AN407

Version 1.00

AN407 – Using Axiomatic Protocol Converter with Aurigo Steering Wheel F202

Introduction

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

RPDO1

By default, RPDO1 is using COB-ID 200+node-id, and the default node-id of the converters, 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:

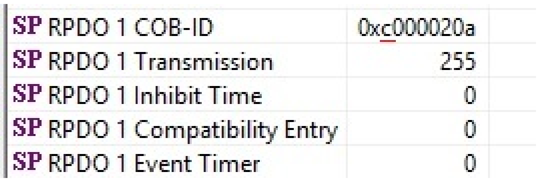
Data Size	= 0x20 = 32 bits
Data location	= object 0x2000
Sub-index	= 0x01

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

RPDO Mapping Parameters

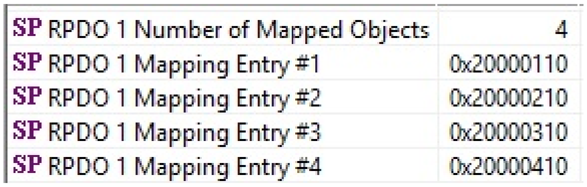
In order to access the RPDO mapping parameters we must first destroy the RPDO. Do this by writing a value ‘C’ as the leading byte of the COB-ID.

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



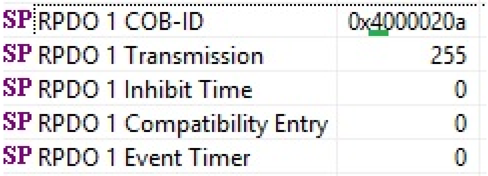
SP RPDO 1 COB-ID	0xc000020a
SP RPDO 1 Transmission	255
SP RPDO 1 Inhibit Time	0
SP RPDO 1 Compatibility Entry	0
SP RPDO 1 Event Timer	0

- Set RPDO1 ‘Number of Mapped Objects’ = 0 (this opens access to the mapping objects, do this if you are configuring by CANopen SDO writes, if using Axiomatic EA software then it 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)



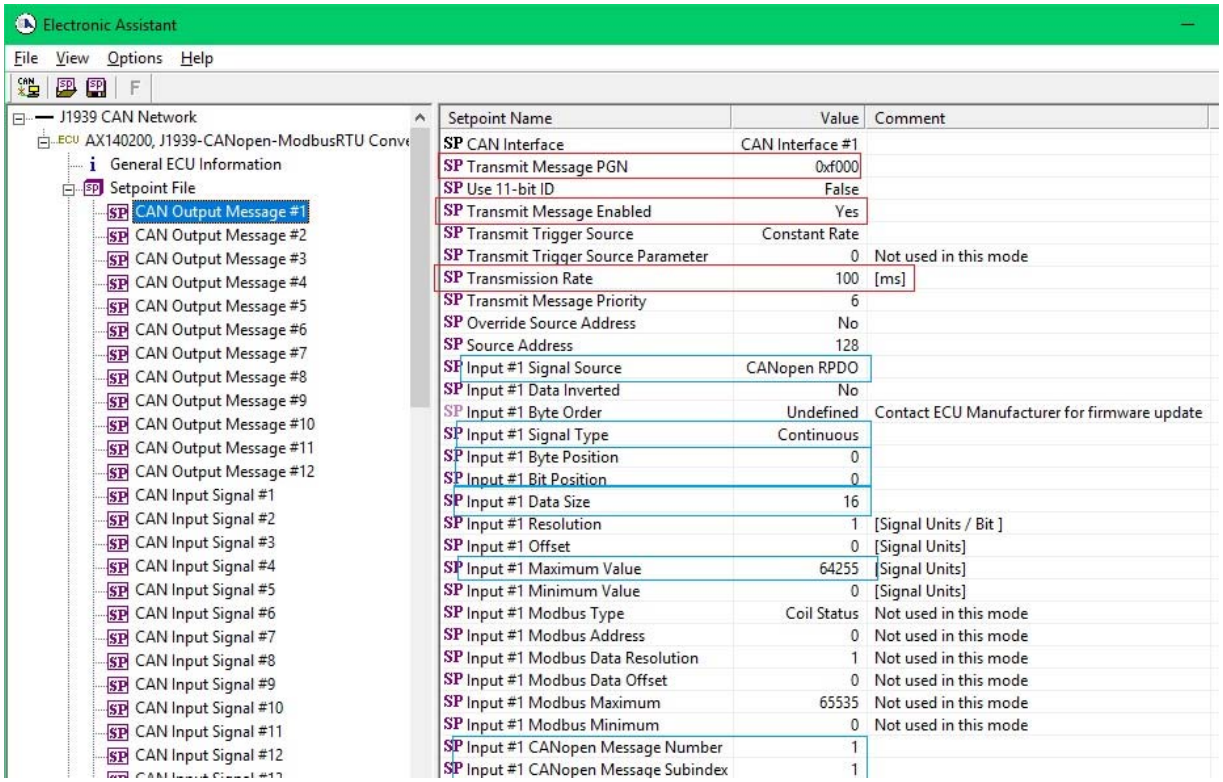
SP RPDO 1 COB-ID	0x4000020a
SP RPDO 1 Transmission	255
SP RPDO 1 Inhibit Time	0
SP RPDO 1 Compatibility Entry	0
SP RPDO 1 Event Timer	0

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

We can create a J1939 message to transmit those four data signals that were received form the CANopen bus.
You do this by editing the CAN Output Message #1 settings. At the start of the settings we activate the message to send at 100ms.

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

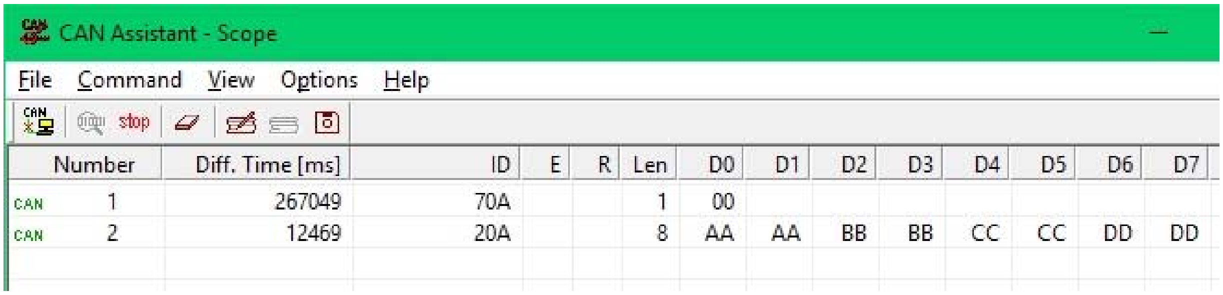


Setpoint Name	Value	Comment
SP CAN Interface	CAN Interface #1	
SP Transmit Message PGN	0xf000	
SP Use 11-bit ID	False	
SP Transmit Message Enabled	Yes	
SP Transmit Trigger Source	Constant Rate	
SP Transmit Trigger Source Parameter	0	Not used in this mode
SP Transmission Rate	100	[ms]
SP Transmit Message Priority	6	
SP Override Source Address	No	
SP Source Address	128	
SP Input #1 Signal Source	CANopen RPDO	
SP Input #1 Data Inverted	No	
SP Input #1 Byte Order	Undefined	Contact ECU Manufacturer for firmware update
SP Input #1 Signal Type	Continuous	
SP Input #1 Byte Position	0	
SP Input #1 Bit Position	0	
SP Input #1 Data Size	16	
SP Input #1 Resolution	1	[Signal Units / Bit]
SP Input #1 Offset	0	[Signal Units]
SP Input #1 Maximum Value	64255	[Signal Units]
SP Input #1 Minimum Value	0	[Signal Units]
SP Input #1 Modbus Type	Coil Status	Not used in this mode
SP Input #1 Modbus Address	0	Not used in this mode
SP Input #1 Modbus Data Resolution	1	Not used in this mode
SP Input #1 Modbus Data Offset	0	Not used in this mode
SP Input #1 Modbus Maximum	65535	Not used in this mode
SP Input #1 Modbus Minimum	0	Not used in this mode
SP Input #1 CANopen Message Number	1	
SP Input #1 CANopen Message Subindex	1	

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

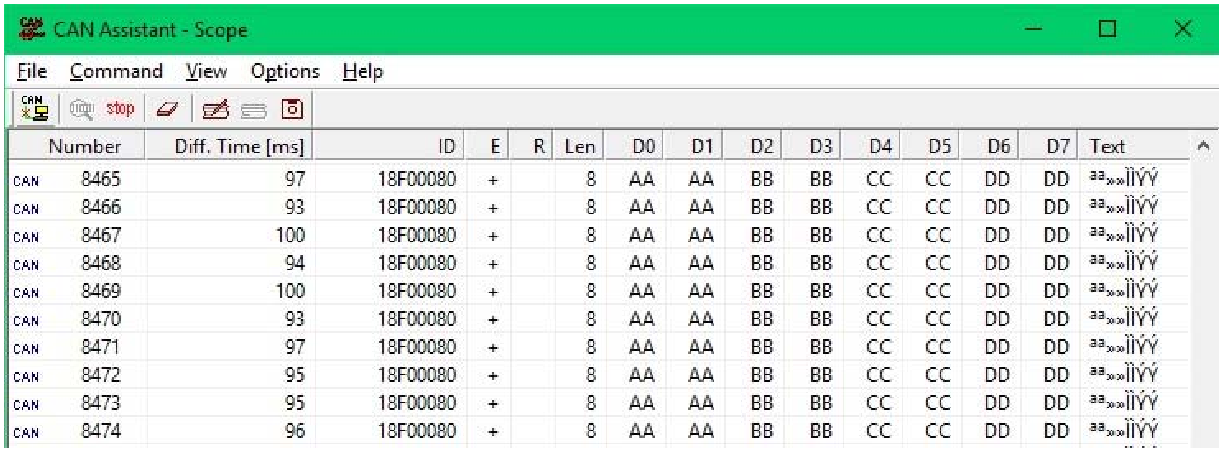
Test Results

On the CANopen network side I will send an RPDO with id = 20A and with four 16-bit signals: AA AA, BB BB, CC CC, DD DD.



	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

Two male 8-pin M12 (5') cables. These are the straight connector (not right angle). Not shielded. The below links are to supplier websites to purchase the specified cable.
TE Connector Cable
Digi Key Cable

Version	Date	Author	Comments
1.00	September 17, 2019	Sue Thomas / Greg Laronde	