

nodeG5 - iotasset.json config guide

CAN-OBD2/QUERY

Firmware version: fw_nodeG5_v2.2
Guide release date: 8MAR2024

Filename :	iotasset.json
Location :	/user

1. Introduction

The file 'iotasset.json' contains the assets configuration that is read by the CAN master program to parse specific data from CAN devices. Acquired data is then inserted into local database for downstream IoT cloud clients.

2. IoTasset name definitions

CAN-OBD2/QUERY IoTasset is defined by JSON objects inside JSON array "CANBUS".

Below is the list of IoTasset names that should be defined in each JSON object.

IoTasset name	Description
Key	Define the data tagname
IOTMODE	Define the data handling mode

Below is the list of CAN-OBD2/QUERY names that should be defined in each JSON object.

CAN-OBD2/QUERY name	Description
CANBUSTYPE	Define the type of CAN bus communication
CANPORT	Define the physical port used for communication
DATATYPE	Define data type to convert from CAN bus raw data

CAN-OBD2 mode	
CANMODEPID	Define the OBD2 Mode and PID
CANIDREQUEST	Define the identifier for request message
CANIDRECEIVE	Define the identifier for response/receive message
CANIDMASK	Define the acceptance mask

CAN-QUERY mode	
CANDATAQUERY	Define the payload data for request message
CANIDREQUEST	Define the identifier for request message
CANIDRECEIVE	Define the identifier for response/receive message
CANIDMASK	Define the acceptance mask

CAN-OBD2/QUERY name	Description
----------------------------	--------------------

Data type = INTEGER, STRING, FLOAT, CANRAW	
BYTESTART	Define the position of starting byte to parse
BYTECOUNT	Define the number of bytes to parse

Data type = BITS	
BITSTART	Define the position of starting byte.bit to parse
BITCOUNT	Define the number of bits to parse

Optional names:

Data type = INTEGER, FLOAT only number=number*multiplier + adder	
SCALEMUL	Define both multiplier & adder when require scaling
SCALEADD	

POLLCYCLE	Define the multiples of Poll Period for any IoTasset
-----------	--

Note: Poll Period is set at web config page 'IoT-Hardware')

3. IoTasset JSON object setup information

“Key”：“value”

Value	Description
string	Unique name for this data value eg temperature, voltage, pressure, rpm

“IOTMODE”：“value”

Value	Description
0	Send to cloud immediately (priority data)
1	Store to local database for local IoT client processing

“CANBUSTYPE”：“value”

Value	Description
OBD2	CAN-OBD2 mode: ISO15765-4 protocol
QUERY	CAN-QUERY mode: CAN request & response messaging

“CANPORT”：“value”

value	Description
canE	On-board CAN 2.0B port
canC	Expansion CAN 2.0B port

Note: refer to nodeG5 user manual for connector pin-out details.

“CANMODEPID”：“value”

Value	Description
0100-0AFF	Mode: 01,02,03...09,0A PID: 00-FF eg value=010C where Mode=01, PID=0C

“CANIDREQUEST”：“value”

value	Description
000-7FF	11-bit identifier (CAN 2.0A)
00000000-1FFFFFFF	29-bit identifier (CAN 2.0B)

“CANIDRECEIVE”：“value”

value	Description
000-7FF	11-bit identifier (CAN 2.0A)
00000000-1FFFFFFF	29-bit identifier (CAN 2.0B)

“CANIDMASK”：“value”

value	Description
000-7FF	11-bit acceptance mask (CAN 2.0A)
00000000-1FFFFFFF	29-bit acceptance mask (CAN 2.0B)

Note: By default mask=7FF if not defined in iotasset.json

“CANDATAQUERY”：“value”

Value	Description
2-16 hexadec	Payload data for request message, length=1 to 8 bytes

“BYTESTART”：“value”

value	Description
1,2,3...8	Position of starting byte (1=left most byte)

“BYTECOUNT”：“value”

value	Description
1,2,3...8	Number of bytes is dependent on data type

“BITSTART”：“value”

value	Description
(byte#1-8).(bit#1-8)	Position of starting byte.bit (1=left most byte/bit)

“BITCOUNT”：“value”

value	Description
1,2,3...8	Number of bits on a single byte

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

4. Data Type definitions for CAN-OBD2

“DATATYPE”：“value”

“BITCOUNT”：“count”

value	count	Description
BITS	1-8	1-8 bits to unsigned integer

Note: Binary value parsed will be converted to decimal value, eg 1110_2 will be reported as 14_{10} .

“DATATYPE”：“value”

“BYTECOUNT”：“count”

value	count	Description
UINT8	1	8-bit data to 8-bit unsigned integer
SINT8		8-bit data to 8-bit signed integer
UINT16HL	2	8-bit data pair to 16-bit unsigned integer
UINT16LH		
SINT16HL		8-bit data pair to 16-bit signed integer
SINT16LH		
UINT32HLhl	4	8-bit data quad to 32-bit unsigned integer
UINT32hIHL		
UINT32LHIh		
UINT32lhLH		
SINT32HLhl		8-bit data quad to 32-bit signed integer
SINT32hIHL		
SINT32LHIh		
SINT32lhLH		

Note: U=unsigned, S=signed, HLhl=big endian, lhLH=little endian
 hIHL=word-little-endian, byte-big-endian
 Lhlh=word-big-endian, byte-little-endian

“DATATYPE”：“value”

“BYTECOUNT”：“count”

value	count	Description
STRING8	8	Set of eight 8-bit data to 8 ASCII characters (abcdefgh)
STRING8R		Set of eight 8-bit data to 8 ASCII characters , reversed (hgfedcba)
STRING4	4	Set of four 8-bit data to 4 ASCII characters (abcd)
STRING4R		Set of four 8-bit data to 4 ASCII characters , reversed (dcba)

“DATATYPE”：“value”
“BYTECOUNT”：“count”

value	count	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		
FLOAT32BADC		
FLOAT32CDAB		

“DATATYPE”：“value”
“BYTECOUNT”：“count”

value	count	Description
CANRAW	8	String of 16 hexadecimal char

5. Example IOTasset JSON configuration

Example for CAN-OB2 mode:

```
{
  "CANBUS":[
    {
      "Key":"OB2EngineCoolantTemp",
      "IOTMODE":"1",
      "CANBUSTYPE":"OB2",
      "CANPORT":"canE",
      "CANIDREQUEST":"7DF",
      "CANIDRECEIVE":"7E8",
      "CANMODEPID":"0105",
      "BYTESTART":"4",
      "BYTECOUNT":"1",
      "DATATYPE":"UINT8",
      "SCALEMUL":"1",
      "SCALEADD":"-40"
    },
    {
      "Key":"OB2EngineRPM",
      "IOTMODE":"1",
      "CANBUSTYPE":"OB2",
      "CANPORT":"canE",
      "CANIDREQUEST":"7DF",
      "CANIDRECEIVE":"7E8",
      "CANMODEPID":"010C",
      "BYTESTART":"4",
      "BYTECOUNT":"2",
      "DATATYPE":"UINT16HL",
      "SCALEMUL":"0.25",
      "SCALEADD":"0"
    },
    {
      "Key":"OB2VehicleSpeed",
      "IOTMODE":"1",
      "CANBUSTYPE":"OB2",
      "CANPORT":"canE",
      "CANIDREQUEST":"7DF",
      "CANIDRECEIVE":"7E8",
      "CANMODEPID":"010D",
      "BYTESTART":"4",
      "BYTECOUNT":"1",
      "DATATYPE":"UINT8"
    }
  ]
}
```

Example for CAN-QUERY mode:

```
{
  "CANBUS":[
    {
      "Key":"LightSensor",
      "IOTMODE":"1",
      "CANBUSTYPE":"QUERY",
      "CANPORT":"canE",
      "CANIDREQUEST":"200",
      "CANIDRECEIVE":"180",
      "BYTESTART":"1",
      "BYTECOUNT":"2",
      "DATATYPE":"UINT16HL",
      "SCALEMUL":"0.5",
      "SCALEADD":"-2",
      "POLLCYCLE":"20"
    },
    {
      "Key":"Power",
      "IOTMODE":"1",
      "CANBUSTYPE":"QUERY",
      "CANPORT":"canE",
      "CANIDREQUEST":"300",
      "CANIDRECEIVE":"280",
      "CANDATAQUERY":"021A05",
      "BYTESTART":"4",
      "BYTECOUNT":"2",
      "DATATYPE":"UINT16HL"
    },
    {
      "Key":"Battery",
      "IOTMODE":"1",
      "CANBUSTYPE":"QUERY",
      "CANPORT":"canE",
      "CANIDREQUEST":"400",
      "CANIDRECEIVE":"380",
      "CANDATAQUERY":"08FF",
      "BYTESTART":"1",
      "BYTECOUNT":"1",
      "DATATYPE":"UINT8",
      "POLLCYCLE":"10"
    },
    {
      "Key":"TurbineSpeed",
      "IOTMODE":"1",
      "CANBUSTYPE":"QUERY",
      "CANPORT":"canE",
      "CANIDREQUEST":"400",
      "CANIDRECEIVE":"7DF",
      "CANDATAQUERY":"02010D5555555555",
      "BYTESTART":"3",
      "BYTECOUNT":"1",
      "DATATYPE":"UINT8"
    }
  ]
}
```


6. Methods to upload 'iotasset.json' file to nodeG5

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

-Put the iotasset.json file in \user folder of USB drive.
Plug the USB drive into any USB-A port and click the 'Upload to nodeG5' button in the 'Management' tab.

-Use SCP/Putty console or WinSCP.

<EOF>