Zebra Battery Programmer Box User Manual

This document is for Zebra internal use ONLY!

Table of Contents

Overview....3
Battery Programmer Features....3
Batteries Currently Supported....3
Hardware....4
Software Installation....4
Firmware Updating....6
Command List....7
Detailed Command Descriptions....8
- B – Select battery type....8
- R(E|A) – Read battery data....8
- W(E|A) – Write battery data....8
- U(E|A) – Update battery data....9
- V(E|A) – Verify battery data....9
- E – Erase buffer....9
- LS – Load Hex file from serial port....9
- DS – Dump Hex file to the serial port....10
- LF – Load Hex file from SD card....10
- SF – Save Hex file to SD card....10
- VBD – Validate battery data....10
- MV2 – “Manufacture” PP+ V2 battery....10
- DIR – Directory of SD card....10
- DPP – Display PP data....10
- DPP+ – Display PP+ data....10
- A – Authenticate battery....10
- DGG – Display gas gauge registers....10
- DGGH – Display gas gauge registers in Hex....10
- DGGF – Dump gas gauge flash in Hex....11
- SGGF – Save gas gauge flash....11
- HGG – Hibernate gas gauge....11
- FSGG – Set FULLSLEEP Bit....11
- RGG – Reset gas gauge....11
- SGG – Seal gas gauge....11
- USGG – Unseal gas gauge....11
- FUSGG – Fully unseal gas gauge....11
- VERB – Toggle verbose mode....11
- VER – Show version information....11
- STAT – Show device status....12
- ? – Display help....12

Overview

This manual covers the use of the Zebra Battery Programmer Box which allows for the programming and reading/verifying of most Zebra batteries.

Battery Programmer Features

Simple serial interface over USB. Only requires serial terminal software such as TeraTerm.

Compatible with Windows, Mac OS, and Linux.

Compatible with various battery I2C voltage levels and drive circuits.

- Supports 3.3V, V_{Batt} and 5V I2C drive voltages
- Supports pull up resistor values from 2.3K to 10K
- Supports both active and passive clock drive. (Active clock drive is at 3.3V)

Support for all Zebra EEPROM sizes/addresses

Ability to check for a valid authentication IC.

Support for reading and writing hex files to/from EEPROM, Authentication IC, SD card and the serial interface.

Support for “updating” battery data. This function replaces all the data in the battery except for those items set during manufacture. (Serial number, part number, revision, date made, etc..) This allows a battery to have it's hex file data replaced without un-manufacturing the battery.

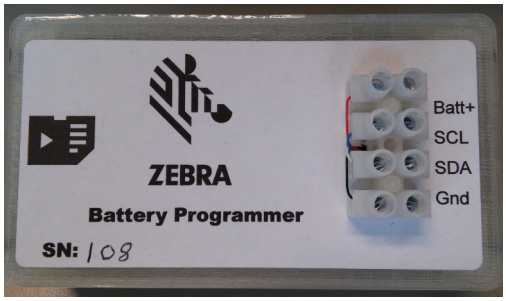
Can display battery data in human readable form. This includes battery EEPROM data as well as gas gauge data and registers.

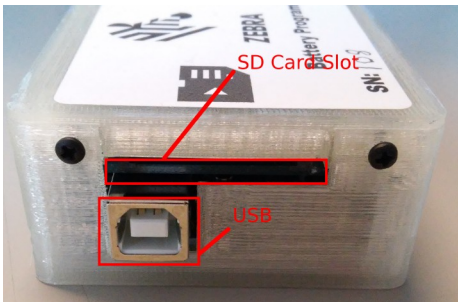
Batteries Currently Supported

- MPA2/MPA3 Smart battery
- MPA3 Gifted battery (MC18, Rogue, Frenzy)
- Pollux battery
- Falcon/IronMan/Thunder/Lightning Battery
- Hawkeye Battery
- Sentry Battery
- Galactus/Badger/Frozone battery
- Meteor/Gravity/Simba battery
- Value Tier Battery
- PP+ Version 2 Battery (New Gas Gauge)

Hardware

The battery programmer is based on an Arduino Mega 2560 board with a custom shield for controlling the I2C interface. As can be seen in the images below it has a terminal block for the battery connections, a USB connector for power and host PC hookup and an SD card slot.

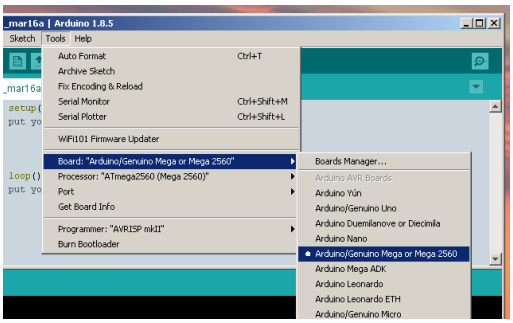




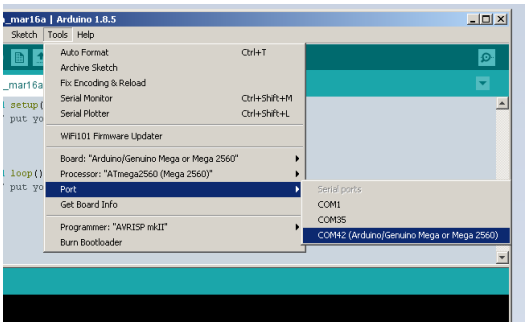
Software Installation

First you must install the Arduino IDE. This will load the needed USB serial port drivers and has a serial port console. The latest Arduino IDE can be found at: https://www.arduino.cc/en/Main/Software Download the version of the IDE appropriate to your system. (Make sure you select “Download the Arduino IDE” and not “Access the Online IDE”

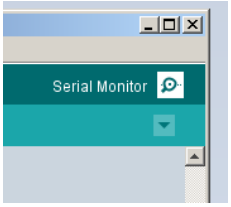
Once the IDE is installed use a USB cable to plug the battery programmer into a USB port on your machine. The device should be recognized and the driver installed. Once that's done start the Arduino IDE, and then select Tools->Board from the menu as shown below. Select the “Arduino/Genuino Mega or Mega2560”.



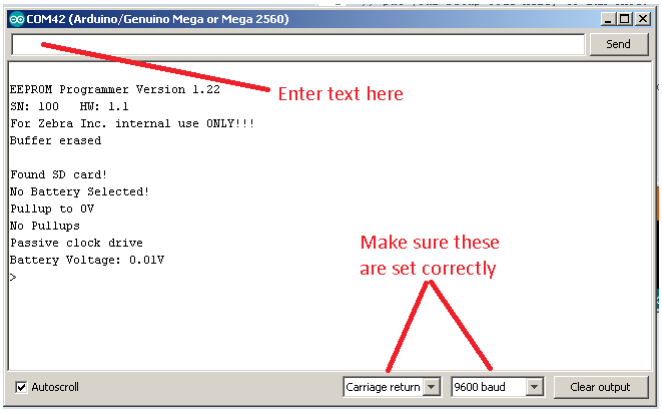
Then select Tools->Port from the menu as shown below. Select the port your box was assigned, it should be labeled “Arduino/Genuino Mega or Mega2650”



Then start the Serial Monitor. Click on the magnifying glass icon in the upper right corner of the window, as shown below.

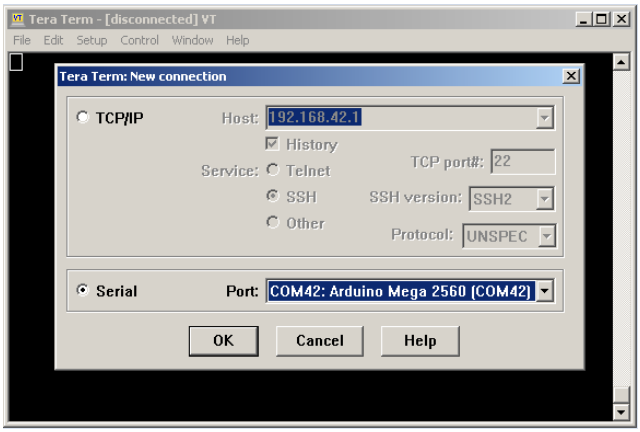


Once the Serial Monitor starts you will see a window similar to the one below. Set the line ending to “Carriage Return” and the BAUD rate to 9600. To send commands, enter the text in the box at the top of the window and then hit “Enter”.

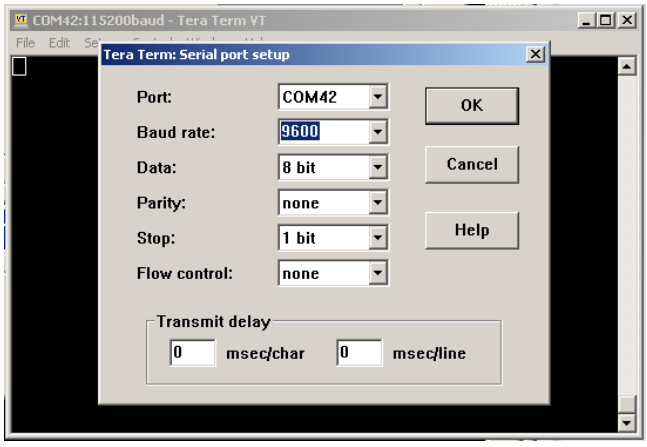


While you can run the battery programmer using the Arduino Serial Monitor, it will not allow sending or receiving hex files over the serial port. Alternatively, you can use any serial terminal program you prefer. One program that works well is TeraTerm. (Windows only)
It can be downloaded from: https://osdn.net/projects/ttssh2/releases/

When you run TeraTerm you will see the following dialog, select Serial Port and then pick the port labeled “Arduino Mega 2560”.



After that select Setup-> Serial Port from the menu and then set the BAUD rate to 9600 as shown below.

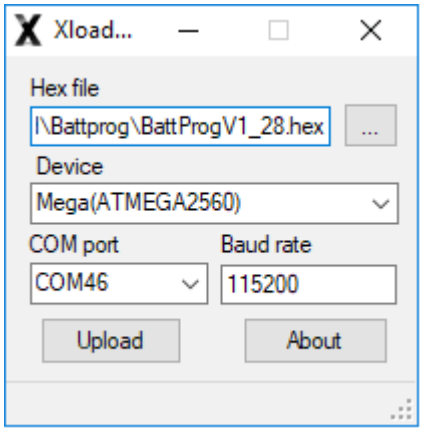


You can use File->Send to load a hex file into the battery programmer, and File->Log to record the box output to a file, which allows you to download a hex file from the battery programmer.

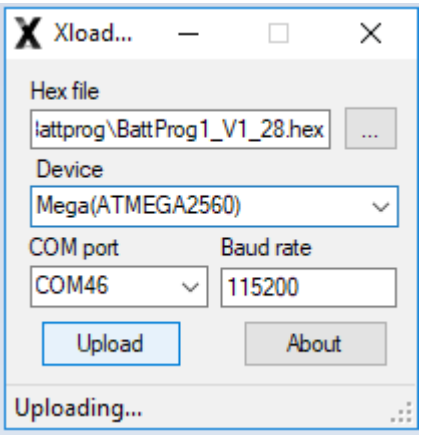
Firmware Updating

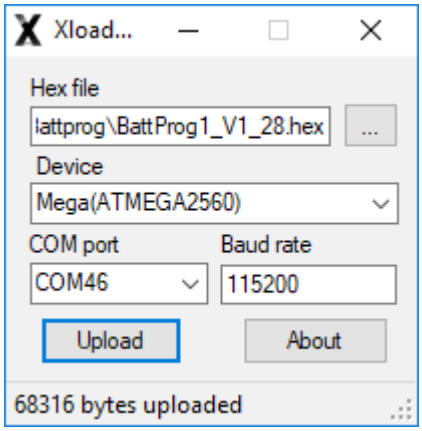
Make sure you have installed the Arduino IDE as described above. Firmware updates are supplied in a zip archive file. The naming convention for these files is: BattprogVx_yy.zip where x.yy is the version number of the firmware. The archive contains a directory called BattProg. Extract this directory to your local machine. Inside the directory you will find a program called Xloader.exe. Run this program.

In the Xloader window set the “COM port” to the serial port your box is connected to. Make sure the “Baud rate” is set to “115200”, and that “Device” is set to “Mega(ATMEGA2560)”, as shown below.



For “Hex file” browse to the “BattProg” directory you just extracted from the archive and select the appropriate hex file. There will be one or two hex files named as follows: BattProgVx_yy.hex, and BattProgVx_yyJ.hex. The x.yy is the version number and the file name that ends with a ‘J’ is the JDM version of the firmware. The other one is the Zebra internal version. Once you have all the settings as shown above, click the “Upload” button. You should see “Uploading” appear in the bottom of the window as shown below. After a minute or so that should change to “xxxxx bytes uploaded” as shown below. At that point the box has been successfully updated.





Command List

Cmd	Function	Notes
B	Select battery type	Pick the type of battery, must be done before other operations.
R(E|A)[1]	Read battery data	Reads entire battery data storage into the buffer.
W(E|A)[1]	Write battery data	Overwrites entire battery data storage with the data in the buffer. Verifies the data after the write.
U(E|A)[1]	Update battery data	Updates the battery using the data in the buffer. Only overwrites fixed data, does not change values set at manufacture time. Verifies the data after the write.
V(E|A)[1]	Verify battery data	Checks the battery data against the data in the buffer
E	Erase buffer	Flush buffer to all 0xFF's
LS	Load hex file into buffer	Loads a hex file from serial port. Does NOT clear buffer first.
DS	Dump hex file from buffer	Dumps a hex file out the serial port from the buffer.
LF	Load buffer from file on SD card	Loads a hex file from the SD card. Does NOT clear buffer first.
SF	Save buffer to file on SD card	Writes a hex file of the buffer to the SD card.
DIR	Directory of SD card	Display contents of the SD card.
VBD	Validate Battery Data	Perform battery data validation (Check checksums, etc..)
MV2	"Manufacture" PP+ V2 battery	Sets SN, date made and other GG data in PP+ V2 battery
DPP	Display PP Data	Displays PP data in human readable form
DPP+	Display PP+ Data	Displays PP+ data in human readable form
A	Authenticate Battery	Performs a full authentication on the battery auth chip.
DGG	Display Gas Gauge Regs	Dumps the gas gauge registers in human readable form
DGGH	Display Gas Gauge Regs in Hex	Dumps the gas gauge registers in hex
DGGF	Dump Gas Gauge Flash	Dumps the gas gauge flash data in hex
SGGF	Save Gas Gauge Flash	Saves the gas gauge flash data into a hex file on the SD card
HGG	Hibernate Gas Gauge	Forces the gas gauge into hibernate mode. Also sets FULLSLEEP
FSGG	Set FULLSLEEP	Sets the FULLSLEEP bit in the gas gauge control register
RGG	Reset gas gauge	Does a reset of the gas gauge
SGG	Seal the gas gauge	Seals the gas gauge
USGG	Unseal the gas gauge	Unseals the gas gauge
FUSGG	Fully unseal the gas gauge	Does a full unseal of the gas gauge
VERB	Toggle verbose mode	Switches output to/from verbose style output
VER	Version	Shows the software/hardware versions
STAT	Show status	Shows the current box status
?	Help	Displays context sensitive list of available commands

1. E or A specifies which device EEPROM or Auth chip to read/write.

Detailed Command Descriptions

Note: Many of the commands use a “buffer” located in RAM to store the data from the battery or to get the data to write into the battery. Data can also be loaded into the buffer from the serial port (In Intel hex file format), or an Intel hex file on an SD card. The data in the buffer can also be written out the serial port in Intel hex file format, or saved onto an SD card as an Intel hex file.

Where mention is made to “erasing” or “flushing” the buffer, that means that the buffer will be filled with hex 0xFF bytes.

B – Select battery type

This command allows the user to choose which battery type is being used. The current choices are:

0: No Battery – This will float the I2C lines so they don't load or drive the battery I2C pins.
1: MPA2 PP – MPA2 style power precision battery. Also know as a “Smart Battery”.
2: MC95 PP
3: MC18 PP+
4: Rouge/TC8000
5: Frenzy/WT6000
6: IronMan/Lightning – TC51/56 PP+ battery.
7: Pollux – TC70/75 PP battery.
8: Falcon/Thunder – TC71/76 PP+ battery.
9: Hawkeye/TC20
10: Sentry/Elektra
11: Galactus/Badger/Firebird/Frozone Auth
12: Frozone EEPROM – Frozone EEPROM, fixed data.
13: Value tier battery
14: PP+ V2 Battery – PP+ battery using new gas gauge

R(E|A) – Read battery data

Read the contents of the battery EEPROM (RE) or the auth chip data area (RA) into the buffer.

W(E|A) – Write battery data

Overwrite the entire contents of the battery EEPROM (WE) or the auth chip data area (WA) using the data currently in the buffer. After writing the data it will verify the write by reading the data back and comparing it to the buffer.

U(E|A) – Update battery data

Update the data in the EEPROM/Auth chip using the data in the RAM buffer. You must load valid data (for the selected battery) into the buffer before using this command or you will corrupt the battery. The command will update the battery data differently depending on what battery type is selected, as follows:

a. For PP+ data the following is done:
The data blocks 0-2 is copied over from the new data.
The data in blocks 3-7 is left untouched. (Dynamic blocks)
The data in blocks 8-14 is copied from the new data.
The data in blocks 15-16 is left untouched. (Cell identifying data)
The data in blocks 17-31 (25 for auth chip data) is copied from the new data.

b. For Pollux data the following is done:
The format revision byte will be updated, only values 0-3 are supported.
Bytes 2-12 will be left untouched. (Part number, rev, serial number, date made)
Bytes 13-239 will be updated.
The checksum at byte 0 will be updated.
Bytes 240-252 will be left untouched. (aggregate charge)
Bytes 253-295 will be updated.
Bytes 296-323 will be left untouched. (Health, part number)
Bytes 324-511 will be updated.

c. Devices that have both types of data, Falcon, IronMan, etc. will have both sets of changes applied.

d. For Hawkeye data the following is done:
The format revision byte will be updated, only values 0-3 are supported.
Bytes 4-39 will be left untouched. (Part number, rev, serial number, date made)
Bytes 40-199 will be updated.
The checksum at byte 0 will be updated.
Bytes 201-215 will be left untouched. (aggregate charge)
Bytes 216-251 will be updated.
Bytes 252-259 will be left untouched. (Health)
Bytes 260-415 will be updated.

V(E|A) – Verify battery data

Checks that the entire contents of the battery EEPROM (VE) or the auth chip data area (VA) matches the data currently in the buffer. It will stop at the first mismatch.

E – Erase buffer

Flushes the buffer to all 0xFF's.

LS – Load Hex file from serial port

Load data in Intel hex file format into the buffer thru the serial port connection to the PC. It does not flush the buffer first so it is possible to load multiple hex files into different sections of the buffer, or to over write just part of the buffer with a small hex file.

DS – Dump Hex file to the serial port

Dump the contents of the buffer in Intel hex file format out the serial port connection to the PC.

LF – Load Hex file from SD card

Load the buffer from an Intel hex file located on an SD card plugged into the box. It does not flush the buffer first so it is possible to load multiple hex files into different sections of the buffer, or to over write just part of the buffer with a small hex file.

SF – Save Hex file to SD card

Saves the contents of the buffer into an Intel hex file on the SD card plugged into the box. You are limited to an 8 character long name. (Not including the ".hex" which is added automatically)

VBD – Validate battery data

Performs a full validation of the data in the battery. Does basically the same steps a terminal does when checking a battery.

1. Checks all check sums and data types for valid values. (Both EEPROM/Auth chip and gas gauge data)
2. For PP+ batts checks that the data in the EEPROM matches the data in the gas gauge.
3. For PP+ batts checks that unused blocks have the correct default pattern.

Note: This command will overwrite the buffer contents with the current EEPROM/Auth chip data.

MV2 – "Manufacture" PP+ V2 battery

Sets the date of manufacture and serial number stored in the V2 gas gauge to user supplied values. Also generates the "Initial data" record based on the current battery conditions.

DIR – Directory of SD card

Displays a list of the files on the SD card plugged into the box.

DPP – Display PP data

Displays the data from the PP section of battery storage in human readable form.

DPP+ – Display PP+ data

Displays the data from the PP+ section of battery storage in human readable form.

A – Authenticate battery

Checks the validity of the battery auth chip to insure it's a valid Zebra battery.

DGG – Display gas gauge registers

Displays selected gas gauge registers in human readable form.

DGGH – Display gas gauge registers in Hex

Dumps all the gas gauge registers in hex out the serial port.

DGGF – Dump gas gauge flash in Hex

Dumps the contents of the gas gauge flash as a hex file out the serial port.

SGGF – Save gas gauge flash

Saves the contents of the gas gauge flash as a hex file on the SD card.

HGG – Hibernate gas gauge

Sets the HIBERNATE and FULLSLEEP bits in the gas gauge control register. If the gauge is not disturbed it will normally enter hibernate mode in a few minutes.

FSGG – Set FULLSLEEP Bit

Sets the FULLSLEEP bit in the gas gauge control register.

RGG – Reset gas gauge

Performs a reset of the gas gauge, similar to a power on reset.

SGG – Seal gas gauge

Sends the “SEALED” command to the gas gauge.

USGG – Unseal gas gauge

Unseals the gas gauge, does not enable full access mode.

FUSGG – Fully unseal gas gauge

Fully unseals the gas gauge, enabling full access mode.

VERB – Toggle verbose mode

Switches test output to/from verbose mode.

VER – Show version information

Displays the version of the software running on the box as well as the hardware revision.

STAT – Show device status

Shows the current status of the box. This includes the type of battery selected as well as the size of the battery memory, the current verbose mode setting, the clock drive type, the pull up resistors and voltage selected, and the current battery voltage.

? – Display help

Displays a list of the available commands.

NOTE: This list is context sensitive and will not show commands the are not available, e.g. there will be no SD card commands shown if there is no SD card plugged in.