USER GUIDE,
BT-70939Ax/Cx(H)
CAN BUS

Bren-Tronics , Inc.
P: 631-499-5155 | F: 631-499-5504
www.bren-tronics.com

This document contains technical data subject to change without notice.

NOTICE TO PERSONS RECEIVING THIS DRAWING AND/OR TECHNICAL INFORMATION

"Bren-Tronics, Inc." claims proprietary rights in the material disclosed herein. This drawing and/or technical information is issued in confidence for information only and may not be reproduced or used to manufacture anything shown or referred to herein without direct permission from "Bren-Tronics, Inc." to the user. This drawing and/or technical information is loaned for mutual assistance and is subject to recall by "Bren-Tronics, Inc." at any time. This drawing and/or technical information is the property of "Bren-Tronics, Inc."

Revision History

Revision	Date	Changes
A	03-JUL-2019	Not released
B	09-JUL-2019	First release
C	05-SEP-2019	Clarify FET control / diag mode
D	20-OCT-2020	Add Maintenance ‘Key’

Table of Contents

Revision History 3

List of Tables 5

Electrical Connection 6

Connector 6

CAN Bus Enable and Bus Termination 7

Protocol Overview 8

Introduction 8

Bit Rate 8

Battery Network Address 8

Communication PDU 8

Communication PGN 8

Frame Contents 8

CAN Frame Descriptions 9

CAN Frame Examples 10

Battery Firmware Version Request 10

Battery Status Request 10

Set the AtRate Current Value 10

Enter the Diag State 11

Exit the Diag State 11

Activate Heaters 11

Set the FET Control State 12

Battery Status Command Codes 13

Battery Status Flags 14

Firmware Update Procedure 15

Fuel Gauge Update Procedure 17

Full Update Procedure 18

Appendix 1 – CANbus Interface Cable Wiring Diagram 19

List of Tables

Table 1 - BT-70939 CAN Bus Connector Pin Designations 6
Table 3 - CAN Request Frame 8
Table 4 - GetBatteryVersion 10
Table 5 - GetBatteryStatus 10
Table 6 - SetAtRate 11
Table 7 - SetFETState 12
Table 8 - Battery Status Command Codes 13
Table 9 - Battery Status Flags 14

Electrical Connection

Connector

The BT-70939 family of 6T Lithium Ion batteries has a +5V CAN bus interface available on two circular connectors. The two connectors are wired in parallel to enable a daisy chain configuration. Mating connector P/N is Bren-Tronics part number 350486.

Table 1 - BT-70939 CAN Bus Connector Pin Designations

Pin Designations	Function	Description
A	+5V Input	Apply +5V at 20mA to this pin to activate the CAN bus.
B	Reserved	Do not connect
C	+28V Input	Apply +6V to +36V at 20mA to this pin to activate the CAN bus.
D	CANH	CAN communication signal, high line of pair
E	CANL	CAN communication signal, low line of pair
F	RTN	Ground return for all of the above signals
G	Reserved	Do not connect
H	Reserved	Do not connect
J	Reserved	Do not connect
K	Reserved	Do not connect

1. The two CAN connectors are wired in parallel, pin A to pin A, pin B to pin B, etc.
2. The CAN interface MUST be activated, by applying voltage to either Pin A, or Pin C, within the range shown for the pin being used.
3. If the battery is located at the end of the equipment CAN bus, connect a 120-ohm resistor between pins D and E of one connector, and connect the CAN bus to the other connector.

CAN Bus Enable and Bus Termination

The BT-70939 battery is always active and ready for service, however in order to communicate with the battery using the CAN bus, the CAN bus must be enabled by applying an external power signal to the CAN bus interface. Power can be applied by either connecting a +5V (20ma) source on pin A, or connecting a 6-30V (20mA) source on pin C. Table 1 describes the pins on the CAN bus connector.

Note that an external 120 ohm termination resistor must be added between Pin D (CANH) and Pin E (CANL) of the connector for proper CAN bus operation.

Protocol Overview

Introduction

The BT-70939 battery utilizes the differential CAN bus for communication. The battery telemetry protocol rides on the back of the CAN and J1939 protocol standards.

Bit Rate

The BT-70939 battery uses CAN differential voltage signaling at 250kbps.

Battery Network Address

Battery devices obtain their network address using the J1939 Address Claim Method. BT-70939 batteries claim addresses in the 192 – 239 address range. While the BT-70939 battery contains a subset of the J1939 protocol as it relates to dynamic addressing, it DOES NOT support commanded addressing.

Communication PDU

The BT-70939 battery replies to the host using PDU format number 254.

Example Response: 18FED0C0{data}

Communication PGN

The BT-70939 battery expects data queries from a host on PGN number 61184 (PropA, Data Page 0).
Example Request:
18EFC0D0000000000000000 {GetBatteryVersion}

Frame Contents

Request packets are sized to fit an 8-byte CAN frame described as follows:

Table 2 - CAN Request Frame

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
COMMAND	TYPE	ARG0	ARG1	ARG2	ARG3	ARG4	MSGID[1]

[1] MSGID – 8 bit message identifier. Return packets will return this tag as a method to pair responses to a given request.

CAN Frame Descriptions

- REQUEST FRAME DEFINITIONS:
  - COMMAND 0 = BMS Information
    TYPE 0 = BMS firmware version
  - COMMAND 1 = BMS Telemetry
    (reserved) TYPE 0 = SMBUS
    Read
    - ARG0 = SMBUS data address
    - ARG1 = SMBUS data return
      type o 0 = byte
      o 1 =
      word o 2
      = block

    TYPE 1 – TYPE 255 Reserved
  - COMMAND 2 = BMS Diagnostics
  - COMMAND 3 = BMS Control
    TYPE 0 = BMS FET Control
    - ARG0 = FET state
      - 0 = charge and discharge off
      - 1 = charge FET on, discharge FET off
      - 2 = charge FET off, discharge FET on
      - 3 = charge FET on, discharge FET

    on TYPE 1 = BMS Reset
    - ARG0 = Reset Type
      - 0 = reset the Telemetry controller
      - 1 – 255 = reserved

  - MSGID - The last byte in the frame (BYTE 7) is defined as a message identifier, and is used to pair a message response to a given request. In cases where a response may be ambiguous due to other traffic on the buss, or a response received asynchronously “out of order”, it becomes necessary to correlate the response to a given request. The MSGID provides that mechanism. Typically the requestor keeps a rolling count, incremented on each request. The 6T provides that same number in the MSGID field of it’s reply such as the two can be matched together.

CAN Frame Examples

Battery Firmware Version Request

Table 3 - GetBatteryVersion

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0	0	X	X	X	X	X	MSGID

Battery Version Return Packet

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Version Major	Version Minor	Patch number LSB	Patch number MSB	Build number LSB	Build number MSB	X	MSGID

Battery Status Request

Table 4 - GetBatteryStatus

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
1	0	Command code	1 or 2[2]	Len[3]	2	X	MSGID

Battery Status Return Packet (status words)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
response LSB	Response MSB	1	0	Command code	1	X	MSGID

Battery Status Return Packet (strings)

Byte 0	Byte 1	Byte 2 – Byte (Len – 1)	Byte Len	Byte (Len+1)	Byte (Len+2)	Byte (Len+3)	Byte (Len+4)
response byte 1	Response byte 2	Response bytes 3 – Len	1	0	Command code	2	MSGID

Set the AtRate Current Value

[2] 1 is used to read status words (voltage, current, temperature), 2 is used to return strings.
[3] Maximum length of a returned string. When reading words it is a do not care.

Table 5 - SetAtRate

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
1	1	4	1	X	X	AtRate LSB	AtRateMSB

Enter the Diag State

*The Diag(nostics) state opens a 5 second window allowing control of the BMS FET Control State, BMS reset, and alternate BMS heater control. The BMS will automatically exit the Diag state if the command is not issued prior to the 5 second expiration.

Table 7 – Enter Diag State

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
3	3	0x37	0x30	0x39	0x33	0x39	X

Exit the Diag State

*The Diag(nostics) state opens a 5 second window allowing control of the BMS FET Control State, BMS reset, and alternate BMS heater control. The BMS will automatically exit the Diag state if the command is not issued prior to the 5 second expiration.

Table 8 – Exit Diag State

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
3	4	X	X	X	X	X	X

Activate Heaters

*The Diag(nostics) state must be enabled for this command to operate. The BMS will activate the onboard heaters for 10 seconds. To maintain heater operation, retransmit this command prior to the 10 second timeout.

Table 9 – Activate Heaters

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
3	5	X	X	X	X	X	X

To activate the BMS heaters:

1. Enter the Diag state as defined by the EnterDiagState command
2. Within 5 seconds, issue the EnableHeaters command before the Diag state automatically times out.
3. Exit the Diag state or allow it to timeout.

Set the FET Control State

*The FET Control State commands are subject to the safety protection features of the battery and will perform accordingly.

Commands that attempt to turn either of the FETs off would always be successful as they place the battery in a “safe” state.

Commands the attempt to turn either FET on, may or may not be successful due to voltage, current, or temperature conditions.

Table 10 - SetFETState

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
3	0	FET State	X	X	X	X	X

FET State:

0. Both FETS off
1. Charge FET ON
2. Discharge FET ON
3. Charge and Discharge FETs ON Expected response: 3-0-FETState-0-X-X-X-X

To change the BMS FET state:

1. Enter the Diag state as defined by the EnterDiagState command
2. Within 5 seconds, issue the SetFETState command before the Diag state automatically times out.
3. Exit the Diag state or allow it to timeout.

Notes:

1. Current specified in mA / 40, i.e. 100A = (100,000mA/40) = 2500mA.
2. Positive current values for charge, negative current values for discharge (100A discharge = -2500mA = 0xF63C)
3. To query an AtRate run time, issue the SetAtRate command followed by the appropriate request. A GetBatteryStatus request can be issued with command code 4 to read back the selected AtRate current setting.
4. After setting the AtRate current value, an AtRateOK request returns a Boolean indicating whether the battery can deliver that energy for 10 seconds. Allow one second between setting the current and querying AtRateOK for processing.

Battery Status Command Codes

Codes and units conform to the smart battery specification, version 1.1

Table 11 - Battery Status Command Codes

Function	Code (hex decimal)	Data	Sign
AtRate	0x04	mA / 40	unsigned
AtRateTimeToFull	0x05	minutes	unsigned
AtRateTimeToEmpty	0x06	minutes	unsigned
AtRateOK	0x07	Boolean (0=no, else yes)	unsigned
Temperature	0x08	0.1°K	signed
Voltage	0x09	mV	unsigned
Current	0x0a	mA / 40	signed
Avg Current	0x0b	mA / 40	signed
MaxError	0x0c	Percent	unsigned
RelStateofCharge	0x0d	Percent	unsigned
AbsoluteStateofCharge	0x0e	Percent	unsigned
RemainingCapacity	0x0f	mAh / 40	unsigned
FullChargeCapacity	0x10	mAh / 40	unsigned
RunTimeToEmpty	0x11	minutes	unsigned
AvgTimeToEmpty	0x12	minutes	unsigned
AvgTimeToFull	0x13	minutes	unsigned
ChargingCurrent	0x14	mA / 40	unsigned
ChargingVoltage	0x15	mV	unsigned
BatteryStatus[4]	0x16	bit flags	unsigned
CycleCount	0x17	Count	unsigned
DesignCapacity	0x18	mAh / 40	unsigned
DesignVoltage	0x19	mV	unsigned
ManufactureDate	0x1b	unsigned int	unsigned
SerialNumber	0x1c	number	unsigned
ManufacturerName	0x20	string	-
DeviceName	0x21	string	-

[4] See Table 9 – Battery Status Flags

Battery Status Flags

Table 12 - Battery Status Flags

Description	Bit Number	Comments
Overcharged Alarm	15	1 = Active, 0 = Inactive
Terminate Charge Alarm	14	1 = Active, 0 = Inactive
Over Temperature Alarm	12	1 = Active, 0 = Inactive
Terminate Discharge Alarm	11	1 = Active, 0 = Inactive
Remaining Capacity Alarm	9	1 = Active, 0 = Inactive
Remaining Time Alarm	8	1 = Active, 0 = Inactive
Initialization	7	1 = Active, 0 = Inactive
Charge FET Test	6	1 = Battery discharging or at rest, 0 = Battery is charging
Fully Charged	5	1 = Battery is fully charged, 0 = Battery in not fully charged
Fully Discharged	4	1 = Battery is fully discharged, 0 = Battery is ok
Error Codes	3:0	0 = ok 1 = busy 2 = reserved command 3 = unsupported command 4 = access denied 5 = overflow/underflow 6 = bad size 7 = unknown error

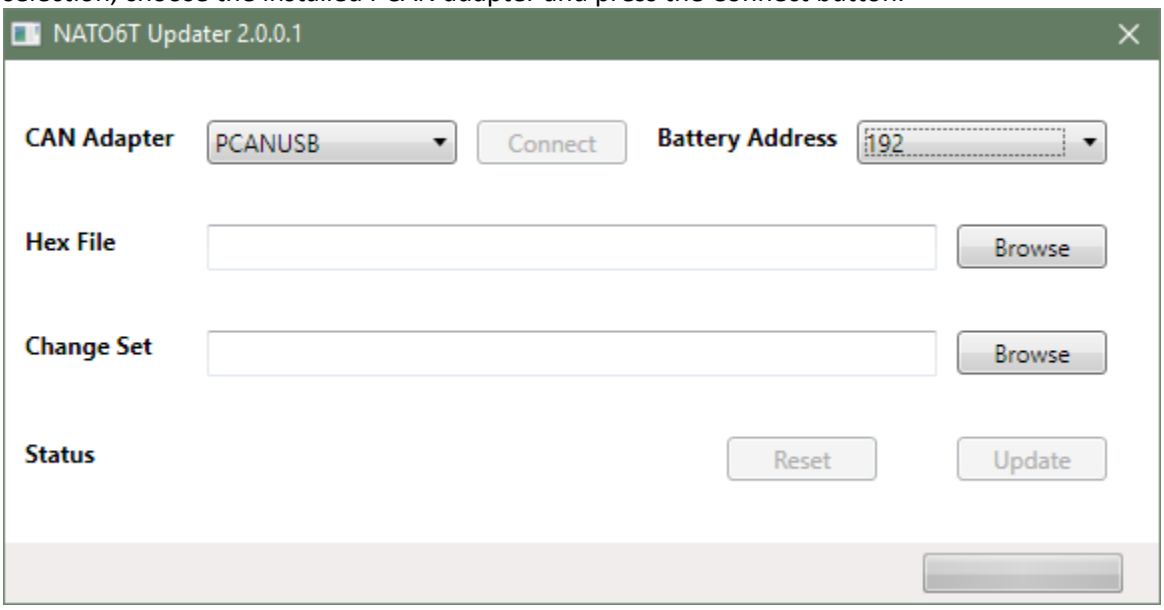
Firmware Update Procedure

Required Items:

1. The Firmware Update application and associated files are included in the associated zip file (*.zip).
2. A Personal Computer (PC) running Windows XP or newer with a PEAK-System Technik GmbH CAN interface adapter (PCAN-USB model IPEH-002022).
3. An interface cable - DB9 Male to mating circular 6T CAN connector (see Appendix 1).

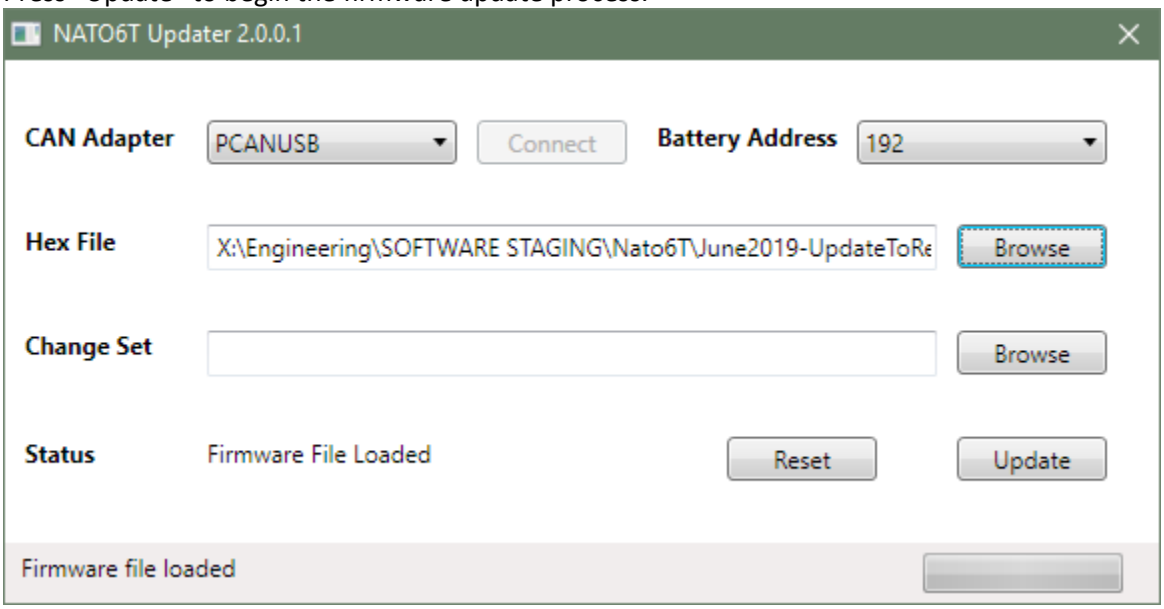
Procedure:

1. Extract the files contained in the associated zip file to a local directory on your hard drive.
2. Install the PCAN adapter, allowing it to install the drivers required for your computer.
3. Connect the PCAN adapter to the battery using the required cable and activation power supply.
4. Execute the Windows program “NATO6TUpdater.exe”. Using the CAN Adapter drop down selection, choose the installed PCAN adapter and press the Connect button.



5. From the pulldown menu, select the address of the battery to be updated.
6. With the CAN adapter and the battery selected, the Hex File Browse button will be available. Press Browse and navigate to the new hex file. Once selected, the “Browse” button will change to “Update”. If for any reason you need to select a different file, press “Reset”.

7. Press “Update” to begin the firmware update process.



8. The bootload process typically takes between 2 and 3 minutes.
9. Once finished, disconnect the cables, and continue to use that battery as necessary.

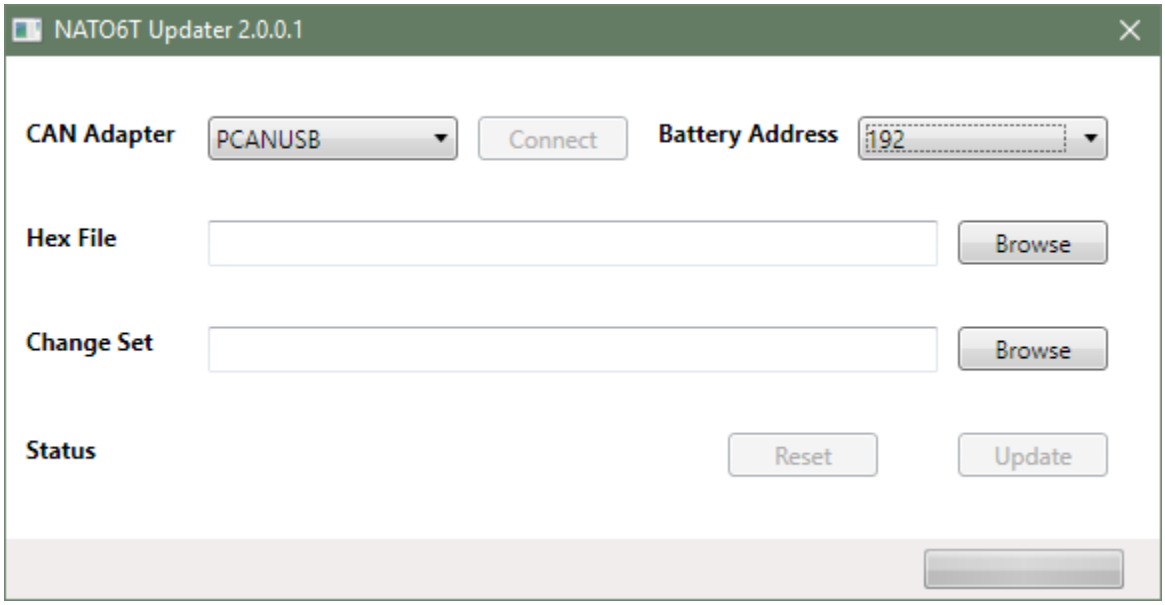
Fuel Gauge Update Procedure

Required Items:

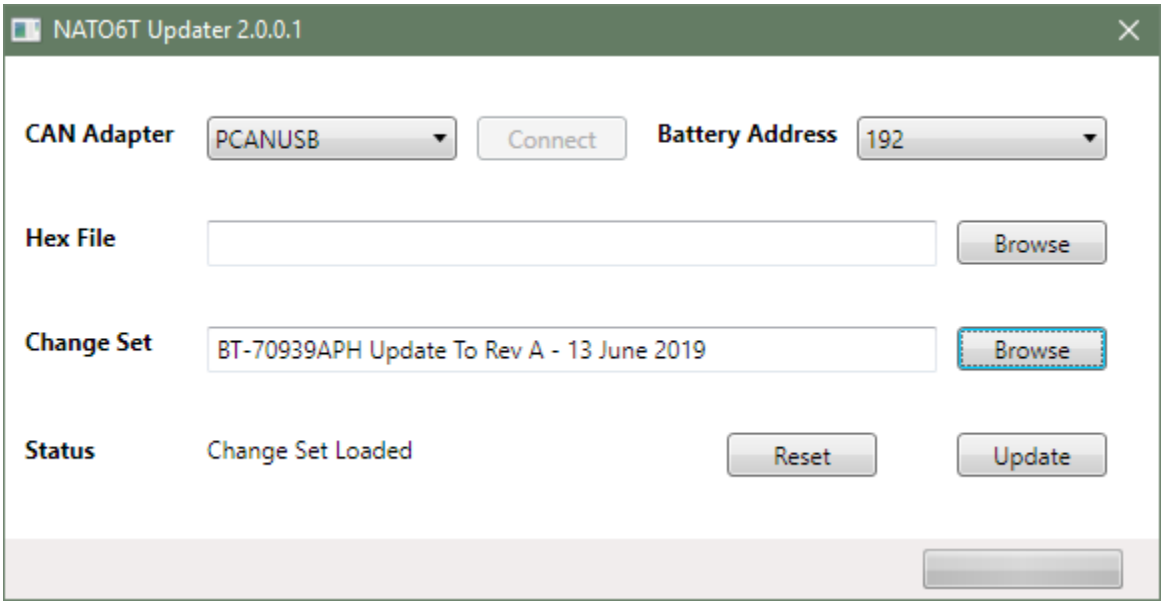
1. The Gauge Update application and associated files are included in the associated zip file (*.zip).
2. A Personal Computer (PC) running Windows XP or newer with a PEAK-System Technik GmbH CAN interface adapter (PCAN-USB model IPEH-002022).
3. An interface cable - DB9 Male to mating circular 6T CAN connector (see Appendix 1).

Procedure:

1. Extract the files contained in the zip file to a local directory on your hard drive.
2. Install the PCAN adapter, allowing it to install the drivers required for your computer.
3. Connect the PCAN adapter to the battery using the required cable and activation power supply.
4. Execute the Windows program “NATO6TUpdater.exe”. Using the CAN Adapter drop down selection, choose the installed PCAN adapter and press the Connect button.
5. From the pulldown menu, select the address of the battery to be updated.



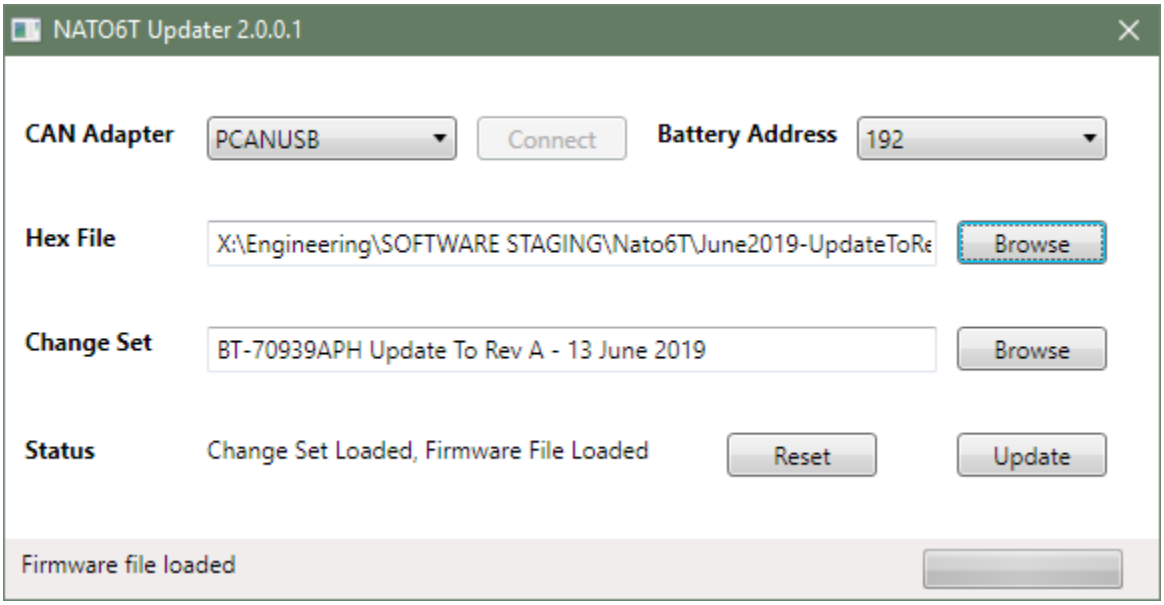
6. The Browse button should now be available. Browse to the appropriate changeset file - *.btcs. When selected, you should see something similar to the following:



7. Press update.

Full Update Procedure

1. In some cases, both a Firmware and Fuel Gauge update will be required. Select the CAN adapter, the battery address, Firmware (*.hex) and Fuel Gauge Update (*.btcs) and press Update.



Appendix 1 – CANbus Interface Cable Wiring Diagram

Build CANbus harness

FOR REFERENCE ONLY:

PCAN-USB DE9 connector

http://gridconnect.com/can-usb.html

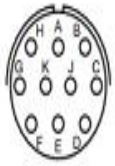
Pin assignment 9-pole connector male:

Pin	Configuration
1	+12 V / +5 V / Not connected
2	CAN-L
3	CAN-GND / Not connected
4	Not connected
5	Not connected
6	CAN-GND / Not connected
7	CAN-H
8	Not connected
9	+12 V / +5 V / Not connected

FOR REFERENCE ONLY:

Front face of PINS illustrated, page 8 of

http://www.amphenol-aerospace.com/pdf/12-070.pdf



Insert Arrangement	**12-10**
Service Rating	**I**
Number of Contacts	**10**
Contact Size	**20**

CANbus Harness using #393134 cable assy				
			Circular (M) to DE9 (F) Cable	
Color	**PCBA contact #**	**Schem net name**	**Circular Connector Pin BT#350486 AMPHENOL PT06E-12-10P(SR)**	**DE-9 Connector pin**
Black	1	+5V input	A	
Red	2	PACK_POS	B -> C	
White	3	CAN_24V	C -> B	
Green	4	CANH	D	7
Yellow	5	CANL	E	2
Blue	6	GND	F	3

NOTE: Circular connector pins B & C are jumpered together so that Cell Pack Positive supplies power to CAN bus IF circuitry on 6T main PCBA

10 NOV 2014 PM Engineering Reference Only 1