

## **G3 CAN-J1939 configuration guide (v1.4)**

Revision log:

v1.4>update for fw6.4.1 sqlite\_db  
v1.3>add info for hexadecimal format of CANPGN, CANSA  
v1.2>add support for data type INT12, CANRAW  
>change key from CANSPN to CANSA (source address)  
v1.1>add support for data type BITS (replace BOOLEAN)

Filename :	iotasset.txt
Location :	\user

### 1. Introduction

The file 'iotasset.txt' contains the assets configuration that is required by the CAN-J1939 program to acquire data from CAN-J1939 devices. Acquired data is then inserted into database for downstream IoT clients.

### 2. IOT asset 'KEY,VALUE' general format

Each IOT asset is defined by using a BLOCK of 'key, value' pairs (CSV format). There are four CAN bus key names that must be present for each IOT asset.

<b>CAN bus KEYS</b>	<b>Description</b>
<b>TYPE</b>	Define the type of CAN bus communication
<b>CANPGN</b>	Define the Parameter Group Number (PGN)
<b>CANSA</b>	Define the Source Address (SA)
<b>CANDATA</b>	Define the J1939 raw data parsing and data type conversion

Each asset block must include the pair 'Key, Field\_name' for database purpose. Backslash (\) and double quote mark (") char cannot be used. Comments can be inserted by using the hash (#) sign.

To ease parsing of different types of assets, the asset blocks need to be located between the start and end of block markers.

<b>CAN bus BLOCK MARKER</b>	<b>Description</b>
CAN_START	Define the start of CAN bus assets
CAN_STOP	Define the end(stop) of CAN bus assets

### 3. IOT asset 'KEY,VALUE' setup information

#### **TYPE**, m

Argument	Value	Description
m	J1939	CAN-J1939 protocol

#### **CANPGN**, n

Argument	Format	Value	Description
n	4 digit hex	eg. FEEE/F004 <sup>#1</sup>	n = Parameter Group Number

#1 Refer to your CAN device manual for supported PGN lists.

#### **CANSA**, s

Argument	Format	Value	Description
s	2 digit hex	eg. 00/09/F7 <sup>#2</sup>	s = Source Address

#2 Refer to your CAN device for parameter 'CAN Node-ID'.

**CANDATA**, t, u, v [, x, y]

datatype: INTEGER, STRING, FLOAT, CANRAW

**CANDATA**, B.b, c, v [, x, y]

datatype: BITS

Argument	Value	Description
t, u	t=Byte start u=Byte length	Position of starting byte (dec:1-8) Length of byte (dec:1-8)
B.b, c	B.b= (Byte_start). (bit_start) c=bits length	Position of starting Byte.bit (dec:1.1-8.8) Length of bits (dec:1-8) <sup>#3</sup>
v	Data Type	Data type as conversion from CAN bus raw data
x	Multiplier	Value = Value*Multiplier + Adder <sup>#4</sup>
y	Adder	Value = Value*Multiplier + Adder <sup>#4</sup>

#3 Bits parsing can only be applied on single byte of CAN data and not across multiple bytes.

#4 Optional: for Data Type INTEGER & FLOAT, **both** x & y arguments required when applied.

#### 4. Data Type definitions for OBD/CAN bus

##### DATA TYPE BITS

v [Data Type]	c [Data Length (bits)]	Description
BITS	1-8	1-8 bits to unsigned integer <sup>#5</sup>

#5 Binary value parsed will be converted to decimal value, eg  $1110_2$  will be reported as  $14_{10}$ . Bits parsing can only be applied on single byte of CAN data and not across multiple bytes.

##### DATA TYPE INTEGER

v [Data Type]	u [Data Length (bytes)]	Description
UINT8	1	8-bit data to <b>8-bit unsigned integer</b>
SINT8		8-bit data to <b>8-bit signed integer</b>
UINT12HL	2	8-bit data pair to <b>12-bit unsigned integer</b> , big endian
UINT12LH		8-bit data pair to <b>12-bit unsigned integer</b> , little endian
SINT12HL		8-bit data pair to <b>12-bit signed integer</b> , big endian
SINT12LH		8-bit data pair to <b>12-bit signed integer</b> , little endian
UINT16HL	2	8-bit data pair to <b>16-bit unsigned integer</b> , big endian
UINT16LH		8-bit data pair to <b>16-bit unsigned integer</b> , little endian
SINT16HL		8-bit data pair to <b>16-bit signed integer</b> , big endian
SINT16LH		8-bit data pair to <b>16-bit signed integer</b> , little endian
UINT32HLhl	4	8-bit data quad to <b>32-bit unsigned integer</b> , big endian
UINT32hIHL		8-bit data quad to <b>32-bit unsigned integer</b> , Word - little endian, Byte - big endian
UINT32LHlh		8-bit data quad to <b>32-bit unsigned integer</b> , Word - big endian, Byte - little endian
UINT32hlLH		8-bit data quad to <b>32-bit unsigned integer</b> , little endian
SINT32HLhl		8-bit data quad to <b>32-bit signed integer</b> , big endian
SINT32hIHL		8-bit data quad to <b>32-bit signed integer</b> , Word - little endian, Byte - big endian
SINT32LHlh		8-bit data quad to <b>32-bit signed integer</b> , Word - big endian, Byte - little endian
SINT32hlLH		8-bit data quad to <b>32-bit signed integer</b> , little endian

##### DATA TYPE STRING

v [Data Type]	u [Data Length (bytes)]	Description
STRING8	8	Set of eight 8-bit data to <b>8 ASCII characters</b> (abcdefgh)
STRING8R		Set of eight 8-bit data to <b>8 ASCII characters</b> , reversed (hgfedcba)
STRING4	4	Set of four 8-bit data to <b>4 ASCII characters</b> (abcd)
STRING4R		Set of four 8-bit data to <b>4 ASCII characters</b> , reversed (dcba)

### DATA TYPE FLOAT

v [Data Type]	u [Data Length (bytes)]	Description
FLOAT32ABCD	4	Set of four 8-bit data to IEEE-754 single precision floating point number. Byte orientation=ABCD,DCBA,BADC,CDAB A,B,C,D=canbyte1,canbyte2,canbyte3,canbyte4
FLOAT32DCBA		
FLOAT32BADCD		
FLOAT32CDAB		

### DATA TYPE CANRAW

v [Data Type]	u [Data Length (bytes)]	Description
CANRAW	8	String of 16 hexadecimal char

#### 5. Example for IOT asset configuration

#iotasset example for CAN-J1939 protocol

```

CAN_START                                #start of CAN bus block

TYPE, J1939                                #CAN type=J1939
CANPGN, FEEE                                #PGN=0xFEEE
CANSAs, F7                                  #SA=0xF7
CANDATA, 1, 1, UINT8, 1,-40                #byte start=1, byte length=1, value=value*1-40
Key, EngineTemp

TYPE, J1939
CANPGN, F004                                #PGN=0xF004
CANSAs, F7
CANDATA, 4, 2, UINT16HL, 0.25, 0          #value=value*0.25 + 0
Key, EngineRPM

TYPE, J1939
CANPGN, FE6C                                #PGN=0xFE6C
CANSAs, F7
CANDATA, 7, 2, UINT16HL
Unit, kmh
Key, VehicleSpeed

```

```
TYPE, J1939
CANPGN, F003 #PGN=0xF003
CANSA, F7
CANDATA, 6.3, 2, BITS #Byte start=6 ,bit start=3, length=2
Key, SpeedLimitStatus

CAN_STOP #end of CAN bus block
```

#### 6. Method to upload 'iotasset.txt' file to G3

-Upload the iotasset.txt file from your computer using the 'Upload iotasset.txt' button in the 'IoT Hardware' tab.

-Put the iotasset.txt file in \user folder of USB drive (with label 'FATBOX'). Plug the USB drive into G3 and click the 'Upload to FATBOX' button in the 'Management' tab.

-Use SCP/Putty console or WinSCP.

<EOF>