AN407 – Using Axiomatic's Protocol Converter with Aurigo Steering Wheel F202

Introduction

Axiomatic Technologies' AX140200 module can be used in conjunction with the Aurigo CAN Steering Wheel version F202 to convert messages from CANopen® to J1939 without a CANopen® master on the network. The module can be modified by its user to function as desired.

RPDO1

By default, RPDO1 uses COB-ID 200+node-id, and the default node-id of the converter's CANopen® port is 10 (0x0A). Since the RPDOs are active by default, the RPDO1 COB-ID has a value of 0x4000020A. The leading byte with value '4' means that the RPDO is active.
The default mapping for RPDO1 is 1 entry of 0x20000120. Breaking the mapping entry down into its smaller parts shows that:

Data Size = 0x20 = 32 bits
Data location = object 0x2000
Sub-index = 0x01

So RPDO1 is expecting to receive a message with identifier 0x20A from the CANopen® bus, holding one 32-bit data, and it will store the data at object 0x2000_sub-index_1.
For example: users could change the RPDO1 mapping to accept 4 X 16bit data signals on the 0x20A message.

Axiomatic Technologies Oy
Höytämöntie 6
33880 LEMPÄÄLÄ, Finland
Tel. +358 103 375 750
salesfinland@axiomatic.com
www.axiomatic.fi

Axiomatic Technologies Corporation
1445 Courtneypark Dr. E.
Mississauga, ON Canada L5T 2E3
Tel. 1 905 602 9270
sales@axiomatic.com
www.axiomatic.com

RPDO Mapping Parameters

In order to access the RPDO mapping parameters users must first destroy the RPDO. Users can do this by writing a 'C' value as the leading byte of the COB-ID.

- set RPDO1 'COB-ID' = 0x**C**000020A (destroy RPDO1)

SP RPDO 1 COB-ID	0xc000020a
SP RPDO 1 Transmission	255
SP RPDO 1 Inhibit Time	0
SP RPDO 1 Compatibility Entry	0
SP RPDO 1 Event Timer	0

- Set RPDO1 'Number of Mapped Objects' = 0 (this opens access to the mapping objects. Users should do this if they are configuring by CANopen® SDO writes. If they are using the Axiomatic EA software, then this is not needed)
- Set RPDO1 'Mapping Entry #1' = 0x20000110 (this means: 'store a 16-bit data in object 0x2000_sub-index_1')
- Set RPDO1 'Mapping Entry #2' = 0x20000210 (this means: 'store a 16-bit data in object 0x2000_sub-index_2')
- Set RPDO1 'Mapping Entry #3' = 0x20000310 (this means: 'store a 16-bit data in object 0x2000_sub-index_3')
- Set RPDO1 'Mapping Entry #2' = 0x20000410 (this means: 'store a 16-bit data in object 0x2000_sub-index_4')
- Set RPDO1 'Number of Mapped Objects' = 4 (now there are four mapped entries)

SP RPDO 1 Number of Mapped Objects	4
SP RPDO 1 Mapping Entry #1	0x20000110
SP RPDO 1 Mapping Entry #2	0x20000210
SP RPDO 1 Mapping Entry #3	0x20000310
SP RPDO 1 Mapping Entry #4	0x20000410

- Set RPDO1 'COB-ID' = 0x**4**000020A (activate RPDO1)

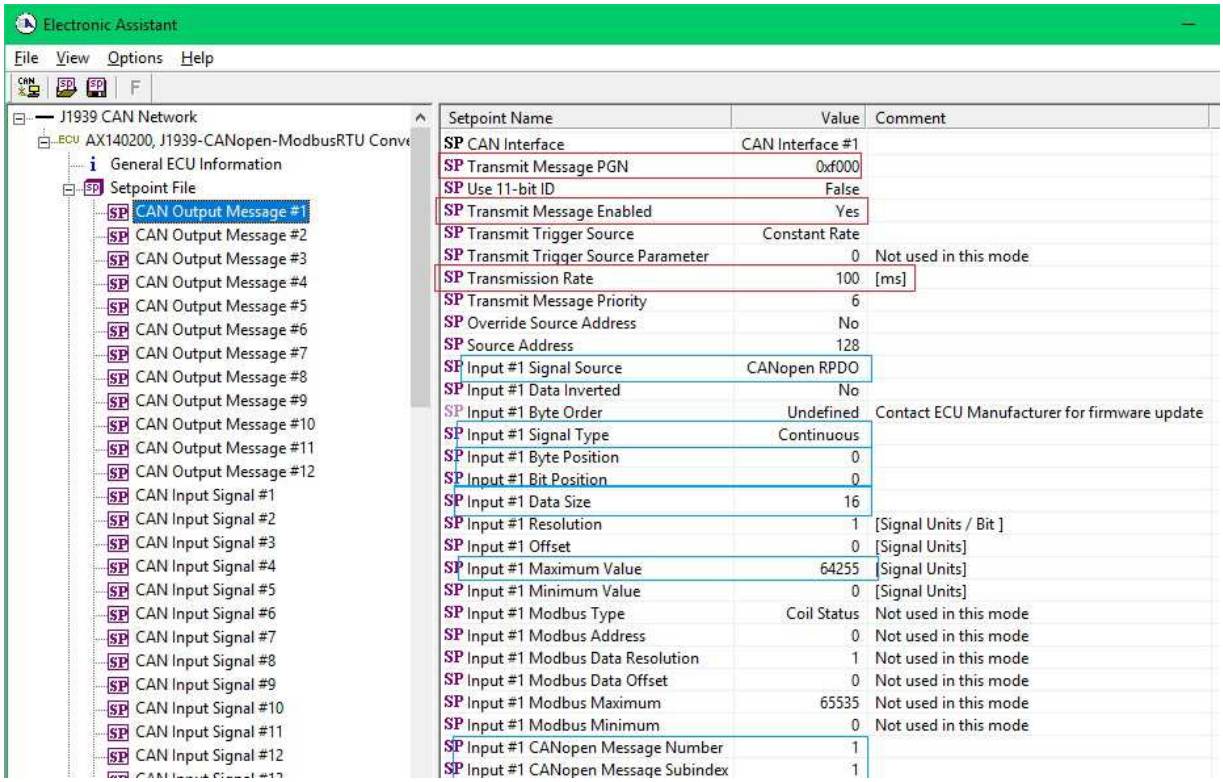
SP RPDO 1 COB-ID	0x4000020a
SP RPDO 1 Transmission	255
SP RPDO 1 Inhibit Time	0
SP RPDO 1 Compatibility Entry	0
SP RPDO 1 Event Timer	0

So now with that done, RPDO1 should be sent on ID 0x20A and contain four 16-bit data signals.

Creating A J1939 Message to Send

Users can create a J1939 message to transmit those four data signals that were received from the CANopen® bus.
Users can do this by editing the CAN Output Message #1 settings. At the start of the settings, users should activate the message to send at 100ms.

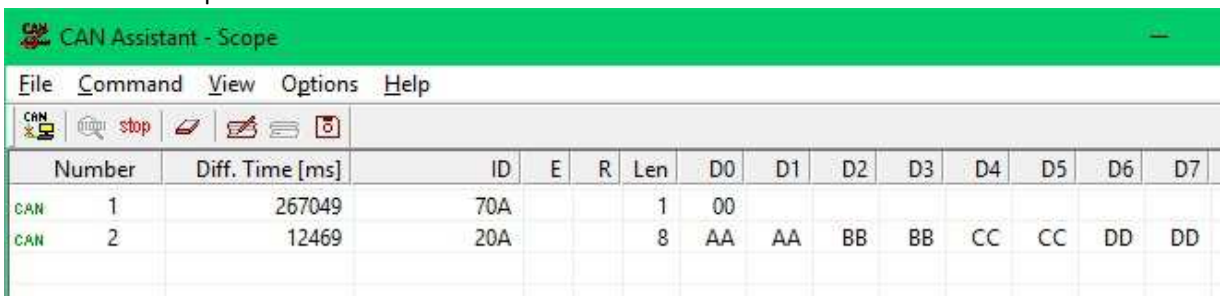
Then, users should assign the first signal to be sent as 16-bit data, starting at byte 0 (so it will use the first two-byte positions), and assign the Data Source as CANopen® RPDO message 1, and sub-index 1. Here is a screen capture from the Axiomatic EA setup:



The byte position is shifted by 2-bytes every time, and the CANopen® RPDO sub index is incremented by 1 every time. So, this process basically puts all four of those RPDO mappings side by side in the J1939 message.

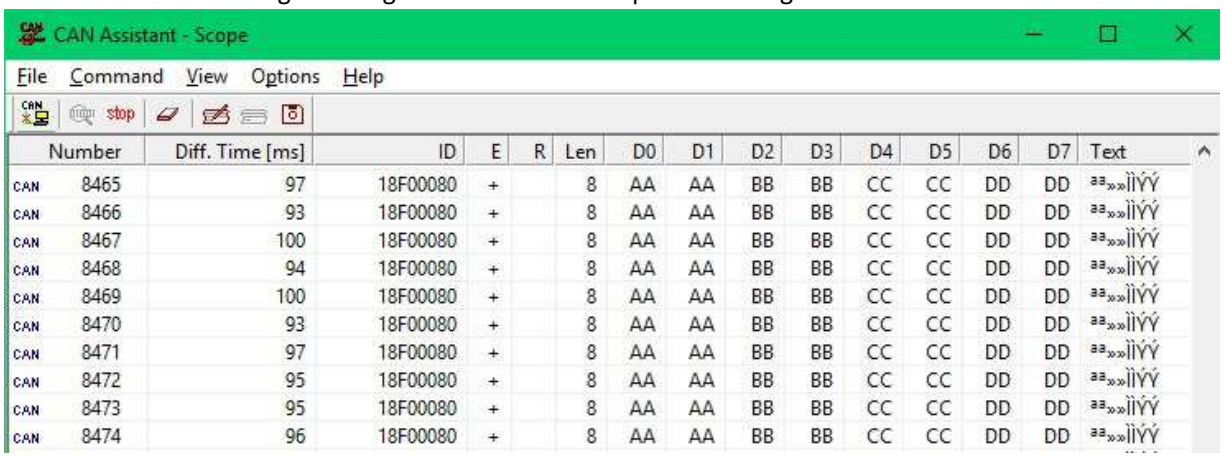
Test Results

In this example, an RPDO with ID = 20A and with four 16-bit signals: AA AA, BB BB, CC CC, DD DD has been sent on the CANopen® network side.



	Number	Diff. Time [ms]	ID	E	R	Len	D0	D1	D2	D3	D4	D5	D6	D7
CAN	1	267049	70A			1	00							
CAN	2	12469	20A			8	AA	AA	BB	BB	CC	CC	DD	DD

Here is the J1939 message holding the four 16-bit data passed through the converter:



	Number	Diff. Time [ms]	ID	E	R	Len	D0	D1	D2	D3	D4	D5	D6	D7	Text
CAN	8465	97	18F00080	+		8	AA	AA	BB	BB	CC	CC	DD	DD	ªª»»ÌÌÝÝ
CAN	8466	93	18F00080	+		8	AA	AA	BB	BB	CC	CC	DD	DD	ªª»»ÌÌÝÝ
CAN	8467	100	18F00080	+		8	AA	AA	BB	BB	CC	CC	DD	DD	ªª»»ÌÌÝÝ
CAN	8468	94	18F00080	+		8	AA	AA	BB	BB	CC	CC	DD	DD	ªª»»ÌÌÝÝ
CAN	8469	100	18F00080	+		8	AA	AA	BB	BB	CC	CC	DD	DD	ªª»»ÌÌÝÝ
CAN	8470	93	18F00080	+		8	AA	AA	BB	BB	CC	CC	DD	DD	ªª»»ÌÌÝÝ
CAN	8471	97	18F00080	+		8	AA	AA	BB	BB	CC	CC	DD	DD	ªª»»ÌÌÝÝ
CAN	8472	95	18F00080	+		8	AA	AA	BB	BB	CC	CC	DD	DD	ªª»»ÌÌÝÝ
CAN	8473	95	18F00080	+		8	AA	AA	BB	BB	CC	CC	DD	DD	ªª»»ÌÌÝÝ
CAN	8474	96	18F00080	+		8	AA	AA	BB	BB	CC	CC	DD	DD	ªª»»ÌÌÝÝ

Connecting Cable

Axiomatic Technologies' AX140200 module employs two male 8-pin M12 (5') cables. These are the straight connector (not right angle). The connectors are not shielded.

The links below are to supplier websites, where users can purchase the specified cables:
TE Connector Cable
Digi Key Cable

Version	Date	Authors	Comments
1.00	September 17, 2019	Sue Thomas / Greg Laronde	Initial Release
1.01	July 6, 2023	Kiril Mojsov	Legacy Updates & Marketing Review; Fixed the TE Connector Cable link

Note: CANopen® is a registered community trademark of CAN in Automation e.V.