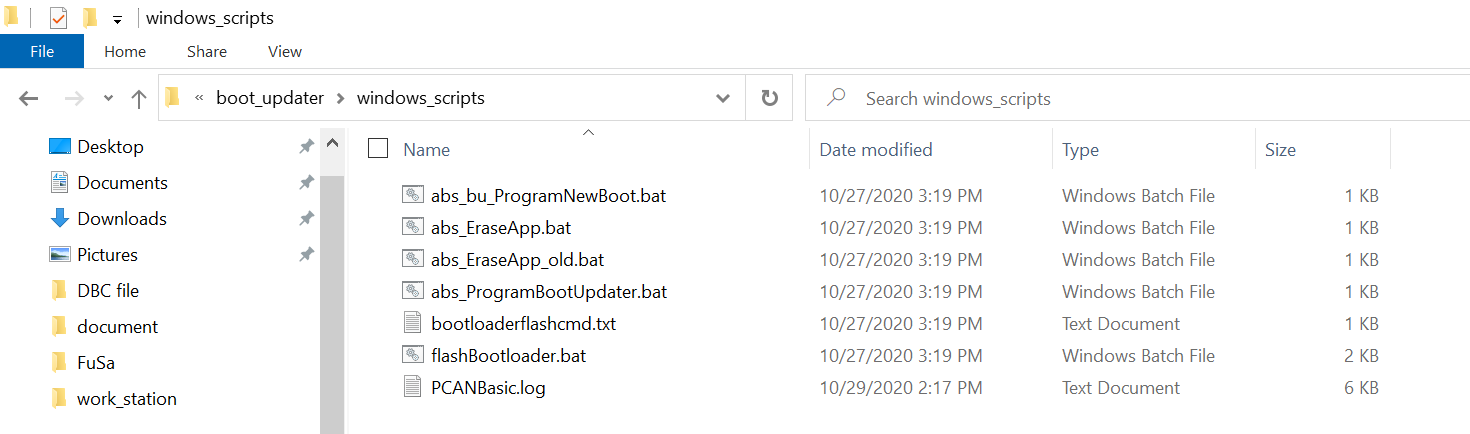
Then the flashing script will run and automatically erase the old program and start flashing the new program to the BMS board. If it fails initially, try it again right after it, it might be caused by the timeout fault while BMS was in sleep for the first time, normally it takes two times to successfully program it, or go back to step 3 and ensure the battery is not in Drive or charge

**Step 6: Arbitration (for multi\_module application)**

If you are flashing App software versions >= V5 and >= 2 modules connected in parallel, then please refer to the “**Node ID Arbitration steps**” to automatically flash the instance ID to each module for the multi-module configuration

**Bootloader update steps (PEAK CAN dongle needed):**

**Step 4.1:** go to the boot\_updater - > windows\_scripts



**Step 4.2**: double click “abs\_EraseApp.bat" to erase the old app software, and only let the bootloader running, you shall see a message showing the process done

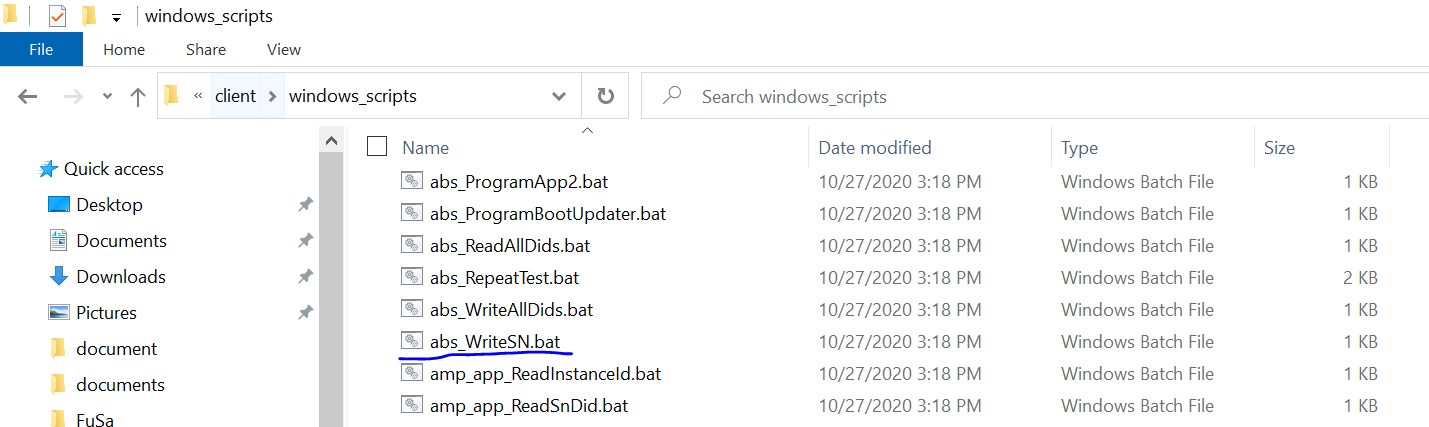
**Step 4.3:** double click “abs\_ProgramBootUpdater.bat", to load the updater for the new bootloader

**Step 4.4:** double clock “abs\_bu\_ProgramNewBoot.bat" to load the new bootloader

**Node ID Arbitration steps:**

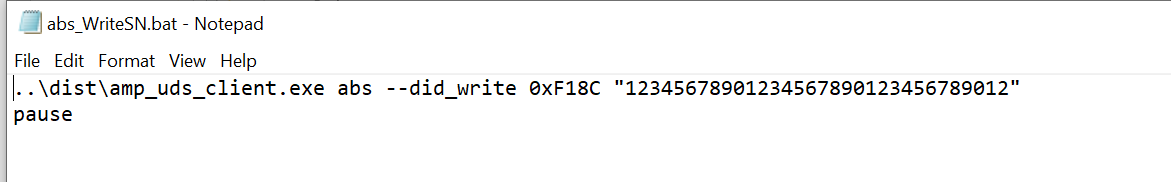
**Step 6.1: Serial number flashing**

This step needs PEAK CAN dongle, go to folder: \client\windows\_scripts, right click “abs\_WriteSN.bat” to edit the S/N and flash the serial number to the modules



Please use below format to edit the S/N in the batch file (19 digits for the S/N following 13 digits “0” s):





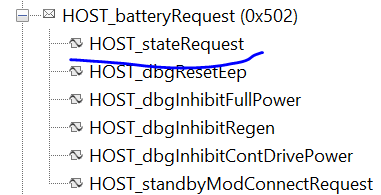
Replace "12345678901234567890123456789012" with "1000001222001210000000000000"

Then save, close the editor, double click” “abs\_WriteSN.bat” to flash the S/N to the modules, this has to be done one by one for the modules.

**Step 6.2: Arbitration (comparing S/N number and automatically write instance ID to the modules)**

**Step 6.2.1:**

Wakeup the modules by sending the signal **HOST\_stateRequest = 0** cyclically (Busmaster /CANalyzer, or other CAN tools) and make sure the module in standby mode



**Step 6.2.2:**

Send out the HOST\_nodeRequest message manually (**make sure it is not cyclic messages**)

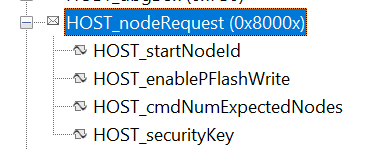
Configure the signals in HOST\_nodeRequest like below:

**HOST\_startNodeId = 1**

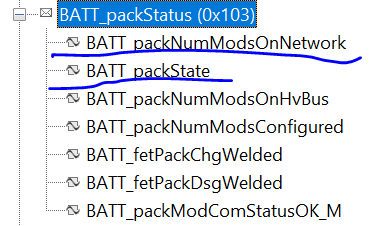
**HOST\_enablePFlashWrite = 1**

**HOST\_cmdNumExpectedNodes = 6**

**HOST\_securityKey = A3A3A3/10724259 (depends on using hex number / decimal number)**



Normally it will not work for just one time, a few times manual send retrial needed before successfully arbitrate the modules **(better wait around 2 minutes between each retrial),** if below signal **BATT\_packState** change from 0 to 1 (standby not ready to standby ready), that means the process had been completed. Please stop sending the HOST\_nodeRequest message after the process complete.



Also, the **BATT\_packNumModsOnNetwork** will be equal to the **HOST\_cmdNumExpectedNodes** if the process successfully completed

**Step 6.2.3 (optionally, if step 6.2.2 fails):**

Ideally, step 6.2.1 and step 6.2.2 shall complete the arbitration process, but since there is bug in V5 software (this bug will be fixed in V6 SW), normally it will not work directly with the **HOST\_cmdNumExpectedNodes > 2,** if step 6.2.2 fails, then please do it incrementally, for example, if 6 modules configuration is needed, then repeat step 6.2.1 and step 6.2.2 following incremental order 2 -> 3 -> 4 -> 5 -> 6, and modify the **HOST\_cmdNumExpectedNodes number accordingly:**

* Connect 2 modules, and follow step 6.2.1 and 6.2.2,

**HOST\_cmdNumExpectedNodes = 2**

* Add one module, totally 3 modules, and follow step 6.2.1 and 6.2.2, **HOST\_cmdNumExpectedNodes = 3**
* Add one module, totally 4 modules, and follow step 6.2.1 and 6.2.2, **HOST\_cmdNumExpectedNodes = 4**
* Add one module, totally 5 modules, and follow step 6.2.1 and 6.2.2, **HOST\_cmdNumExpectedNodes = 5**
* Add one module, totally 6 modules, and follow step 6.2.1 and 6.2.2, **HOST\_cmdNumExpectedNodes = 6**