# CAN to Bluetooth BRIDGE AND DATALOGGER WITH CAN

## USER MANUAL

**P/N: AX141100**

**In Europe:**
**Axiomatic Technologies Oy**
**Höytämöntie 6**
**33880 LEMPÄÄLÄ - Finland**
**Tel. +358 3 3595 600**
**Fax. +358 3 3595 660**
**www.axiomatic.fi**

**In North America:**
**Axiomatic Technologies Corporation**
**5915 Wallace Street**
**Mississauga, ON Canada L4Z 1Z8**
**Tel. 1 905 602 9270**
**Fax. 1 905 602 9279**
**www.axiomatic.com**

**VERSION HISTORY**

| Version | Date | Author | Modification |
|---|---|---|---|
| 1.0.0. | Nov 18, 2016 | Antti Keränen | Initial Version |
| -- | Nov. 18, 2016 | Amanda Wilkins | Marketing Review |
| 1.0.1. | Mar. 5, 2017 | Antti Keränen | CAN Rx filtering description updated, CAN Scope function and BT MAP Tool description added. |
| 1.0.2. | Mar. 20, 2017 | Antti Keränen | New functionality added (firmware v1.01). SPP command reference updated. |
| -- | March 20, 2017 | Amanda Wilkins | Updated connection and operating range in Technical Spec. |
| 1.0.3. | June 12, 2017 | Antti Keränen | Bluetooth node ID configuration added. BT module communication failure detection added (in section 1). |
| 1.0.4. | June 15, 2017 | Antti Keränen | CAN RX Filtering description enhanced. |
| 1.0.5. | February 16, 2018 | Antti Keränen | Android apps' links updated. |
| 1.0.6. | February 26, 2018 | Antti Keränen | Default PIN codes note added. |

**DEFAULT PIN CODES**
**Pairing: 000000**
**Configuration mode: 000000**

## ACRONYMS

| | |
|---|---|
| ACK | Positive Acknowledgement (from SAE J1939 standard) |
| BATT +/- | Battery positive (a.k.a. Vps) or Battery Negative (a.k.a. GND) |
| BD ADDR | Bluetooth Device Address |
| BLE | Bluetooth Low Energy |
| BT | Bluetooth |
| EA | Electronic Assistant®, p/n AX070502 (A Service Tool for Axiomatic ECUs) |
| ECU | Electronic Control Unit (from SAE J1939 standard) |
| GND | Ground reference (a.k.a. BATT-) |
| MAP | Memory Access Protocol |
| PGN | Parameter Group Number (from SAE J1939 standard) |
| SPN | Suspect Parameter Number (from SAE J1939 standard) |
| TP | Transport Protocol |

# LIST OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

## REFERENCES

| | |
|---|---|
| J1939 | Recommended Practice for a Serial Control and Communications Vehicle Network, SAE, April 2011 |
| J1939/21 | Data Link Layer, SAE, December 2010 |
| J1939/71 | Vehicle Application Layer, SAE, March 2011 |
| J1939/73 | Application Layer-Diagnostics, SAE, February 2010 |
| J1939/81 | Network Management, SAE, May 2003 |
| TDAX141100 | Technical Datasheet, CAN to Bluetooth bridge and datalogger with CAN, Axiomatic Technologies 2016 |
| UMAX07050x | User Manual V5.13.84, Electronic Assistant and USB-CAN, Axiomatic Technologies, October 2016 |

***This document assumes the reader is familiar with the SAE J1939 standard. Terminology from the standard is used, but not described in this document.***

NOTE: This product is supported by Electronic Assistant® V5.13.84.0 and higher.

# 1. Overview of The Controller

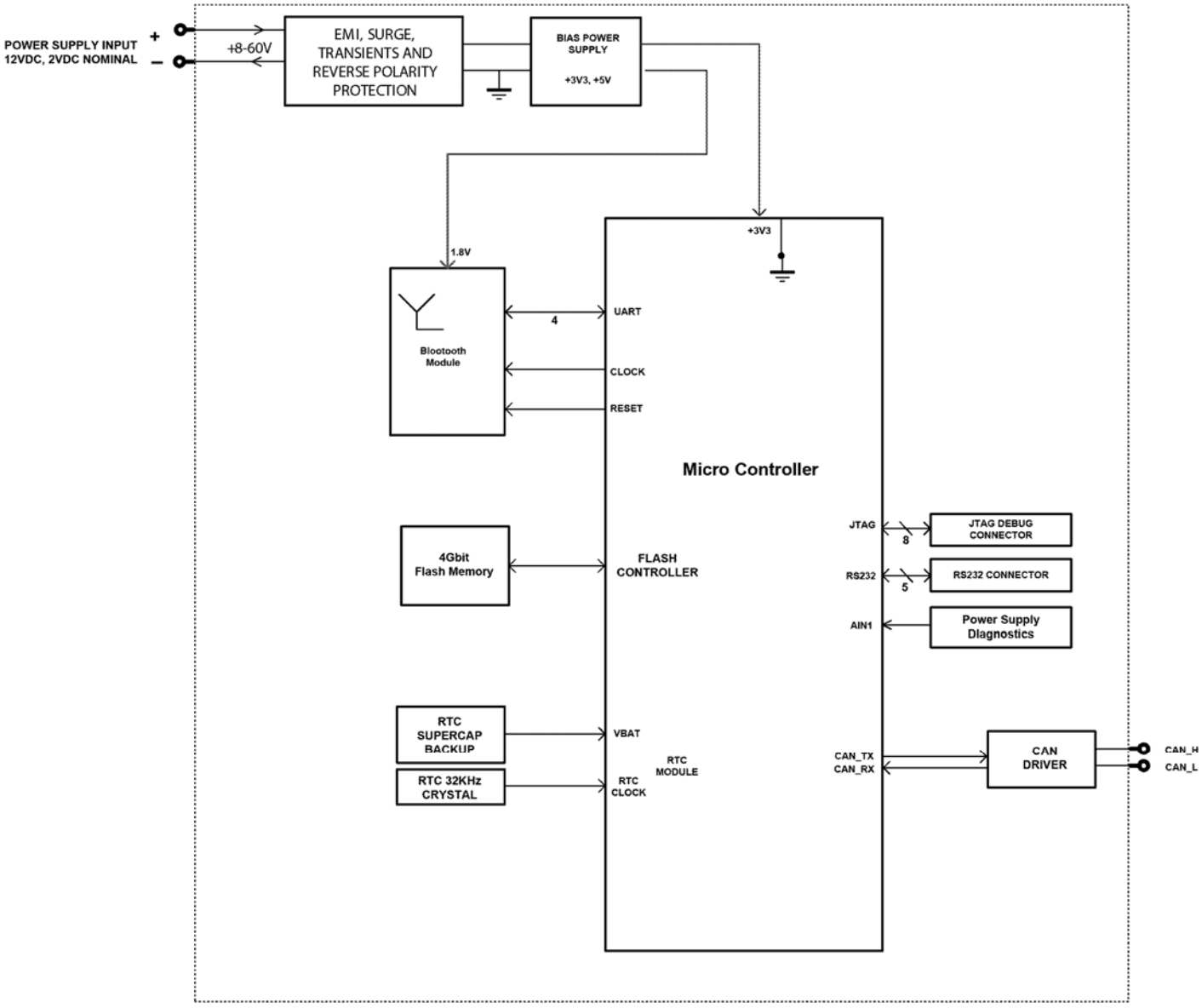



**Figure 1 – AX141100 Block Diagram**

The CAN to Bluetooth device (later CAN2BT) can be used for accessing the CAN bus and the other CAN nodes connected to it using Bluetooth communications and a smart device, such as a phone or a tablet.

The CAN2BT device provides also data logging capabilities. It has 4GBits of internal Flash storage for storing the selected CAN messages with time stamp information.

The CAN2BT device can be configured using an Android application called *CAN2BT Configuration*. With this tool, all the functionality of the CAN2BT device can be configured, such as setting PIN codes, connection options, CAN receive filtering and CAN data logging rules. Also RTC time setting and different Flash memory operations such as download data and data erase are supported.

The CAN2BT device can be operated in two main operation modes, namely the Interface Mode and the Bridge Mode. The Interface Mode is targeted for data logging operation and in this mode the CAN2BT device can be accessed using Electronic Assistant.

When in Bridge Mode, the CAN receive filtering is disabled and the device accepts all incoming CAN frames. The main purpose of the Bridge Mode is to use two CAN2BT devices as a wireless CAN data bridge between two different CAN networks. When in Bridge Mode, the CAN2BT device is transparent to the CAN bus and cannot be seen in Electronic Assistant's list of J1939 nodes. The device can be still accessed using the *CAN2BT Configuration* tool.

In case the module startup fails (a communication error between the CPU and the Bluetooth module is detected), the CAN2BT device will indicate the failure by sending the following DM1 (PGN 0xFECA / 65226) to CAN bus on 1s intervals. In addition to the DM1, the two built-in LEDs are blinking and not responding to user configuration nor control messages. In failure mode, the unit will be accessible only via the CAN bus.

| ***Condition*** | ***Lamp*** | ***SPN*** | ***FMI*** | ***OC*** | ***Src.Address*** |
|---|---|---|---|---|---|
| BT module failure | MALFUNCTION | 3368 (WCM1) | 31 | 1 | 0xFB (251) |

# 2. CAN to Bluetooth Function Blocks

This section explains the different functions and configuration available on the CAN2BT. The *CAN2BT Configuration* Android application is used as a reference. The application is available from Google Play.

## 2.1. PIN codes


**Figure 2 – PIN Code settings**

The PIN Code settings allow the user to define PIN codes that are used when accessing the device. **Pairing PIN** is the PIN code that will be used when pairing the CAN2BT device with a new smart device. **Configuration PIN** code is required when configuration is accessed (most configuration options require that the CAN2BT device is put to configuration mode. Entering the configuration mode requires the user to enter the configuration PIN). **Remote Connection PIN** is the code that is used when the CAN2BT makes a connection with another CAN2BT device (when forming a data brigde for wireless CAN communications).

**DEFAULT PIN CODES**
**Pairing: 000000**
**Configuration mode: 000000**

## 2.2. CAN Bus Configuration


**Figure 3 – CAN bus configuration**

CAN bus configuration allows changing the CAN interface baudrate. The list of available baudrate options include 50k, 100k, 125k, 250k (default), 500k and 1M. When changing the baudrate, the CAN2BT device needs to be restarted (power cycled) to apply the new baudrate to the CAN interface.

The **Echo TX CAN Frames** makes the CAN2BT device to send back all CAN messages that are sent to its Bluetooth interface using commands 0x12 and 0x13. Please see section 5.2 for more info about CAN message sending.

## 2.3. Connection Options


**Figure 4 – Connection options**

Connection options menu list the functions available for connecting the CAN2BT device to other CAN2BT node. **Scan** starts a scan for other Bluetooth nodes (CAN2BT works as a master). **Connect to node makes** a connection to one of the nodes returned in the Scan results. **Disconnect from node** disconnects the currently active connection. **Set autoconnect** can be used for setting a BD ADDRESS of a remote node with which the CAN2BT device will attempt to connect automatically at power on (targeted for data bridge operations). **Disable autoconnect** will reset the BD ADDRESS set using command **Set autoconnect**.

**Define accepted BD ADDR** lets the user to configure a Bluetooth address that will be the only address from which connections are allowed. All other nodes trying to connect to the CAN2BT device will be ignored.

All functions that are used for defining BD ADDRESSes expect to receive an index number that matches to the BD ADDR list returned upon successful device scan (using the **Scan** command). The list of valid numbers is from 1 to number of Bluetooth nodes found. Index 0 is also accepted, it is interpreted as the BD ADDR of the device that is currenty used for configuring the device. This is useful in cases where the accepted BD ADDR needs to be set.

## 2.4. CAN Receive Filtering


**Figure 5 – CAN Receive Filtering**

By default the device won't read in all received CAN messages. With the CAN receive filtering functions, the user can change this behaviour. **Disable CAN fitering** removes all CAN receive filtering and makes the CAN2BT device capable of receiving all CAN frames on the bus. With CAN filtering disabled, the CAN2BT device is invisible to the CAN bus. This is the mode that is preferred for using the device as a bridge between two CAN buses.

**Enable CAN filtering** restores the default CAN receive filtering and makes the CAN2BT device visible to the CAN bus (it will respond to Address Claim messages).

**Add CAN RX filter** lets the user to define a single CAN filter and optional mask that are added to the low level (HW) CAN receive filter registers. Please note, that if the mask is not specified, the firmware uses a default mask of 29 (or 11) bits wide that will compare all bits in the received CAN frame against the specified filter.

Up to 28 filter definitions will be accepted. The **Remove CAN RX filter** will remove the specified filter (the removal is done by comparing the IDs of the configured filters and the ID specified to be removed).

The CAN filter definition for adding and removing filters is as follows:

| bit 31 | bit 30 | bits 29-0 |
|---|---|---|
| Use ExtID | Use RTR | Filter ID |

**Table 1 – CAN Rx filter definition**

*Example 29bit ID filter & mask for receiving all (J1939) frames with Source Address 0xF9*
`Filter: 800000F9, Mask: FF`
*Example 11bit ID filter & mask for receiving ID 0x701 (CANopen master HB)*
`Filter: 701, Mask: 7FF`

## 2.5. CAN Data Logging


**Figure 6 – CAN Logging Configuration**

The Datalogging configuration includes defining different PGNs or Source Addresses to log. These two functions assume PGNs and Source Addresses as defined in the J1939 specification. The **Define ID to log** allows the user to freely define a CAN frame ID that will be logged.

Please note, that the CAN Logging configuration does not change the CAN receive filtering. In most cases (unless the device has CAN receive filtering disabled), the user has to define proper CAN receive filters in order the CAN2BT to be able to receive all CAN frames that need to be logged.

The current data logging rules can be read using the **List all logging rules** function. The **Erase all logging rules** will erase all rules.

## 2.6. Flash Memory Operations


**Figure 7 – Flash memory operations**

Contents of the Flash memory can be downloaded to an Android smart device using the Download flash contents. This function will transfer 96 bytes of the Flash contents at a time, please refer to section 5 for details.

The Flash can be erased few blocks at a time or then a full erase can be done using Erase all flash blocks. The Flash erase functions will require that the device is in configuration mode, so the Flash contents are protected with the Configuration PIN.

The current flash address (block and page) can be set using the Flash address function. All the following Flash writes will use this setting.

Erasing a single block can be done by setting the erase start block and end block to a same value (please note that the flash erase is flash block specific). To download a single flash page, the start address (block & page) and end address (block & page) need to be set to a same value.

| | |
|---|---|
| **Flash type** | Micron MT29F4G08ABADA (4GBits) |
| **Bytes per page** | 2112 |
| **Pages per block** | 64 |
| **Blocks in total** | 4096 |

**Table 2 – Flash chip details**

## 2.7. Real Time Clock Settings


**Figure 8 – Real Time Clock**

There is an RTC on the CAN2BT device. The RTC is powered from a supercap, capable of keeping the time and data information for 96 hours of stand-by if fully charged before the stand-by period. The fully charging means that the CAN2BT device is being powered at least 8 hours continously.

The time can be set using the **Set time** and read using the **Get time** function. Setting the time requires that the CAN2BT device is in configuration mode.

## 2.8. Miscellaneous Settings


**Figure 9 – Miscellaneous settings**

The miscellaneous settings consist of setting the CAN2BT device to the configuration mode, and exiting the configuration mode. The CAN2BT device can be also SW reset and the CAN bootloader can be started for firmware updates (please also see section 8).

The **Enter Configuration** mode requires the user to enter the current Configuration PIN number. In case the smart device is disconnected from the CAN2BT device during configuration, the CAN2BT device automatically exits the confitguration mode.

**SW Reset** will reset the CPU of the CAN2BT device. **Start bootloader** will set the start bootloader flag and then reset the CPU. This makes the CAN2BT device enter the firmware reflash mode, and it will be visible on the CAN bus and accessible from EA as Bootloader #1 (please see section 8 for more detailed description of firmware reflashing). **Default settings** will restore factory default settings and then reset the CPU.

**Bluetooth ID** allows the user to configure the name that device will advertise. The default is “CAN2BT”. In case there are multiple CAN2BT devices in range, it might be advantageous to configure unique names to different controllers. The Bluetooth ID accepts charaters in range 0x20 (‘space’) to 0x7E (‘~’) and can hold up to 248 characters.

The configured Bluetooth ID is also available in the J1939 Software ID (SPN 234).

In case the CAN2BT device is not in configuration mode and the user tries to configure it, the configuration commands return non-zero values.

## 2.9. CAN Scope

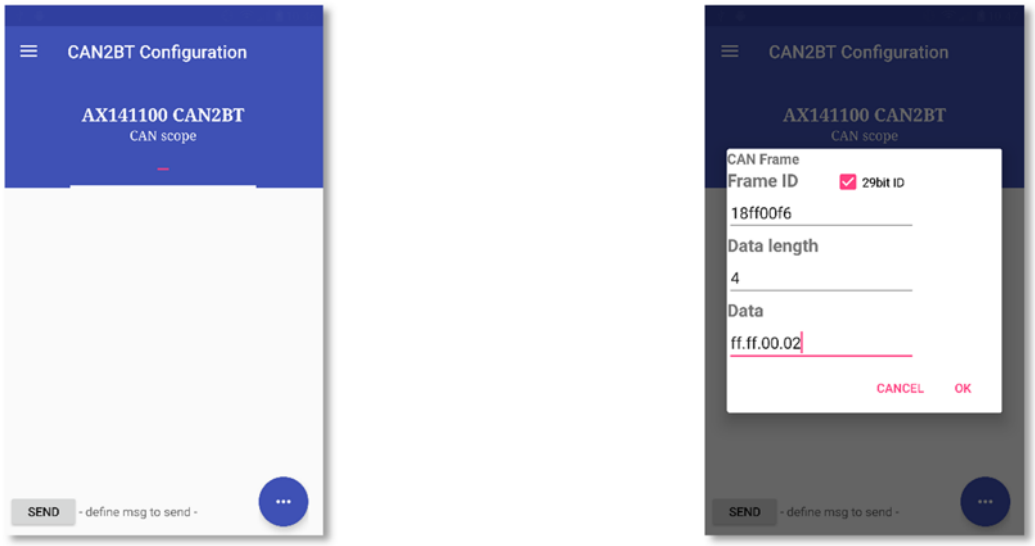



**Figure 10 – CAN Scope**

The CAN Scope function shows the frames on the CAN bus. Only the frames that pass the receive filters (please see section 2.4) are shown. The CAN Scope function can be also used for manually sending CAN frames to tbe bus.

The data for the frame to be sent is given in hexadecimal format. The data bytes for the CAN frame need to be given in sequence of bytes. The separator between the bytes can be one of the following characters:

**: ; - , .**

## 2.10. LED Configuration


**Figure 11 – LED Configuration**

The build in LEDs of the CAN2BT device can be also manually controlled. The LED Mode function allows the user to set the LED function, default – LED activity based on BT RX and TX and manual – LED states as commanded using the LED commands. Please see section 5.2 for more detailed info about the available LED commands.

# 3. Installation Instructions

## 3.1. Dimensions and Pinout

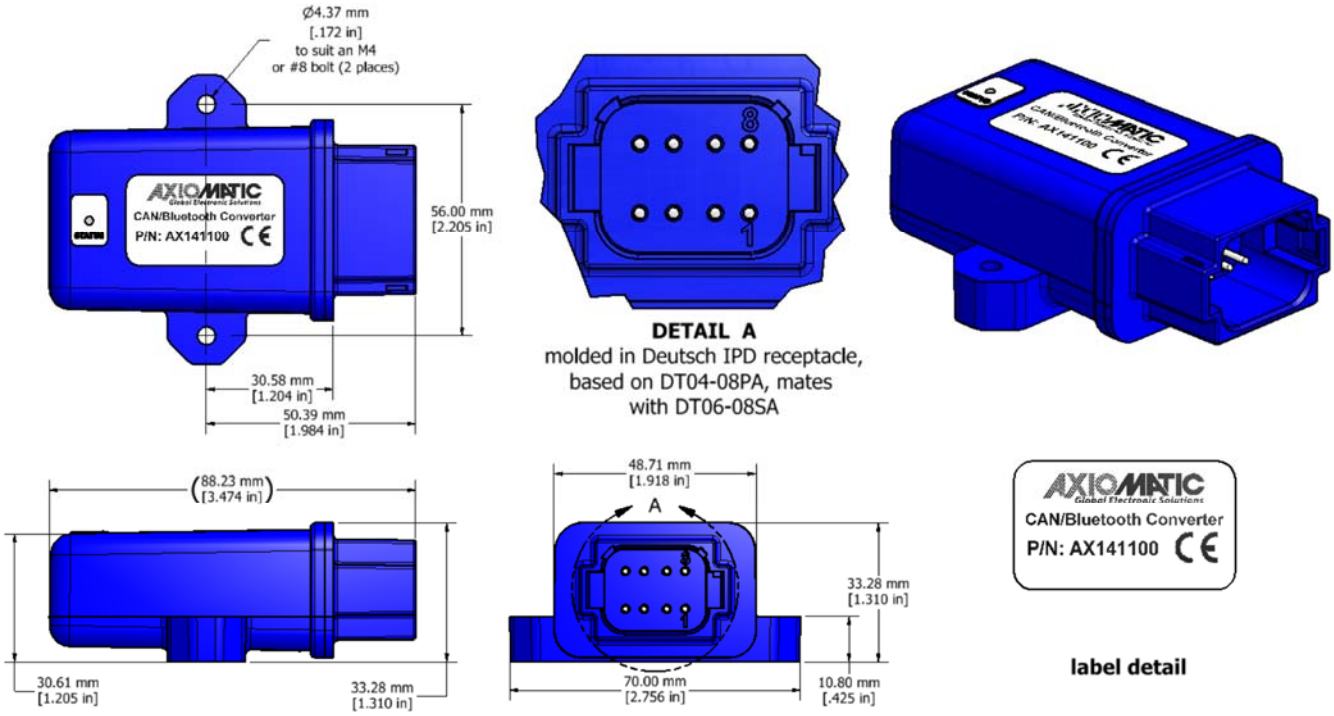



**Figure 12 – AX141100 Dimensional Drawing**

| PIN # | FUNCTION |
|---|---|
| 1 | CAN_L |
| 2 | CAN_H |
| 3 | CAN _SHIELD |
| 4 | NOT USED |
| 5 | NOT USED |
| 6 | NOT USED |
| 7 | BATT + |
| 8 | BATT - |

**Table 3 – AX141100 Connector Pinout**

# 4. Overview of J1939 Features

The software was designed to provide flexibility to the user with respect to messages sent from the ECU by providing:

- Configurable ECU Instance in the NAME (to allow multiple ECUs on the same network)

## 4.1. Introduction to Supported Messages

The ECU is compliant with the standard SAE J1939, and supports following PGNs from the standard.

**From J1939-81 – Network Management**

- Address Claimed/Cannot Claim 60928 0x00EE00
- Commanded Address 65240 0x00FED8

**From J1939-71 – Vehicle Application Layer**

- Software Identification 65242 0x00FEDA

Setpoints are accessed using standard Memory Access Protocol (MAP) with proprietary addresses. The Electronic Assistant® (EA) allows for quick and easy configuration of the unit over CAN network.

## 4.2. NAME, Address and Software ID

The CAN to Bluetooth ECU has the following default for the J1939 NAME. The user should refer to the SAE J1939/81 standard for more information on these parameters and their ranges.

| | |
|---|---|
| Arbitrary Address Capable | Yes |
| Industry Group | 0, Global |
| Vehicle System Instance | 0 |
| Vehicle System | 0, Non-specific system |
| Function | 130, On-board data logger |
| Function Instance | 1, Axiomatic AX141100 |
| ECU Instance | 0, First Instance |
| Manufacture Code | 162, Axiomatic Technologies |
| Identity Number | Variable, uniquely assigned during factory programming for each ECU |

The ECU Instance is a configurable setpoint associated with the NAME. Changing this value will allow multiple ECUs of this type to be distinguishable from one another when they are connected on the same network.

The default value of the “ECU Address” setpoint is 251 (0xFB), which is the default for an on board data logger device. The EA will allow the selection of any address between 0 and 253. ***It is the user’s responsibility to select an address that complies with the standard***. The user must also be aware that since the unit is arbitrary address capable, if another ECU with a higher priority NAME contends for the selected address, the CAN to Bluetooth device will continue select the next highest address until it finds one that it can claim. See J1939/81 for more details about address claiming.

**Software Identifier**

| PGN 65242 | | Software Identification | - SOFT |
|---|---|---|---|
| Transmission Repetition Rate: | | On request | |
| Data Length: | | Variable | |
| Extended Data Page: | | 0 | |
| Data Page: | | 0 | |
| PDU Format: | | 254 | |
| PDU Specific: | | 218 PGN Supporting Information: | |
| Default Priority: | | 6 | |
| Parameter Group Number: | | 65242 (0xFEDA) | |
| Start Position | Length | Parameter Name | SPN |
| 1 | 1 Byte | Number of software identification fields | 965 |
| 2-n | Variable | Software identification(s), Delimiter (ASCII “*”) | 234 |

Byte 1 is set to 5, and the identification fields are as follows.

| **(Part Number)*(Version)*(Date)*(Owner)*(Description)** |
|---|

The EA shows all this information in “General ECU Information”, as shown below.

*Note: The information provided in the Software ID is available for any J1939 service tool which supports the PGN -SOFT.*

# 5. SPP Communications

The communications between a smart device such as a phone or a tablet and the CAN2BT device is based on Bluetooth Serial Port Profile (SPP). By default, the CAN2BT firmware declares itself as a SPP device (UUID: 00001101-0000-1000-8000-00805F9B34FB)

The messages are transferred in binary format, least significant byte first. The list of supported proprietary messages is shown below.

## 5.1. Overall message format

There is an ack response sent by the CAN2BT device after receiving the configuration messages.

The overall message format:

| Byte 0 | Byte 1 | Byte 2 | Byte 3 | … | Byte n+2 | Byte n+3 | Byte n+4 | Byte n+5 | Byte n+6 |
|---|---|---|---|---|---|---|---|---|---|
| <msg type> | <length> | <payload 0> | <payload 1> | … | <payload length-1> | <crc32> | <crc32> | <crc32> | <crc32> |

**Table 4 – Overall message format**

in which the <*msg type*> is as listed in below. <*length*> is full message length without the four CRC32 bytes. CRC32 is selected because the support for it is readily available in Android.

All data that is expressed as Byte 0, Byte 1, … in the message descriptions below, is expected to be either 16 bits or 32 bits wide data, broken down to bytes (8 bits) least significant byte first. The only exception is the PIN code data, that is expected to be formatted one digit per byte. The PIN codes are hard formatted to have 6 digits.

Available CAN baudrate options include: 0 – 50k, 1 – 100k, 2 – 125k, 3 – 250k, 4 – 500k, 5 – 1M

## 5.2. Message types

| Message type | Direction | <msg type> byte |
|---|---|---|
| CAN data with StdID | SD* -> CAN2BT | 0x12 |
| CAN data with ExtID | SD* -> CAN2BT | 0x13 |
| CAN data with StdID | CAN2BT -> SD* | 0x21 |
| CAN data with ExtID | CAN2BT -> SD* | 0x31 |
| MAP write | SD* -> CAN2BT | 0x34 |
| MAP write response | CAN2BT -> SD* | 0x43 |
| MAP read | SD* -> CAN2BT | 0x45 |
| MAP read response | CAN2BT -> SD* | 0x54 |
| Download Flash contents | SD* -> CAN2BT | 0x56 |
| Download Flash contents response | CAN2BT -> SD* | 0x65 |
| Download Flash contents ACK | SD* -> CAN2BT | 0x57 |
| Download Flash contents ACK response | CAN2BT -> SD* | 0x75 |
| Change configuration | SD* -> CAN2BT | 0x67 |
| Change configuration response | CAN2BT -> SD* | 0x76 |
| LED state | SD* -> CAN2BT | 0x78 |
| LED state response | CAN2BT -> SD* | 0x87 |

* SD = Smart Device, a phone or a tablet

**Table 5 – Proprietary SPP message types**

### 5.2.1. CAN messages

| **CAN frame format, StdID** (ID bit 11 = RTR flag, bits 10 to 0, StdID) | | | | | | |
|---|---|---|---|---|---|---|
| <ID 0> | <ID 1> | <len> | <D0> | <D1> | … | <D len-1> |

| **CAN frame format, ExtID** (ID bit 31 = RTR flag, bits 28 to 0, ExtID) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| <ID 0> | <ID 1> | <ID 2> | <ID 3> | <len> | <D0> | <D1> | … | <D len-1> |

### 5.2.2. MAP access

| **J1939 MAP access** (data types: 1=u8, 2=u16, 4=u32, 5=f32) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| <Remote node addr> | <SP A0> | <SP A1> | <SP A2> | <data type> | <SP D0> | <SP D1> | <SP D2> | <SP D3> |

### 5.2.3. Configuration messages

| **Configuration mode** (d0 … d5 are single digits of the PIN code set using command 0x21) | | | | | | | |
|---|---|---|---|---|---|---|---|
| Enter config mode | **0xC0** | <d0> | <d1> | <d2> | <d3> | <d4> | <d5> |
| Exit config mode | **0xC1** | | | | | | |

| **CAN Rx Filters** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Add CAN Filter | **0x01** | <filter b0> | <filter b1> | <filter b2> | <filter b3> | <mask b0> | <mask b1> | <mask b2> | <mask b3> |
| Remove CAN Filter | **0x02** | <filter b0> | <filter b1> | <filter b2> | <filter b3> | | | | |
| Enter promiscuous mode | **0x03** | | | | | | | | |
| Exit promiscuous mode | **0x04** | | | | | | | | |

| **CAN bus configuration** (br index = 0-50k, 1-100k, 2-125k, 3-250k, 4-500k, 5-1M, state = 0-off, 1-on) | | |
|---|---|---|
| CAN bus baudrate | **0x05** | <br index> |
| Set BT CAN TX echo | **0x06** | <state> |

| **Connection functions** | | |
|---|---|---|
| Scan available devices | **0x10** | |
| Connect to remote device | **0x11** | <scan index> |
| Disconnect from remote device | **0x12** | |
| Set autoconnect | **0x13** | <scan index> |
| Define accepted BD ADDR | **0x14** | <scan index, 0 = currently connected device> |
| Disable autoconnect | **0x15** | |

| **PIN Codes** (o0…o5 are single digits of the *OLD* PIN code and n0…n5 are single digits of the *NEW* PIN) | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Set Pairing PIN Code | **0x20** | <o0> | <o1> | <o2> | <o3> | <o4> | <o5> | <n0> | <n1> | <n2> | <n3> | <n4> | <n5> |
| Set Config PIN Code | **0x21** | <o0> | <o1> | <o2> | <o3> | <o4> | <o5> | <n0> | <n1> | <n2> | <n3> | <n4> | <n5> |
| Set Rem.Acc. PIN Code | **0x22** | <o0> | <o1> | <o2> | <o3> | <o4> | <o5> | <n0> | <n1> | <n2> | <n3> | <n4> | <n5> |

| **Data logging** (when defining ID, bit 31 = ExtID, bit 30 = RTR, bits 29-0 define the ID) | | | | | |
|---|---|---|---|---|---|
| Define PGN to log | **0x30** | <PGN 0> | <PGN 1> | <PGN 2> | <PGN 3> |
| Define SrcAddress to log | **0x31** | <Src Address> | | | |
| Define ID to log | **0x32** | <ID 0> | <ID 1> | <ID 2> | <ID 3) |
| List current rules | **0x33** | | | | |
| Erase all rules | **0x34** | | | | |

| **Flash functions** | | | | | |
|---|---|---|---|---|---|
| Erase all flash data | **0x40** | | | | |
| Erase specific flash blocks | **0x41** | <start blk b0> | <start blk b1> | <end blk b0> | <end blk b1> |
| Set flash address | **0x42** | <block b0> | <block b1> | <page b0> | <page b1> |

| **RTC** (hour=0…23, min&sec=0…59, day=1…31, weekday=1(mon)…7(sun), month=1…12, year=16… (16=2016) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Get time | **0x50** | | | | | | | |
| Set time | **0x51** | <hour> | <min> | <sec> | <day> | <weekday> | <month> | <year> |

| **LEDs** (mode: 0 – default, 1 – use commands, state: 0 – off, 1 - on) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Set LED mode | **0x52** | <mode> | | | | | | |

| **Misc. functions** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| SW reset | **0xF0** | <'r'> | <'e'> | <'s'> | <'e'> | <'t'> | | | |
| Start bootloader | **0xF1** | <'b'> | <'l'> | <'o'> | <'a'> | <'d'> | <'e'> | <'r'> | |
| Default settings | **0xF2** | <'d'> | <'e'> | <'f'> | <'a'> | <'u'> | <'l'> | <'t'> | <'s'> |
| Bluetooth ID | **0x60** | <chr 1> | … | <chr n> | | | | | |

### 5.2.4. Download flash contents

| **Download flash contents** (if start blk=0, end blk=65535, downloading all data) | | | | | | | |
|---|---|---|---|---|---|---|---|
| <start blk 0> | <start blk 1> | <end blk 0> | <end blk 1> | <start pg 0> | <start pg 1> | <end pg 0> | <end pg 1> |

| **Download flash contents response** | | | | | | | |
|---|---|---|---|---|---|---|---|
| <chunk no#> | <data 0> | <data 1> | … | <data n> | | | |

| **Download flash contents ACK** | | | | | | | |
|---|---|---|---|---|---|---|---|
| <chunk no#> | | | | | | | |

| **Download flash contents ACK response** (status 0:OK, 1:no more data available) | | | | | | | |
|---|---|---|---|---|---|---|---|
| <status> | | | | | | | |

### 5.2.5. LED commands

| **LED state** (LED #0: Green, LED #1: Red. LED state: 0=off, 1=on) | | | | | | | |
|---|---|---|---|---|---|---|---|
| <LED no#> | <LED state> | | | | | | |

| **LED state response** (status 0:OK, 1:failed) | | | | | | | |
|---|---|---|---|---|---|---|---|
| <status> | | | | | | | |

# 6. ECU Setpoints Accessed with Electronic Assistant

Currently the CAN2BT device does not have any specific setpoints. Electronic Assistant (later EA) can be used for setting the J1939 NAME parameters when the device is in Interface Mode (when in Bridge Mode, the CAN2BT device is invisible to EA and to other J1939 devices on the bus).

The CAN bootloader can be started using EA, when the device is configured to operate in Interface Mode.

# 7. Accessing Remote Axiomatic ECUs with BT MAP Tool

The CAN2BT device can access the configuration setpoints of other Axiomatic CAN devices connected to the CAN bus. The Android app for this is called the BT MAP Tool, available from Google Play.

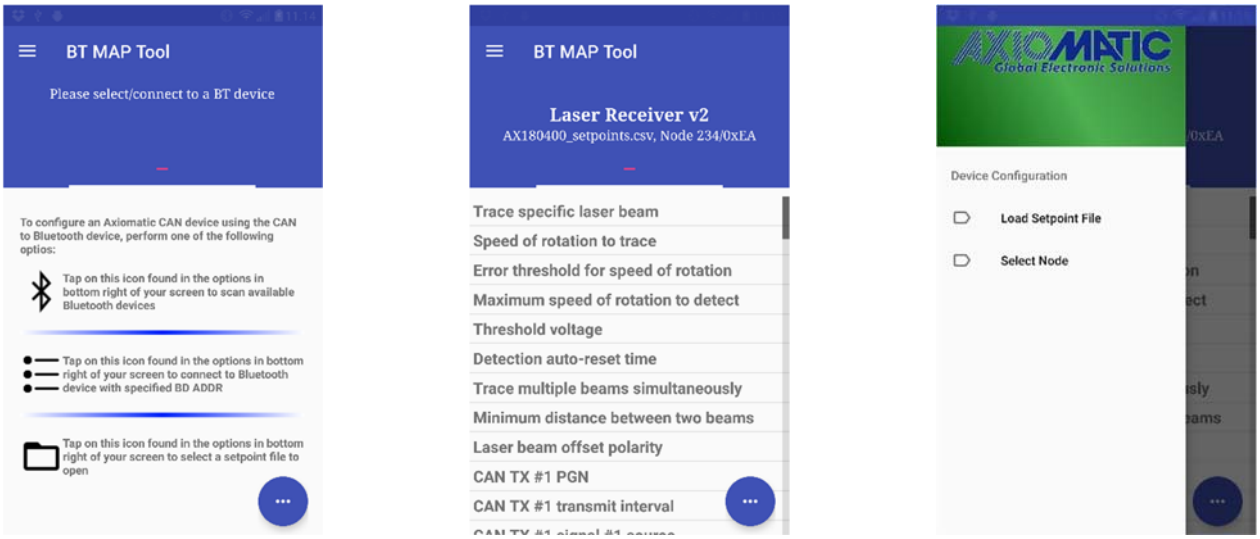



**Figure 13 – BT MAP Tool**

The BT MAP Tool provides an EA like configuration interface to Axiomatic CAN controllers from an Android smart device such as a phone or a tablet.

The BT MAP Tool reads in a setpoint definition file and uses the setpoint addresses and data types specified in that file for accessing the remote Axiomatic CAN device. There are setpoint definition files included in the APK by default, but there are no restrictions of adding more definition files manually.

All Axiomatic CAN controllers that have EA support can be accessed and configured using the BT MAP tool.

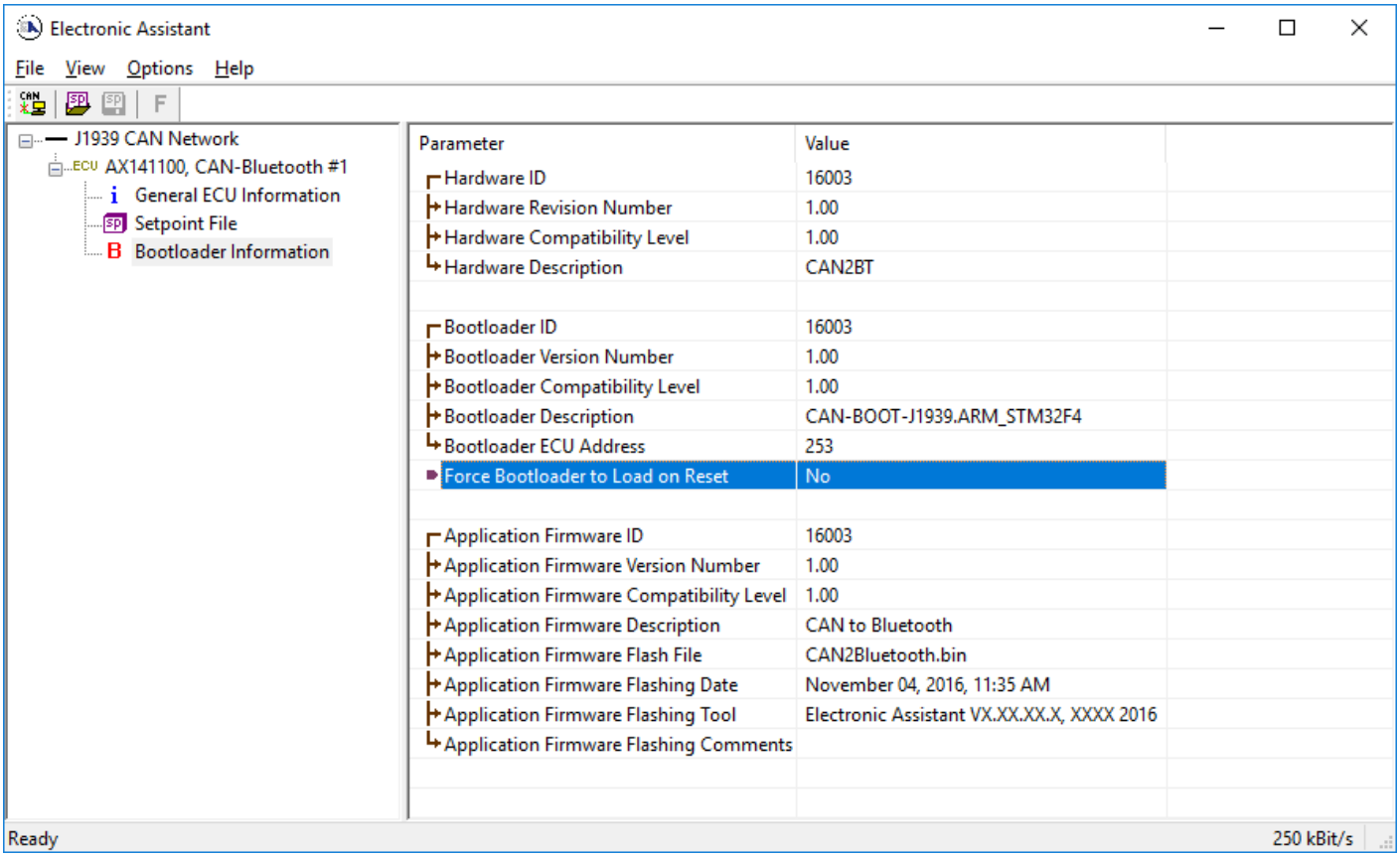
# 8. Reflashing over CAN with EA Bootloader

The AX141100 can be upgraded with new application firmware using the **Bootloader Information** section. This section details the simple step-by-step instructions to upload new firmware provided by Axiomatic onto the unit via CAN, without requiring it to be disconnected from the J1939 network.

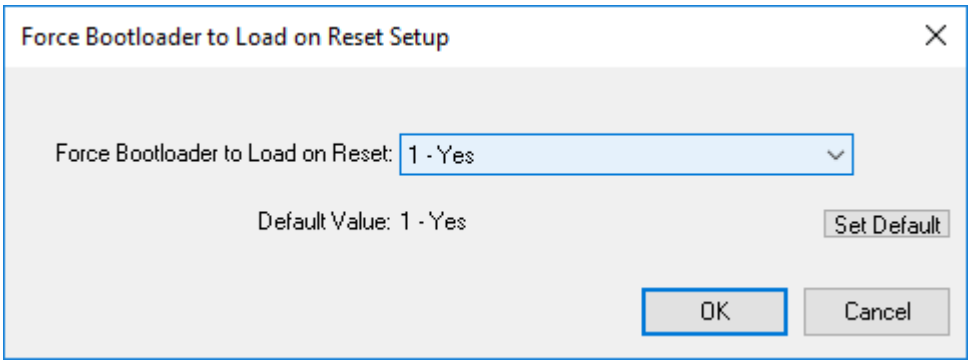
*Note: To upgrade the firmware use Electronic Assistant ® V5.13.84.0 or higher.* ***Further, the AX141100 device needs to be in Interface Mode*** *(CAN receive filtering enabled) in order to be accessible by EA.*

*Note: In case it is preferred not to set the device to Interface Mode, see Step 3.1 below.*

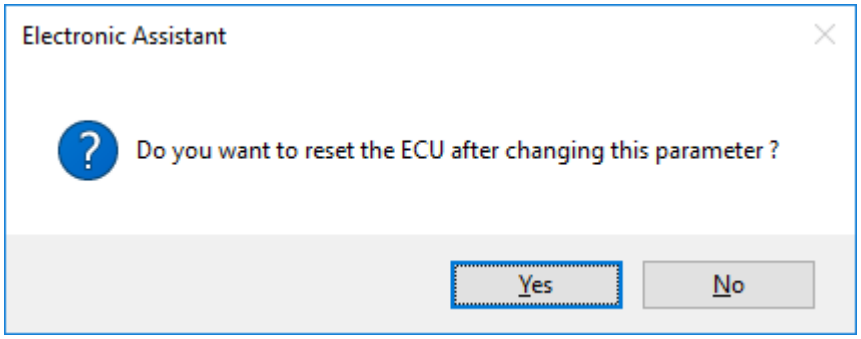
1. When EA first connects to the ECU, the **Bootloader Information** section will display the following information.



2. To use the bootloader to upgrade the firmware running on the ECU, change the variable **"Force Bootloader To Load on Reset**" to *Yes.*

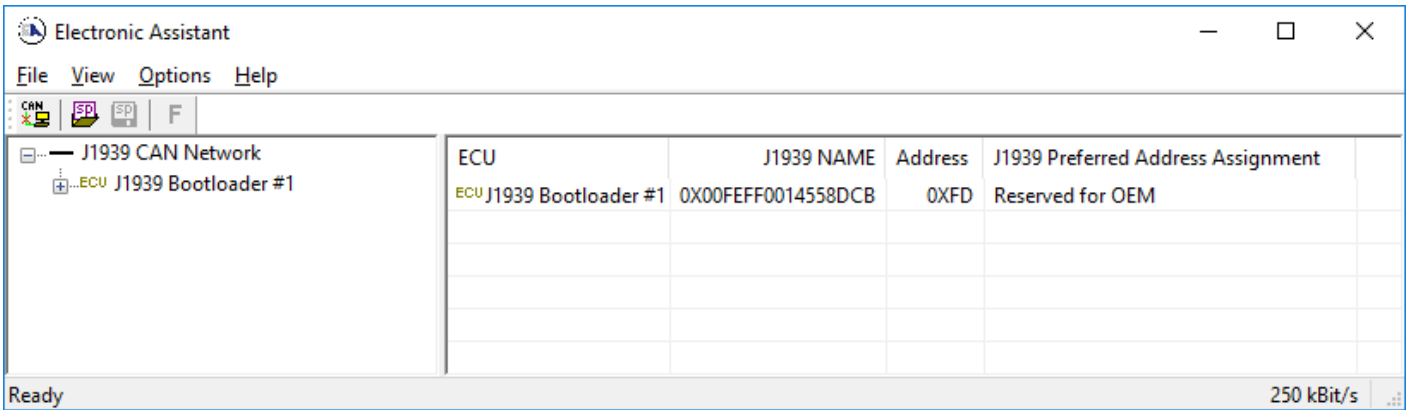


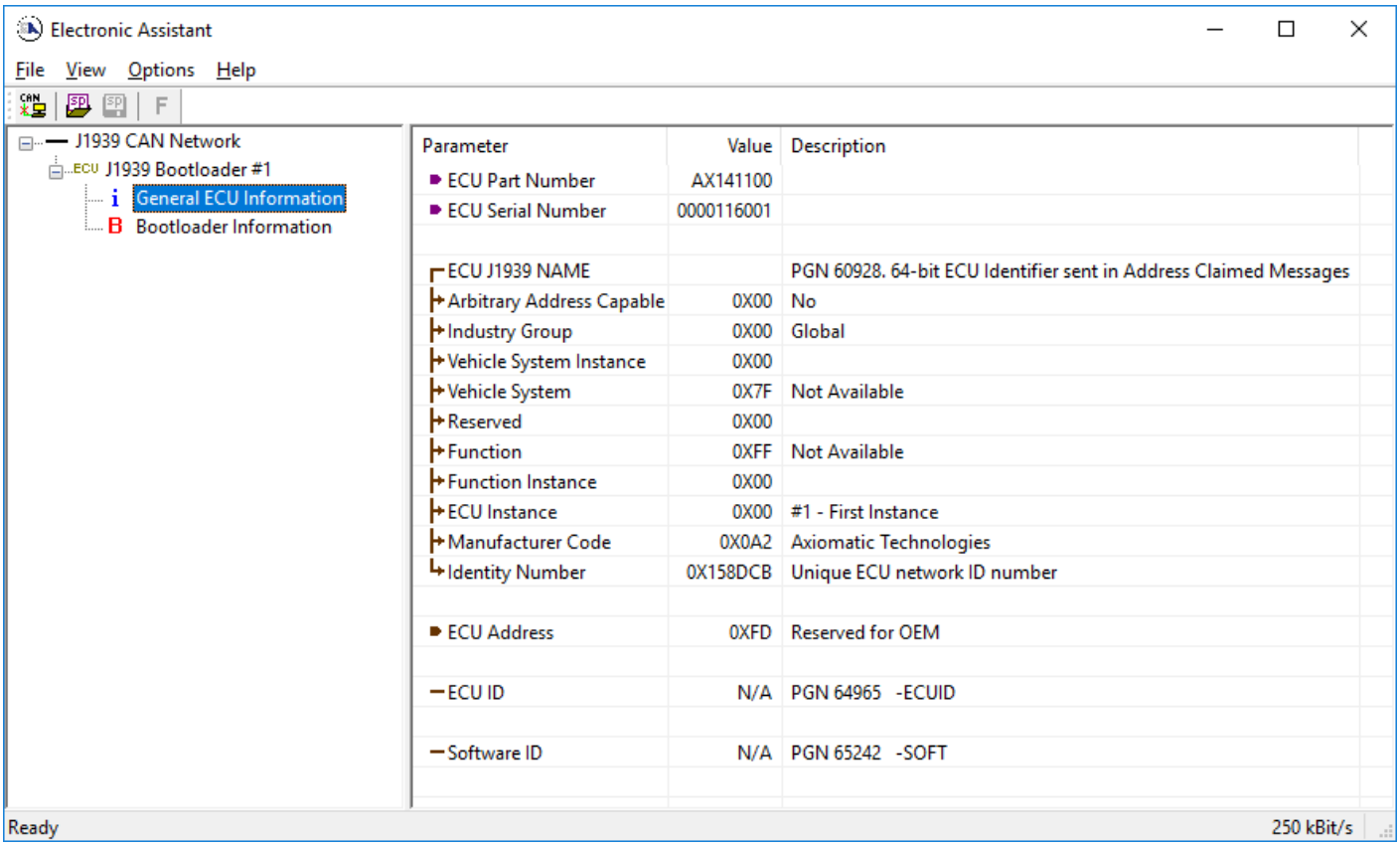

3. When the prompt box asks if you want to reset the ECU, select *Yes*.




3.1 In case the AX141100 device is in Bridge Mode, it cannot be accessed using EA. In this case, it is possible to start the bootloader from CAN2BT Configuration tool. First from the *Misc. configuration* menu, the **Configuration Mode** needs to be set active (on left) and then the **Bootloader** can be started (on right).

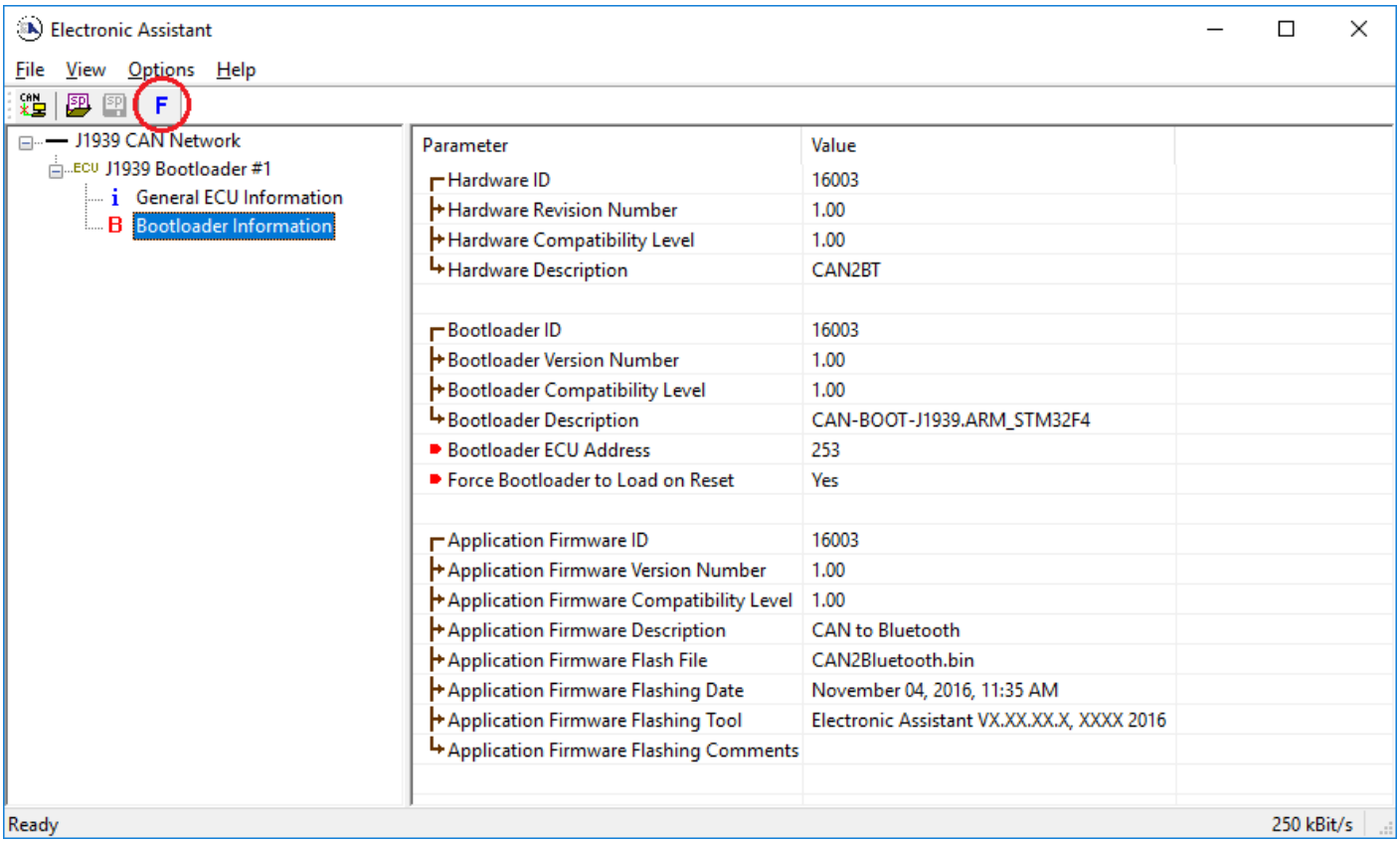

4. Upon reset, the ECU will no longer show up on the J1939 network as an AX141100 but rather as **J1939 Bootloader #1**.




*Note that the bootloader is NOT Arbitrary Address Capable. This means that if you want to have multiple bootloaders running simultaneously (not recommended) you would have to manually change the address for each one before activating the next, or there will be address conflicts. And only one ECU would show up as the bootloader. Once the 'active' bootloader returns to regular functionality, the other ECU(s) would have to be power cycled to re-activate the bootloader feature.*

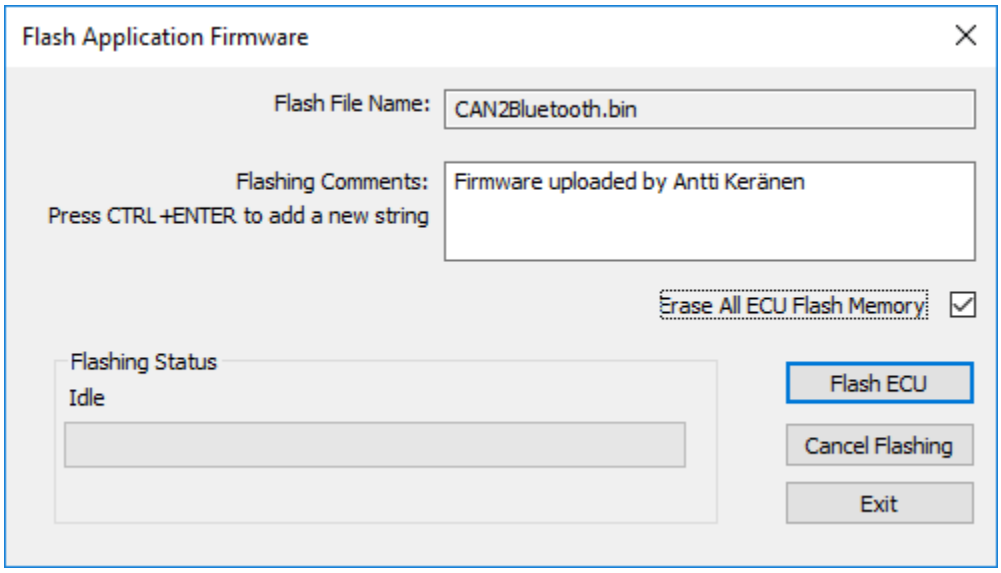
5. When the **Bootloader Information** section is selected, the same information is shown as when it was running the AX141100 firmware, but in this case the **F**lashing feature has been enabled.




6. Select the **F**lashing button and navigate to where you had saved the **CAN2Bluetooth.bin** (or equivalent) file sent from Axiomatic. (Note: only binary (.bin) files can be flashed using the EA tool.)

7. Once the Flash Application Firmware window opens, you can enter comments such as “Firmware upgraded by [Name]” if you so desire. This is not required, and you can leave the field blank if you do not want to use it.

   Note: You do not have to date/time-stamp the file, as this is done automatically by the EA tool when you upload the new firmware.

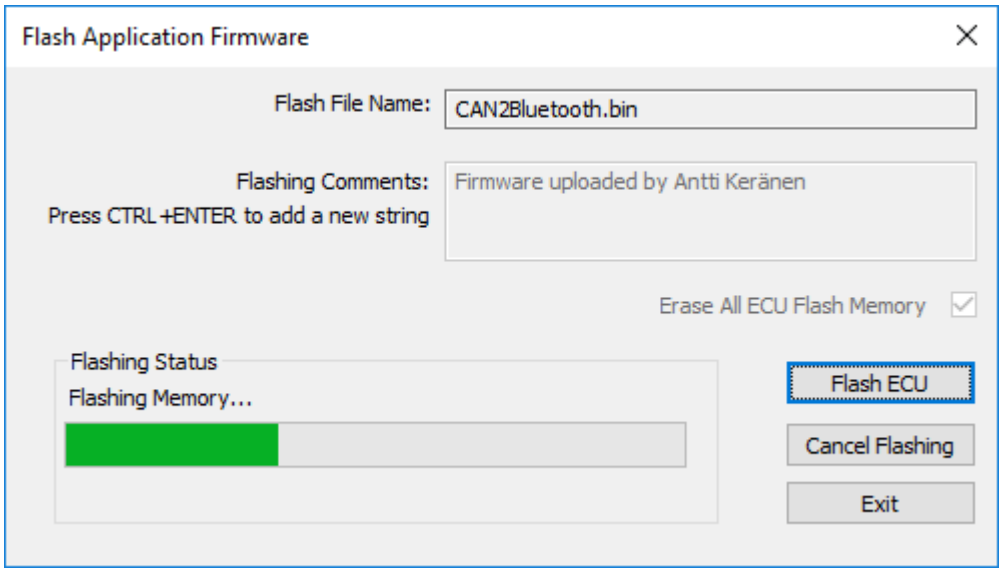



**NOTE:** If the "Erase All ECU Flash Memory" box is checked, all configuration data currently stored in non-volatile flash including PIN codes will be deleted.

**The CAN log data won't be affected by this.**

By leaving this box unchecked, none of the settings will be changed when the new firmware is uploaded, unless it is detected by the new firmware that the old settings are incompatible with the new firmware version.

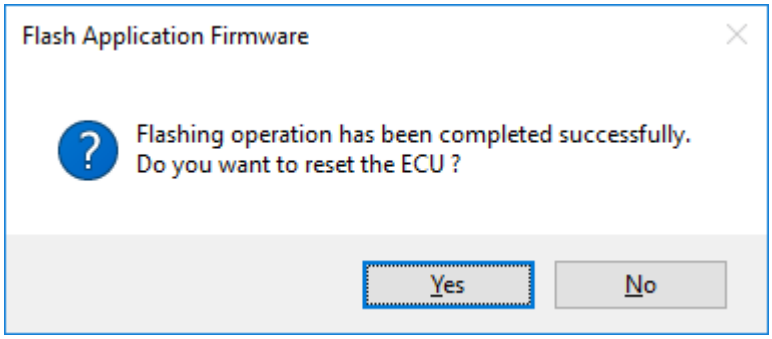
A progress bar will show how much of the firmware has been sent as the upload progresses. The more traffic there is on the J1939 network, the longer the upload process will take.




Once the firmware has finished uploading, a message will pop up indicating the successful operation. If you select to reset the ECU, the new version of the AX141100 application will start

running, and the ECU will be identified as such by EA. Otherwise, the next time the ECU is power-cycled, the AX141100 application will run rather than the bootloader function.

Note, if the settings define the AX141100 to be configured to operate in Brigde Mode, the device will disappear from EA upon reset.




Note: If at any time during the upload the process is interrupted, the data is corrupted (bad checksum) or for any other reason the new firmware is not correct, i.e. bootloader detects that the file loaded was not designed to run on the hardware platform, the bad or corrupted application will not run. Rather, when the ECU is reset or power-cycled the **J1939 Bootloader** will continue to be the default application until valid firmware has been successfully uploaded into the unit.

# APPENDIX A - TECHNICAL SPECIFICATION

## Technical Specifications:

*Specifications are indicative and subject to change. Actual performance will vary depending on the application and operating conditions. Users should satisfy themselves that the product is suitable for use in the intended application. All our products carry a limited warranty against defects in material and workmanship. Please refer to our Warranty, Application Approvals/Limitations and Return Materials Process as described on www.axiomatic.com/service.html.*

| | |
|---|---|
| Power Supply Input - Nominal | 12Vdc or 24Vdc nominal (8…60 VDC power supply range)<br>Load dump protection is provided. |
| Protection | Reverse polarity protection is provided.<br>Overvoltage protection up to 88V is provided. |
| CAN | SAE J1939, 250 kbps Baud Rate |
| Bluetooth | Dual-Mode Bluetooth V4.0 with classic Bluetooth<br>Connection Range*: Up to 50 m (164 ft.)<br>Operating Range*: Up to 150 m (492 ft.) @ 13 dbm (Class 1)<br>Serial Port Profile (SPP)<br>Internal antenna<br>**Range depends on the operating environment and actual results may vary.* |
| Microprocessor | STM32F407VGT7<br>32-bit, 1024 Kbit program flash |
| RTC | Real Time Clock back up power<br>96 hours<br>4 Gbit Flash Memory |
| Quiescent Current | 15 mA @ 24Vdc Typical |
| LED Indicator | User configurable to react to different events or faults |
| Control Logic | User programmable functionality |
| User Interface | CAN2BT Configuration and BT MAP Tool applications are available from Google Play.<br>https://play.google.com/store/apps/details?id=com.axiomatic.can2btconfiguration<br>https://play.google.com/store/apps/details?id=com.axiomatic.btmaptool |
| Network Termination | It is necessary to terminate the network with external termination resistors. The resistors are 120 Ohm, 0.25W minimum, metal film or similar type. They should be placed between CAN_H and CAN_L terminals at both ends of the network. |
| Operating Conditions | -30 to 85 °C (-22 to 185 °F) |
| Enclosure | Molded Enclosure, integral connector<br>Nylon 6/6, 30% glass<br>Ultrasonically welded<br>3.47 x 2.75 x 1.31 inches (88.2 x 70.0 x 33.3 mm)<br>L x W x H including integral connector<br>*Refer to the dimensional drawing in Installation Instructions.* |
| Electrical Connections | Integral TE Deutsch 8 pin receptacle (P/N: DT04-08PA)<br>18 AWG wire is recommended for use with contacts 0462-201-16141.<br>*For Pin Out refer to Installation Instructions.*<br><br>A mating plug kit is available. Ordering P/N: **AX070112 i**s comprised of 1 DT06-08SA, 1 W8S, 8 0462-201-16141, and 3 114017. |
| Protection | IP67 |
| Vibration | Pending<br>MIL-STD-202G, Method 204D test condition C (Sine)<br>and Method 214A, test condition B (Random)<br>10 g peak (Sine)<br>7.68 Grms peak (Random) |
| Shock | Pending<br>MIL- STD-202G, Method 213B, test condition A<br>50g (half sine pulse, 9ms long, 8 per axis) |
| Approvals | CE marking |
| Weight | 0.15 lb. (0.06 kg) |

| | |
|---|---|
| Mounting | Mounting holes are sized for #8 or M4 bolts. The bolt length will be determined by the end-user's mounting plate thickness. The mounting flange of the controller is 0.425 inches (10.8 mm) thick. If the module is mounted without an enclosure, it should be mounted vertically with connectors facing left or right to reduce likelihood of moisture entry.<br><br>The CAN wiring is considered intrinsically safe. The power wires are not considered intrinsically safe and so in hazardous locations, they need to be located in conduit or conduit trays at all times. The module must be mounted in an enclosure in hazardous locations for this purpose.<br><br>No wire or cable harness should exceed 30 meters in length. The power input wiring should be limited to 10 meters.<br><br>All field wiring should be suitable for the operating temperature range.<br><br>Install the unit with appropriate space available for servicing and for adequate wire harness access (6 inches or 15 cm) and strain relief (12 inches or 30 cm). |

## OUR PRODUCTS

Actuator Controls

Battery Chargers

CAN bus Controls, Gateways

CAN/Wifi, CAN/Bluetooth

Current Converters

DC/DC Power Converters

DC Voltage/Current Signal Converters

Engine Temperature Scanners

Ethernet/CAN Converters

Fan Drive Controllers

Hydraulic Valve Controllers

I/O Controls

LVDT Simulators

Machine Controls

Motor Controls

PID Controls

Position Sensors, Angle Measurement Inclinometers

Power Supplies

PWM Signal Converters/Isolators

Resolver Signal Conditioners

Service Tools

Signal Conditioners

Strain Gauge CAN Controls

Surge Suppressors

## OUR COMPANY

Axiomatic provides electronic machine controls, components, and systems to the off-highway, commercial vehicle, electric vehicle, power generator set, material handling, renewable energy and industrial OEM markets.

We provide efficient, innovative solutions that focus on adding value for our customers.

We emphasize service and partnership with our customers, suppliers, and employees to build long term relationships and mutual trust.

## QUALITY DESIGN AND MANUFACTURING

Axiomatic is an ISO 9001:2008 registered facility.

## SERVICE

All products to be returned to Axiomatic require a Return Materials Authorization Number (RMA#).

Please provide the following information when requesting an RMA number:

- Serial number, part number
- Axiomatic invoice number and date
- Hours of operation, description of problem
- Wiring set up diagram, application
- Other comments as needed

When preparing the return shipping paperwork, please note the following. The commercial invoice for customs (and packing slip) should state the harmonized international HS (tariff code), valuation and return goods terminology, as shown in italics below. The value of the units on the commercial invoice should be identical to their purchase price.

*Goods Made In Canada (or Finland)*
*Returned Goods for Warranty Evaluation, HS: 9813.00*
*Valuation Identical Goods*
*Axiomatic RMA#*

## WARRANTY, APPLICATION APPROVALS/LIMITATIONS

Axiomatic Technologies Corporation reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. Users should satisfy themselves that the product is suitable for use in the intended application. All our products carry a limited warranty against defects in material and workmanship. Please refer to our Warranty, Application Approvals/Limitations and Return Materials Process as described on www.axiomatic.com/service.html.

## CONTACTS

**Axiomatic Technologies Corporation**
5915 Wallace Street
Mississauga, ON
CANADA L4Z 1Z8
TEL: +1 905 602 9270
FAX: +1 905 602 9279
www.axiomatic.com

**Axiomatic Technologies Oy**
Höytämöntie 6
33880 Lempäälä
FINLAND
TEL: +358 3 3595 600
FAX: +358 3 3595 660
www.axiomatic.fi

Copyright 2018