RAD-IO2 Series

Isolated Analog, Digital or Temperature Interface to USB and CAN

User's Guide

Version 1.1 - January 27, 2021

INTREPID
CONTROL SYSTEMS
www.intrepidcs.com

1850 Research Drive
Troy, MI 48083 USA
Phone: +1 (586) 731-7950
Fax: +1 (586) 731-2274

automotive engineering tool alliance
www.aeta-rice.com

TABLE OF CONTENTS

FAQ (Frequently Asked Questions (full solutions are in the manual)

- How do you configure the RAD-IO2?
    - *You configure the RAD-IO2 from a PC via the USB port via the JavaScript application or by writing your own code. The JavaScript application runs on Windows, Mac, and Linux. You will need it to change the CAN-ID, sample rates, voltage ranges (if applicable to the module), and any other applicable parameters beyond the defaults.*
- Does RAD-IO2 work directly with VSPY via the PC USB port?
    - *No, VSPY does not support the RAD-IO2 products from the USB port. If you want to use the RAD-IO2 to obtain data directly to a PC you can use the external JavaScript program or write your own software.*
- How do you update the firmware of the RAD-IO2 modules?
    - *Via a PC and the USB port. You will need to go to them ICS Website or Git Hub and download the .EXE program that flashes the RAD-IO2 modules*
    - *You attach the RAD-IO2 USB to the PC and run the program*
- How do you flash the CAN-HUB to update the firmware?
    - *You will need an Intrepid CAN device and VSPY, then use the core-mini function. Detailed instructions are in the manual.*
- How fast is the RAD-IO2?
    - *Individual channels are maximum of 100 samples per second.*
    - *Max aggregate across several chains is 1000 samples per second.*
- How many devices can the CAN-HUB support?
    - *It is dependent upon current draw and each device has a different current draw.*
    - *The CAN-HUB can support one of each device (5).*
- How many devices can the FIRE2 and ION support?
    - *The Fire2 and ION can support 500 mA*
- Starting February 2021, the RAD-IO2 modules come from the factory with the CAN-ID's statically set to 29 bit identifiers, the previous Automode was eliminated. If you chain together 2 modules of the same type without changing the CAN-ID's, you will have CAN conflicts unless you change the settings.
  The Default settings are as follows:
    - CAN-HUB: xx100h,
    - RAD-IO2-TC: xx110h,
    - RAD-IO2-AIN: xx120h
    - RAD-IO2-AOUT: xx130h
    - RAD-IO2-RELAY: xx140h
    - RAD-IO2-DIO: xx150h

1 Introduction and Overview

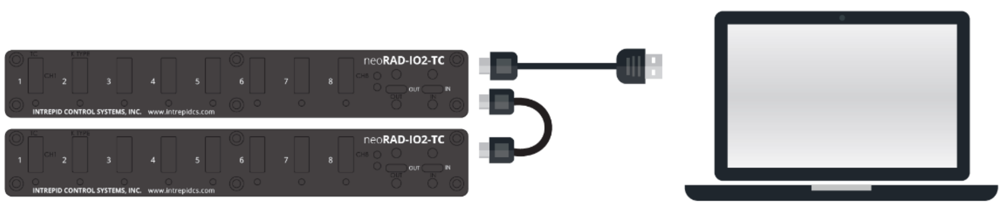
1.1 Introduction

Thank You for purchasing an Intrepid Control Systems RAD-IO2 low-cost measurement and control module. The RAD-IO2 series is a low-cost measurement and control system that connects via USB to a PC or select Intrepid hardware, or it can connect to any CAN tool with the CAN-HUB.

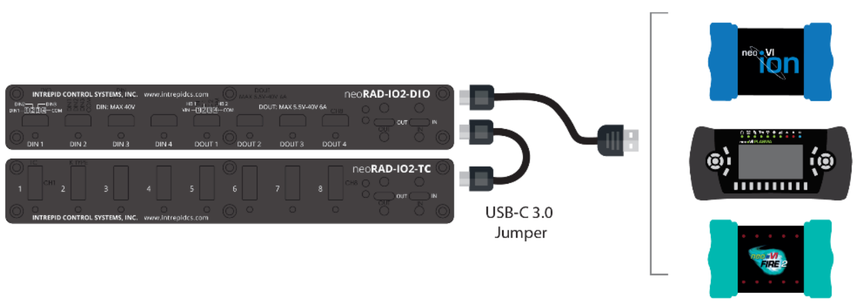
Product Family:

- RAD-IO2-TC: 8 Isolated banks, each with 1 isolated channel of K-type thermocouple
- RAD-IO2-AIN: 8 Isolated banks, 1 channel per bank selectable between a high or low voltage input
- RAD-IO2-AOUT: 8 Isolated analog output banks, each with 3 analog outputs per bank
- RAD-IO2-PWRRLY: 8 Isolated SPDT (single pole double throw) electro-mechanical relays
- RAD-IO2-DIO: 12 digital/analog inputs and 8 digital outputs
- RAD-IO2-CANHUB: CAN FD interface for up to 5 RAD-IO2 devices

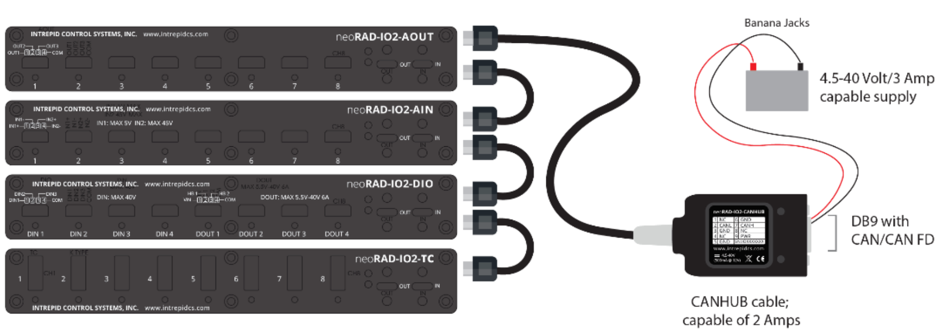
Three methods to obtain data from the RAD-IO2



Use the free JavaScript program to configure, view and log the results or write Python or "C" code.



Connect directly to ION/FIRE2, uses the NeoVI channel



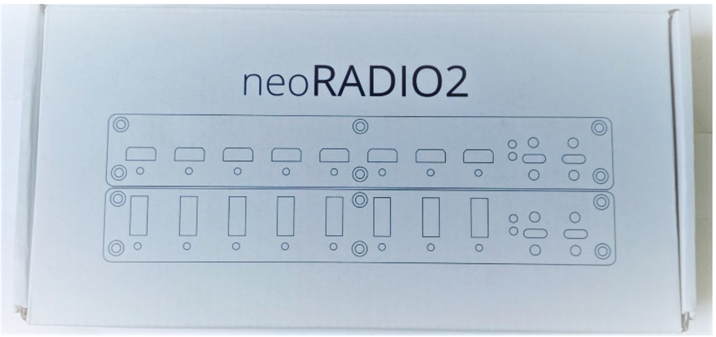
Use the CAN-HUB and convert all the signals to CAN for connection to ANY CAN device

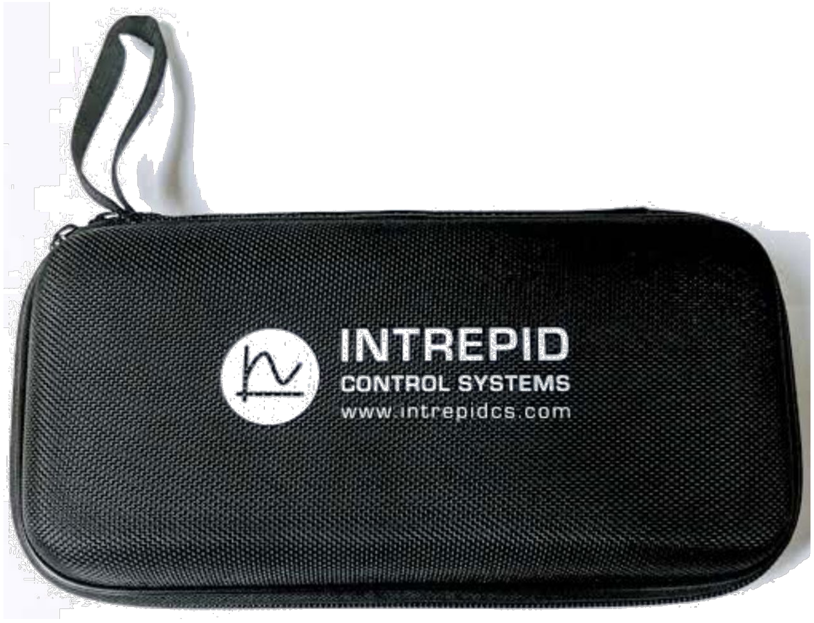
1.2 **Package Contents**

Your RAD-IO2 package includes both hardware and software.

Hardware

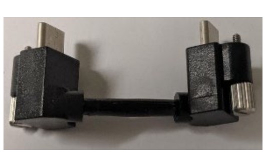
Upon opening the RAD-IO2 box, you should find a CD with the Javascript program for configuring the RAD-IO2 module. Please note that the latest version will be on the ICS GitHub site.



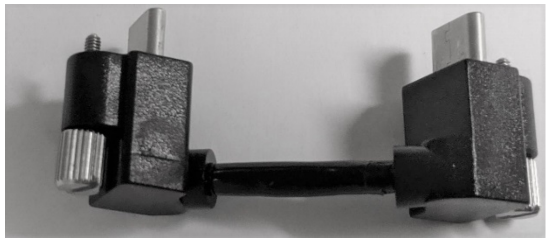


The case will contain 4 items for TC and 5 items for all other modules:

1) The specified module, 2). USB Type-A to Type-C cable 3). attachment plates (to solidly attach two units together), 4). USB-C to USB-C jumper (if you need to connect more than one module together. 5). Phoenix connectors

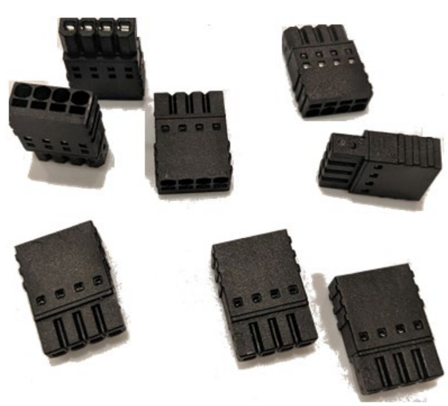


A Jumper is included in a small bag if you wish to connect two or more units together.

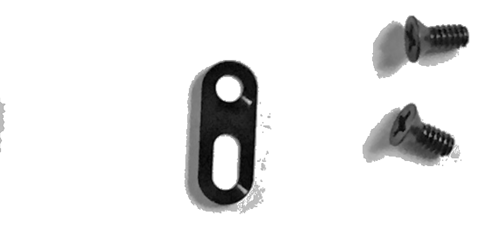


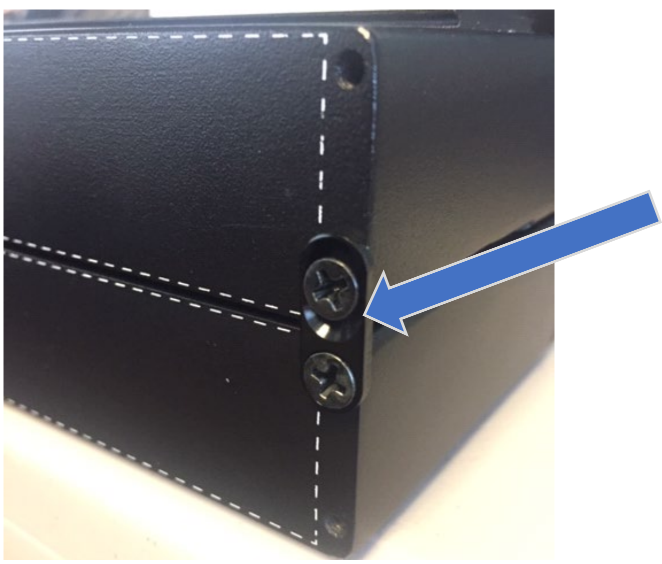
Jumper to connect 2 or more units

Eight (8) Phoenix connectors are included (with every module except for TC):



Attachment plates and fasteners included in a small bag





Attachment plates (to solidly attach two modules together)

1.3 Summary of Key Features

- Sampling rate: Max 1000 sps (samples per second) aggregate across all daisy chained modules, max 100 sps per channel.
- For example, 24 channels would yield 1000 / 24 = ~40 samples per second.
- Required voltage: 5VDC, supplied by USB PC port or powered from the RAD-IO2-CANHUB (4.5VDC-40VDC)
- Current Requirements:
    - Max current per module: RAD-IO2-RELAY: 500 mA (with all relays on), RAD-IO2-AOUT is 250mA plus a max of 5mA per channel used; all other modules are approximately 250 mA.
    - RAD-IO2-CANHUB can supply up to 1.6 Amps via DB9 connector
    - PC USB current supply varies – USB 2 supplies ~500 mA; USB 3 supplies ~900 mA
- 2.5 kV channel to channel and channel to USB isolation
- Direct to neoVI USB feature: neoVI ION, neoVI PLASMA, neoVI FIRE 2
- Includes: Diagnostic and calibration program, APIs and examples for Python, C, C++, and Java

Construction

- Dimensions: 208.8mm x 80.0 mm x 31.9 mm (8.22" x 3.15" x 1.26")
- Weight: 500 g (1.1 lbs)
- Rugged Die Cast Aluminum casing

2 A Tour of RAD-IO2 Hardware

2.2 RAD-IO2-TC Thermocouple Module

- 8 Banks; ONE channel per bank
- Each Bank has it's on CJC and A/D chip 16 BIT Delta Sigma Converter
- "K" Type only
- K-Type range -270C to 1260C; +/- 2C accuracy
- 60Hz and 50Hz common mode rejection > 105dB
- ISOLATED- Since each bank is isolated from another, and there is one thermal couple channel per bank, the effect is that the system has channel to channel isolation- this is without using miniature relays as a multiplexer, which cause a lot of settling time and error.
- Although thermocouples are not as accurate as RTD's, with the high quality of the RAD-IO2-TC, (especially Intrepid calibrates each channel), The RAD-IO2-TC unit is 0.2°C resolution; accuracy better than +/- 1° C is typical.
- Thermocouples are NOT included in production shipments. Go to Omega.com or similar supplier to purchase thermocouples.

2.3 RAD-IO2-AIN Analog Input Module

- There are 8 "Banks", each bank has an one A/D chip 16 BIT Delta Sigma Converter with a Low range and a high range input, you can connect only to the Low or the High not both at the same time.
    - Low range ± 250 mV, ± 1000 mV, ± 5000 mV
    - High range ± 10V, ± 20V, ± 45V
    - **DO NOT exceed the maximum voltage for each range or damage will occur to the unit!**

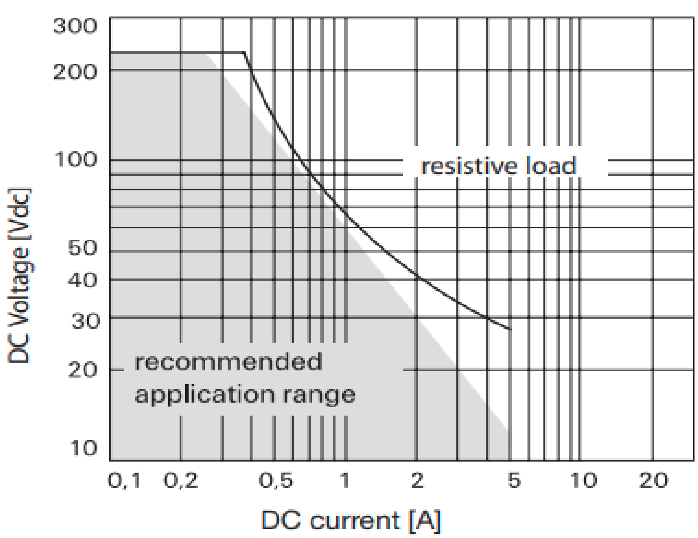
- You can use the AIN module like 8 totally independent "floating" voltmeters without fear of one channel influencing the other.
- Sampling rate: Max 1000 sps (samples per second) aggregate across all daisy chained modules OR max 100 samples per sec channel. For example, 24 channels would yield 1000 / 24 = ~41 samples per second. One channel would yield 100 samples per second.
- RAD-IO2-AIN is great for most static or semi-static measurements but to slow for Electrical Transients, Microphones, Accelerometers.
- The RAD-IO2-AIN is 16 Bit resolution, however it is a Delta-sigma (ΔΣ) analog-to-digital converters (ADCs) so less "jitter" is obtained when slower conversion speeds are used as samples are averaged by the A/D chip internally.
- "Delta-sigma (ΔΣ) analog-to-digital converters (ADCs) are based on the principle of oversampling. The input signal filtered and decimated in the digital domain to yield a conversion result at the respective output data rate.

2.4 RAD-IO2-AOUT- Analog Output Module

- 8 banks of isolated output, each DAC (Bank) has three 0-5V analog outputs and one common ground line per bank, for a total of 24 Analog Outputs
- 16bit DAC (76.3uV per bit)
- 5mA output current per channel, this is limited by the drive chip.
- If all channels were outputting the maximum current, the approximate current draw would be 370 mA.
- Please note, the same 100 samples per second limitation applies when you try to update the analog signal output.

2.5 RAD-IO2-PWRRLY Power Relay module

- 8 isolated Form C relays. Each relay has a normally open and normally closed connection. The two common connections (pin 2 and pin 3) are tied together internally.
- Switching current is 5A for DC voltages up to 220VDC, and 5A for 250VAC, switching power is 60W/62.5VA
- Switching material is AgNi (Silver and Nickel) and gold plated; Initial contact resistance is <50mΩ at 10mA/30mV
- High mechanical shock resistance up to 50g functional
- Electrical endurance
  - (≤30mV/≤10mA) min. 2.5M operations
  - cable load open end min. 2M operations
  - resistive, 220 VDC / 0.27A - 60W min. 500k operations
- Max DC load Breaking capacity



2.5 RAD-IO2-DIO Digital In/Out module

Includes 8 isolated banks with one common ground per bank for a total of 12 inputs and 8 outputs. The first 4 banks are inputs and the second 4 banks are outputs, all banks are isolated from each other. The four isolated input banks inputs have three 0-40V 12bit ADC inputs per bank (12 total), which can be reconfigured during operation as either

- 0-40 VDC analog input
- Digital input with a programmable threshold (in 160mV steps),
- PWM
- Period
- Frequency measurement.

The 4 isolated output banks each have two digital outputs that can be configured as separate digital channels (half bridge) or can be joined as a full H-Bridge output. There are four pairs of outputs for 8 total. Each output can pass 5.5V to 40V at 6A (user supplied).
They can be configured as:

- A simple Digital output,
- A One Shot output
- PWM and frequency programmable output

3 RAD-IO2 module setup and connection to devices

You can configure the modules from your own software or the free JavaScript program.

3.1 Plug the RAD-IO2 directly into the PC or Mac running Windows or Linux.

3.2 Run the Free JAVASCRIPT application on your computer (see instructions in section 4), this will allow you to configure the device and will also allow module control and simple graphing program. If you like, you can program the modules from your own code as well.

3.3 If you are using the CAN HUB:

- 3.3.1 Disconnect the USB cable from the PC and the USB-C port on the RAD-IO2.
- 3.3.2 Connect the USB-C cable of CAN-HUB to the RAD-IO2, and connect the DB9 on the CAN-HUB to a CAN device.
- 3.3.3 Using whatever software you have (VSPY, your own, or a another companies), treat all data like CAN Data. For decoding see examples in section 4.

3.4 FIRE2/ION/VCAN-Industrial/RED2/Fire3/GigaStar

- 3.4.1 Disconnect the USB cable from the PC and connect it to the FIRE2/ION/VCAN-Industrial/RED2/Fire3/GigaStar.
- 3.4.2 Using the NEOVI channel in the Vehicle SPY software treat all data like CAN data for decoding. See examples in section 4.

4 Hardware and Software Setup

4.1 There is a free JAVA program available to setup the RAD-IO2 unit, it allows the user to set critical parameters and "go online" to take measurements.

- 4.1.1 CAN-ID (all modules will come form the factory preset to a static 29-bit CAN-ID.
- 4.1.2 Tag Name for the Bank
- 4.1.3 Sampling frequency- in effect Sample Per Second (SPS); this is for the RAD-IO2-TC, AIN and the Digital Input portion of the DIO.
- 4.1.4 Analog Input LOW or HI input and which voltage range
- 4.1.5 Digital input threshold and input type
- 4.1.6 Digital output type, frequency, and % Duty cycle if PWM is activated (pulse width modulation)
- 4.1.7 Module Calibration for Thermocouple and Analog Input.
- 4.1.8 The ability to create a DBC file to import into VSPY or any other CAN analysis program

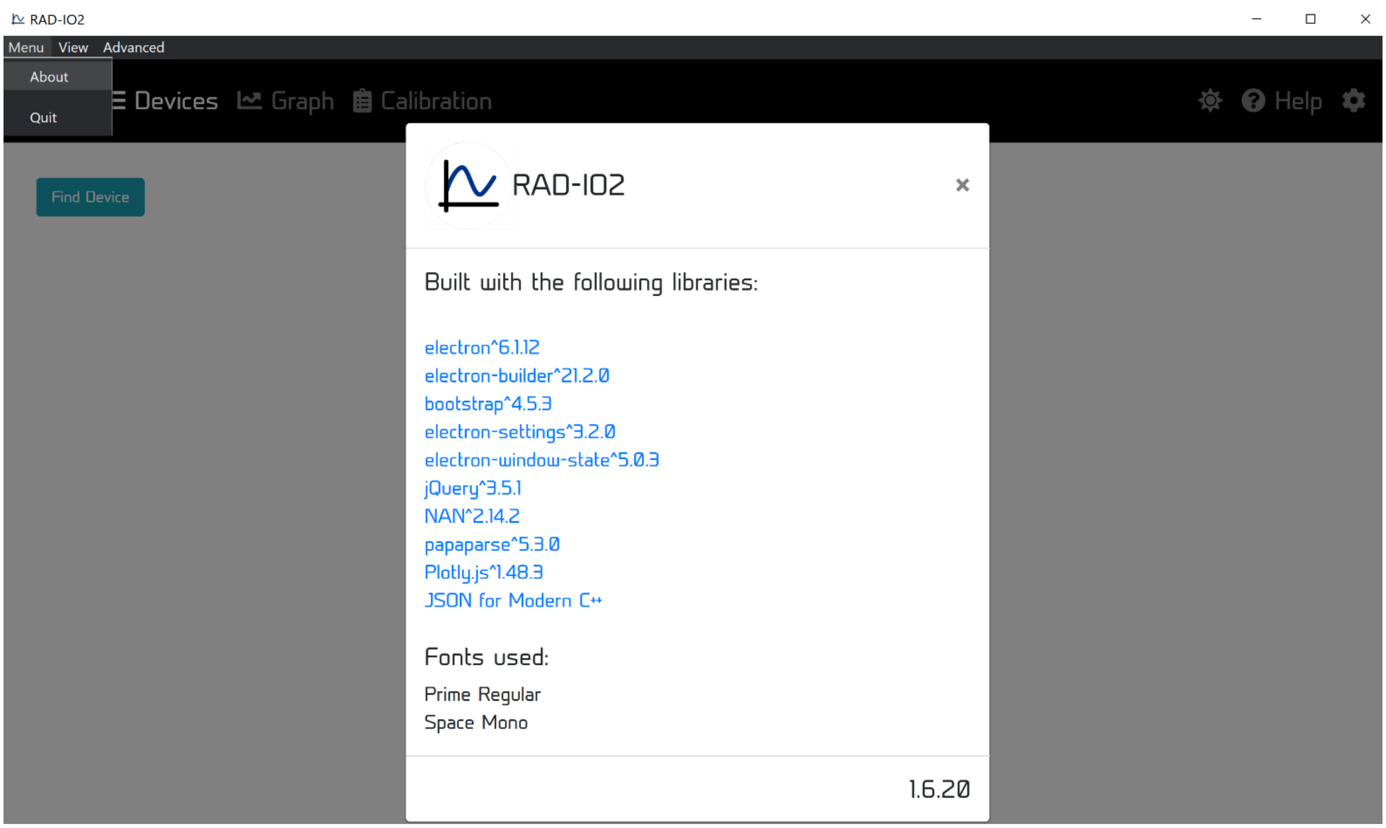
Click the: "Find Device" Button and the devices hooked up should appear.

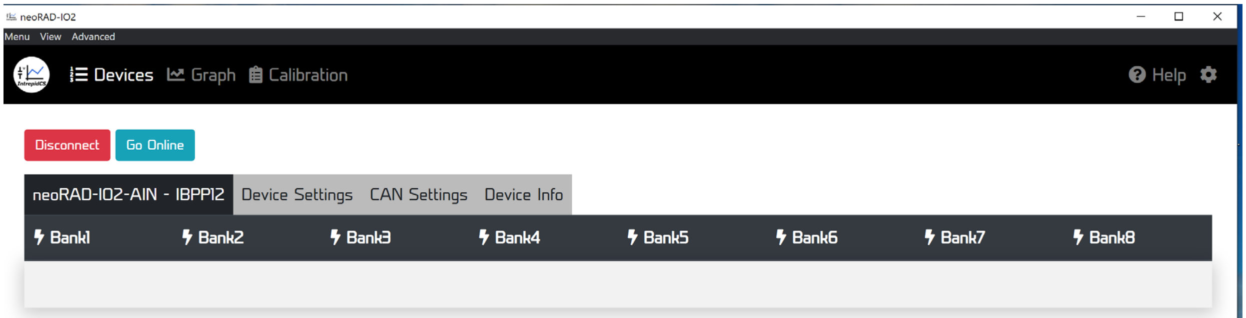
Note: If you have multiple devices of the same type chained together, the default CAN-ID's will be the same, therefore errors will occur until you change the CAN-ID's

The top drop down menu selection is "Menu"; "View" and "Advanced".

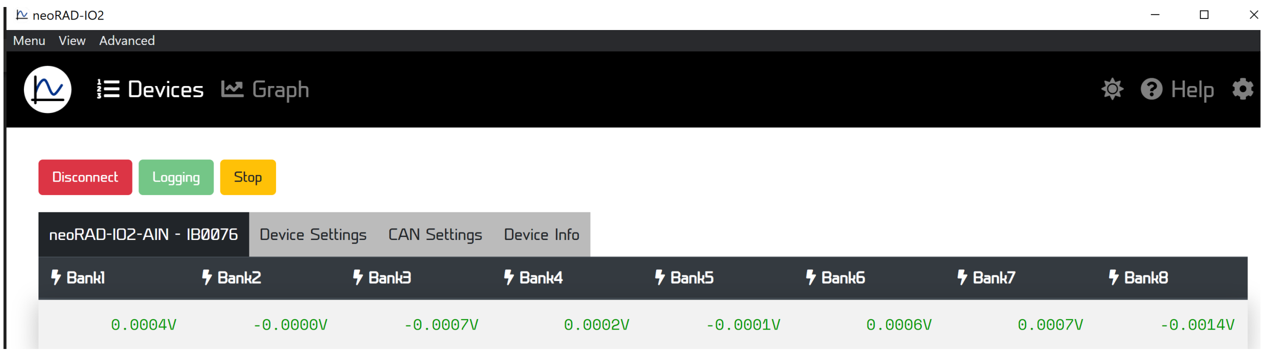
Clicking the "Menu" displays "Quit", which ends the program and "About", which displays a box with the libraries and the version. The source code for this project is on GitHub.

https://github.com/intrepidcs/libneoradio2



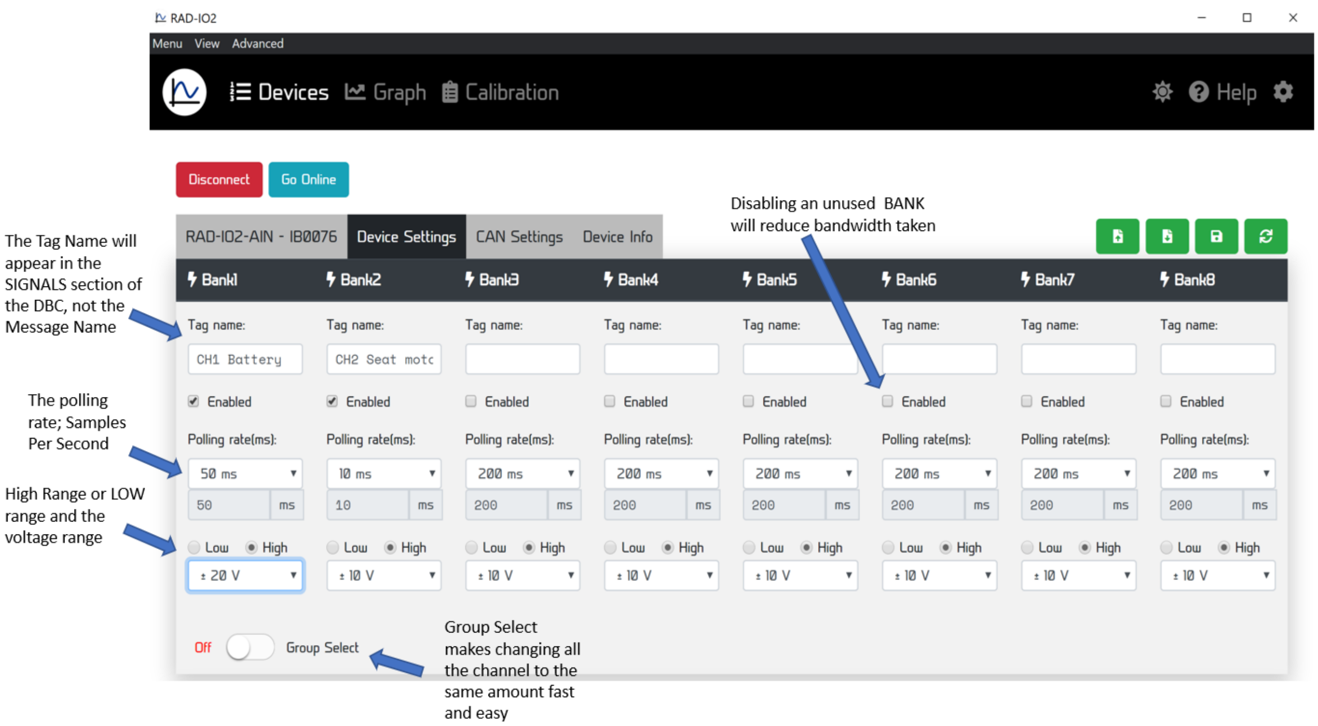


If you "Go Online" the device will start to function, the example below is an Analog Input module.

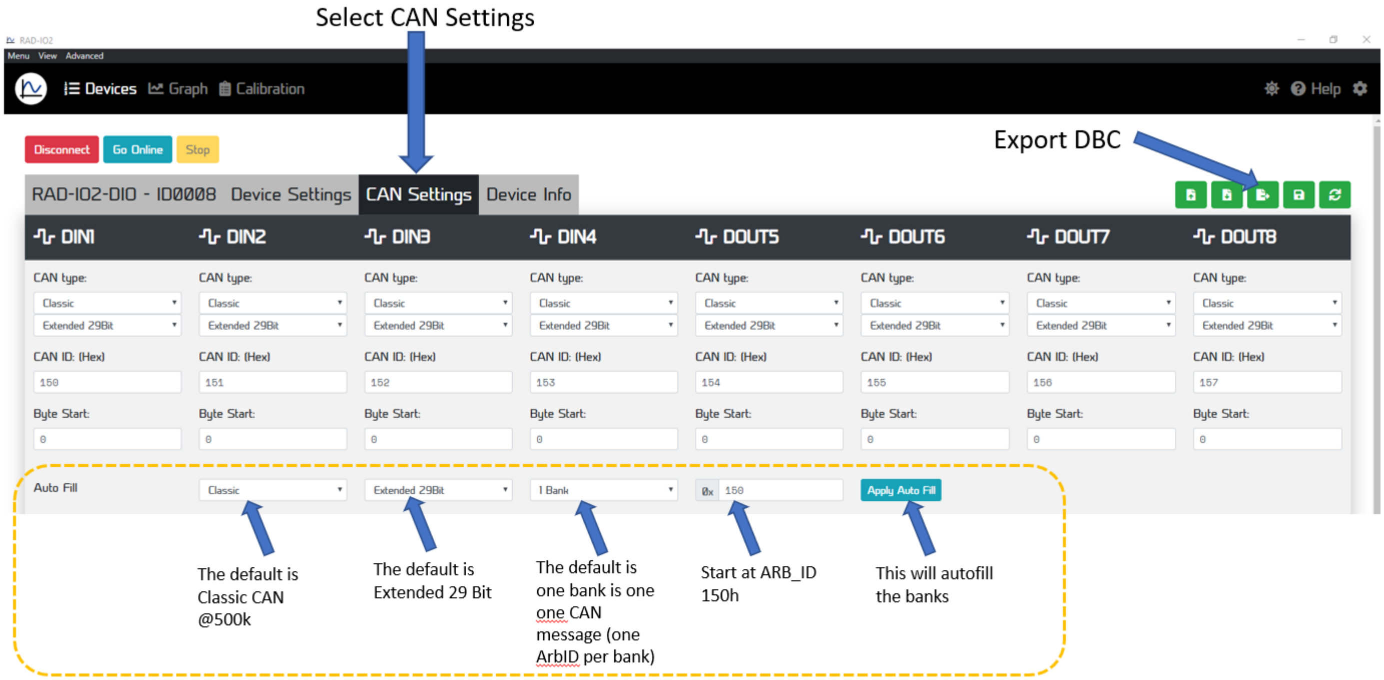


The secondary list is for the "Devices", "Graph", and "Calibration".

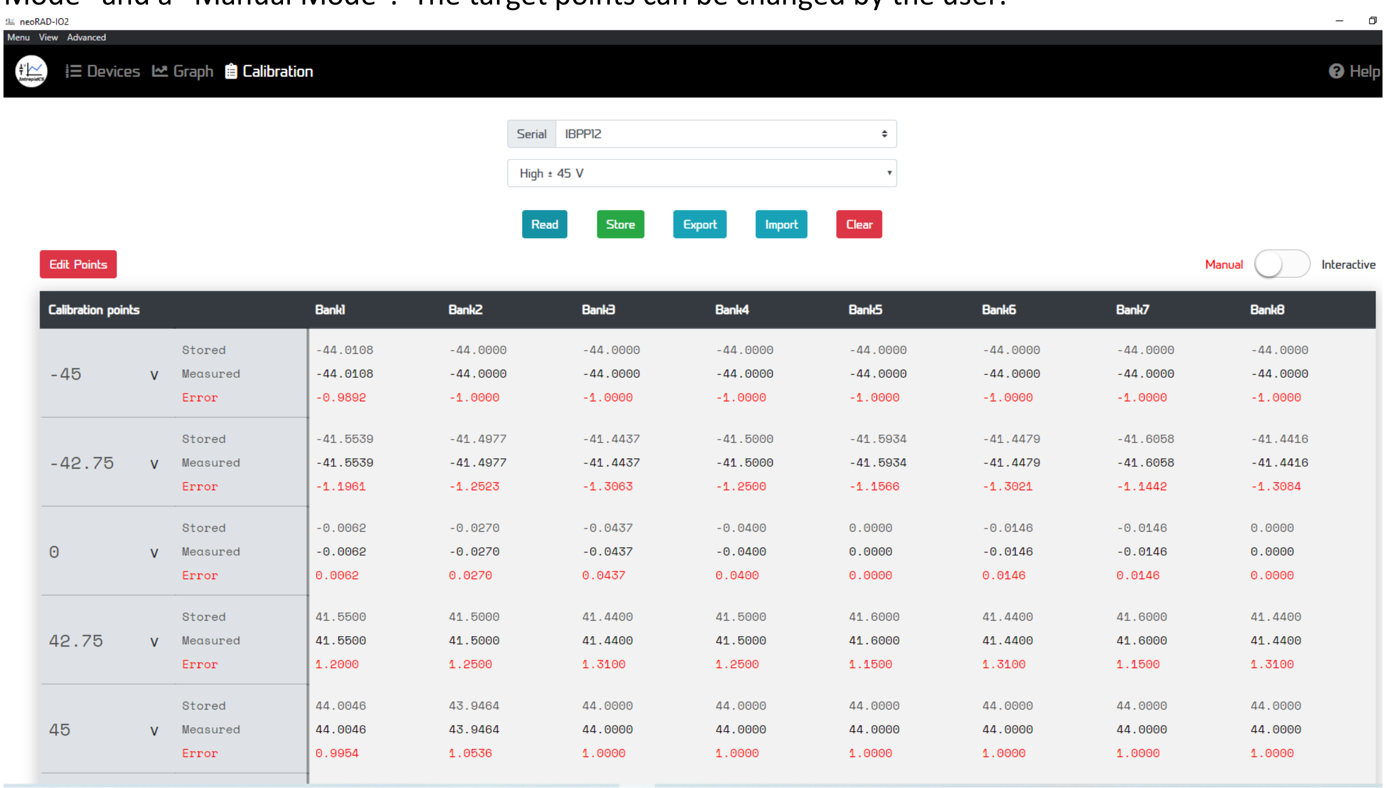
The "Device Settings" TAB



The **CAN Settings** tab allows the user to set CAN-ID and the CAN mode; either CAN or CAN-FD mode and the CAN-ID of each bank.



The CALIBRATION screen is somewhat complex and subject to change, there is currently an "Interactive Mode" and a "Manual Mode". The target points can be changed by the user.



Calibration points			Bank1	Bank2	Bank3	Bank4	Bank5	Bank6	Bank7	Bank8
-45	V	Stored	-44.0108	-44.0000	-44.0000	-44.0000	-44.0000	-44.0000	-44.0000	-44.0000
		Measured	-44.0108	-44.0000	-44.0000	-44.0000	-44.0000	-44.0000	-44.0000	-44.0000
		Error	-0.9892	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
-42.75	V	Stored	-41.5539	-41.4977	-41.4437	-41.5000	-41.5934	-41.4479	-41.6058	-41.4416
		Measured	-41.5539	-41.4977	-41.4437	-41.5000	-41.5934	-41.4479	-41.6058	-41.4416
		Error	-1.1961	-1.2523	-1.3063	-1.2500	-1.1566	-1.3021	-1.1442	-1.3084
0	V	Stored	-0.0062	-0.0270	-0.0437	-0.0400	0.0000	-0.0146	-0.0146	0.0000
		Measured	-0.0062	-0.0270	-0.0437	-0.0400	0.0000	-0.0146	-0.0146	0.0000
		Error	0.0062	0.0270	0.0437	0.0400	0.0000	0.0146	0.0146	0.0000
42.75	V	Stored	41.5500	41.5000	41.4400	41.5000	41.6000	41.4400	41.6000	41.4400
		Measured	41.5500	41.5000	41.4400	41.5000	41.6000	41.4400	41.6000	41.4400
		Error	1.2000	1.2500	1.3100	1.2500	1.1500	1.3100	1.1500	1.3100
45	V	Stored	44.0046	43.9464	44.0000	44.0000	44.0000	44.0000	44.0000	44.0000
		Measured	44.0046	43.9464	44.0000	44.0000	44.0000	44.0000	44.0000	44.0000
		Error	0.9954	1.0536	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

5 CAN-HUB Device Configuration

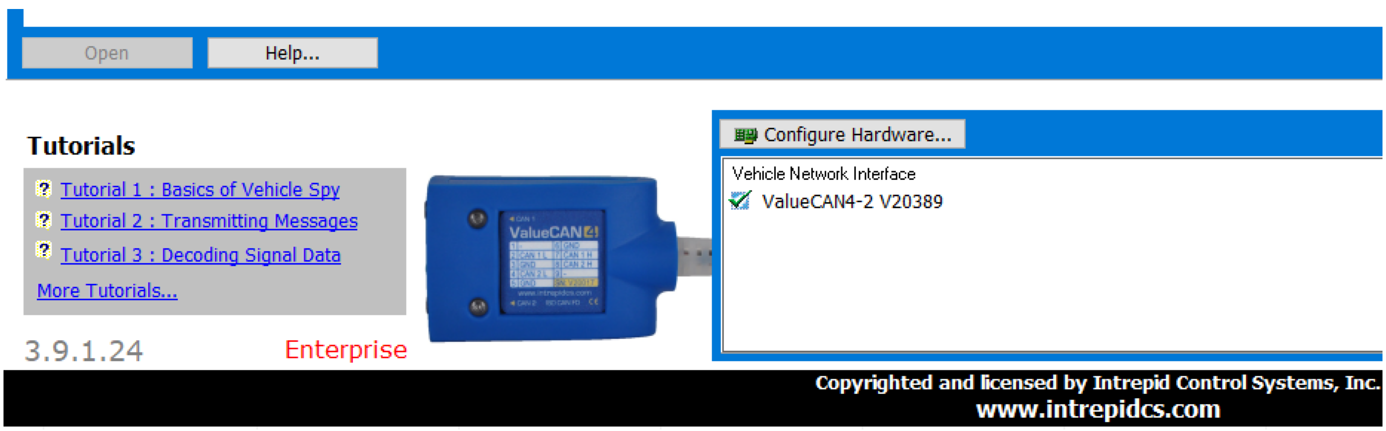
5.1 Update CAN-HUB firmware- IF NEEDED TO UPDATE THE SETTINGS OR FIRMWARE OF THE CAN-HUB PLEASE SEE THIS SECTION.

- 5.1.1 Update the Firmware
- 5.1.2 Update the CAN bus speeds
- 5.1.3 Update the CAN bus termination
- 5.1.4 Update the CAN Sleep ID Address

The CAN-HUB can only be updated VIA VSPY and an Intrepid Control Systems CAN device.

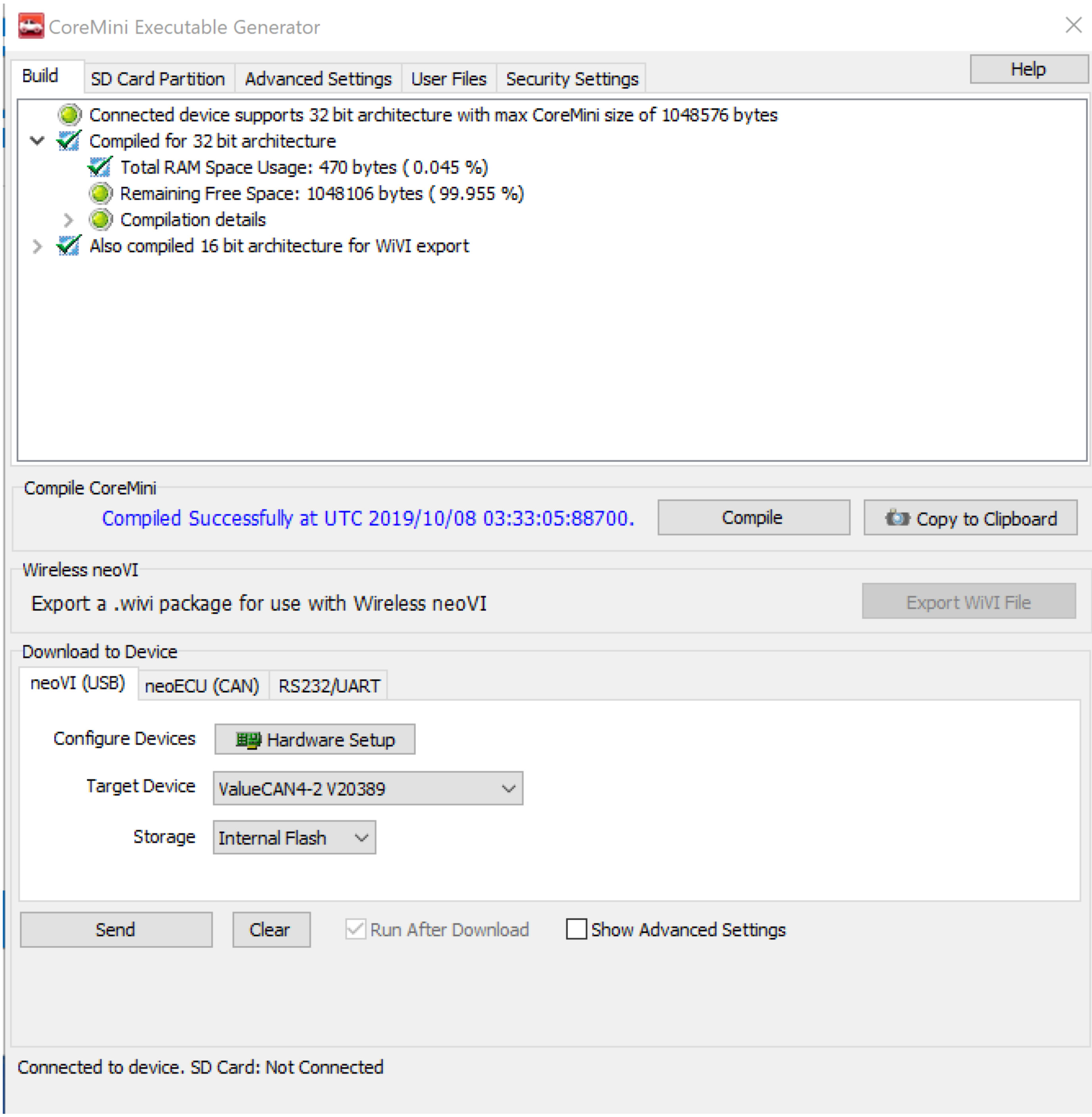
Open Vehicle Spy 3.

You will see the version number in the lower left corner (only 3.9.1.24 or above will work) and the level in red text (i.e. Basic, Pro, or Enterprise). This is the device connected to the computer via the USB cable that will program the RAD-IO2 over the CAN Bus.



Next, select "CoreMini Console" from the Tools pull-down menu and wait for the CoreMini Console window to open.

Hardware Setup: The initial screen will default on the NeoVI(USB) tab.



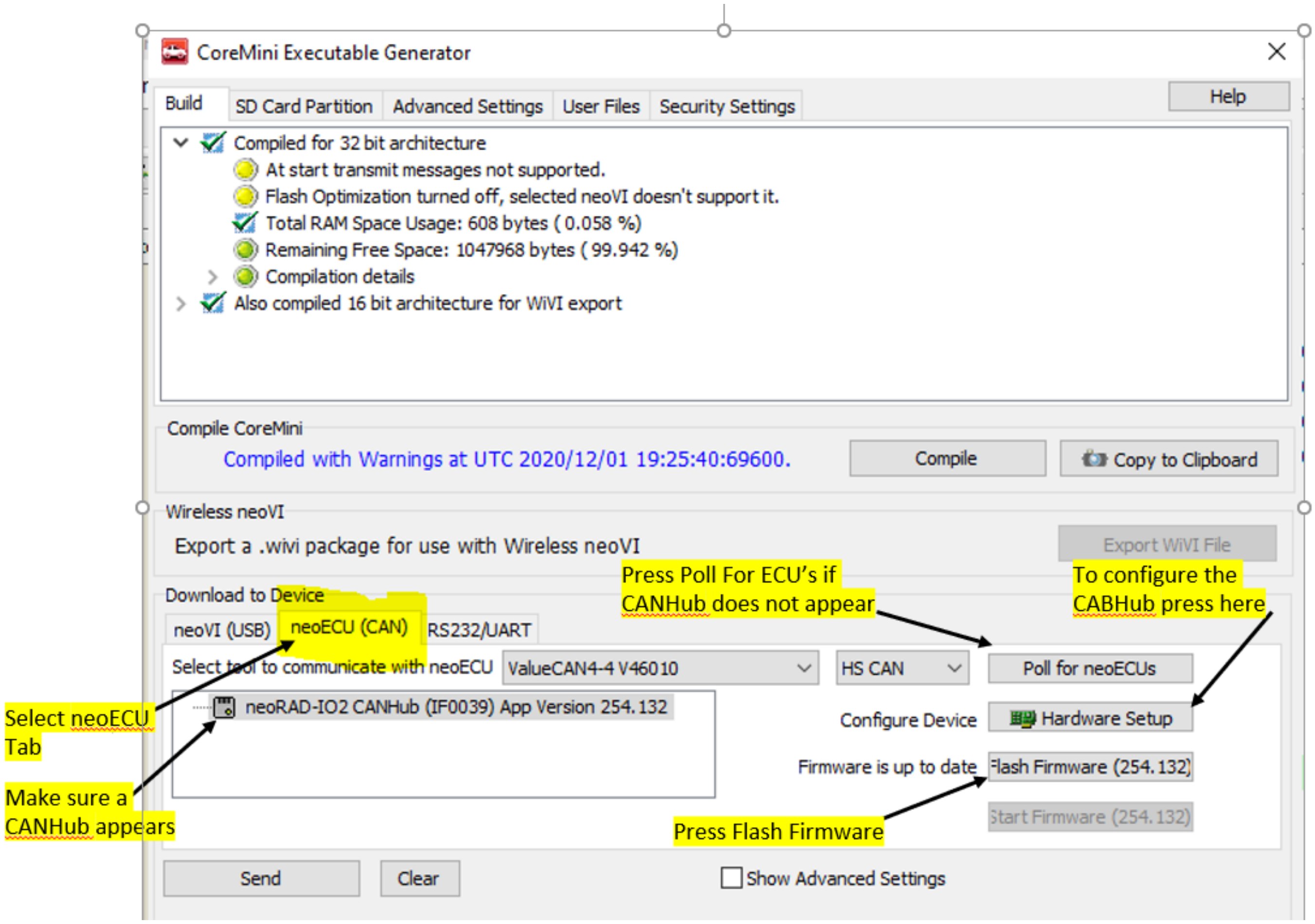
The NeoVI Explorer window should pop-up.

Click on the **neoECU** CAN tab

The **neoRAD-IO2 CAN-HUB** should be listed. If not, click on the "Poll for neoECU's button".

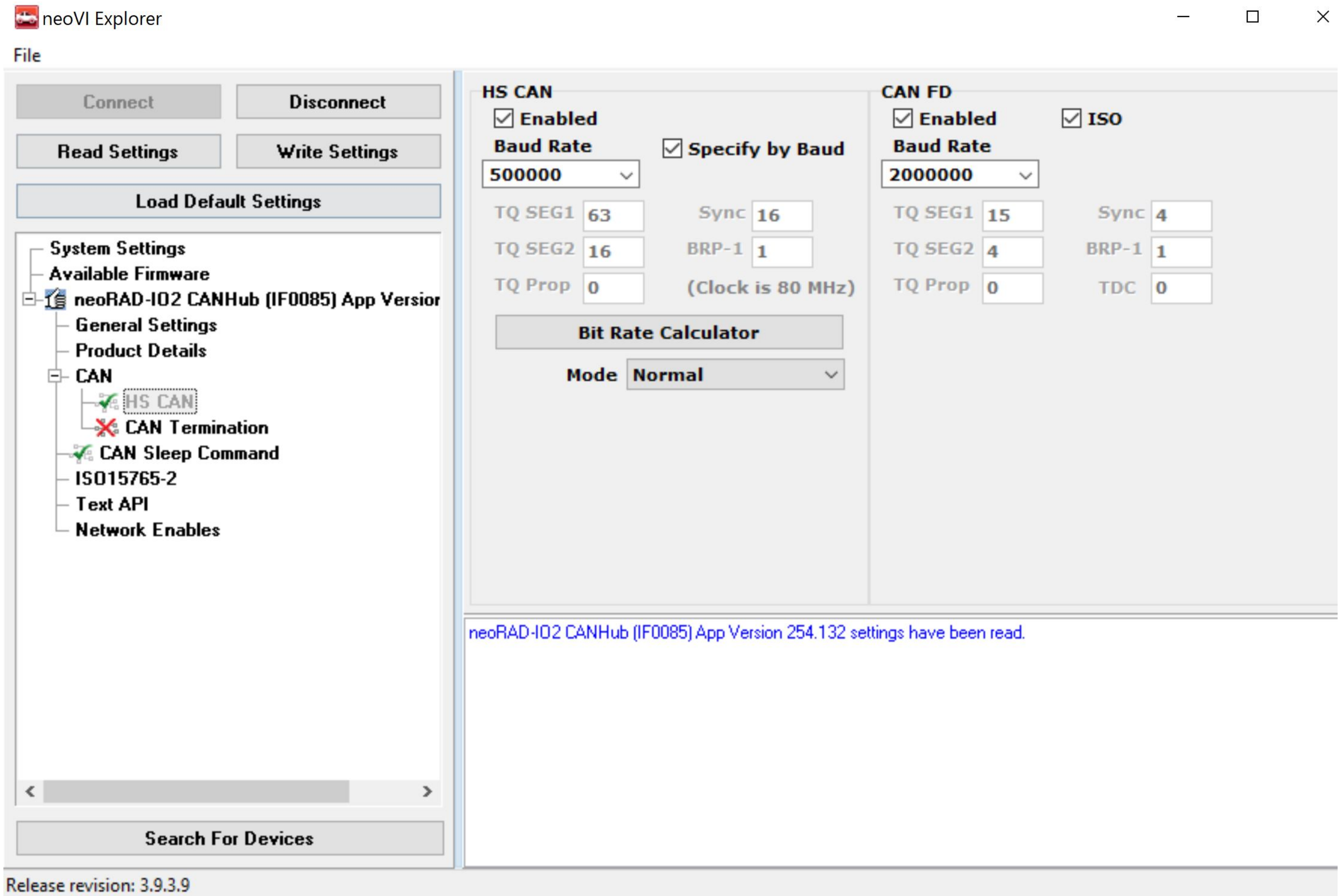
Select the CAN-HUB by clicking on it and click the "Flash Firmware" button to download the latest software.

If you need to configure the CAN-HUB then press Hardware setup. Refer to the screen shot below.



You will then get the NeoVI explorer for the CAN-HUB, which may be familiar to many.

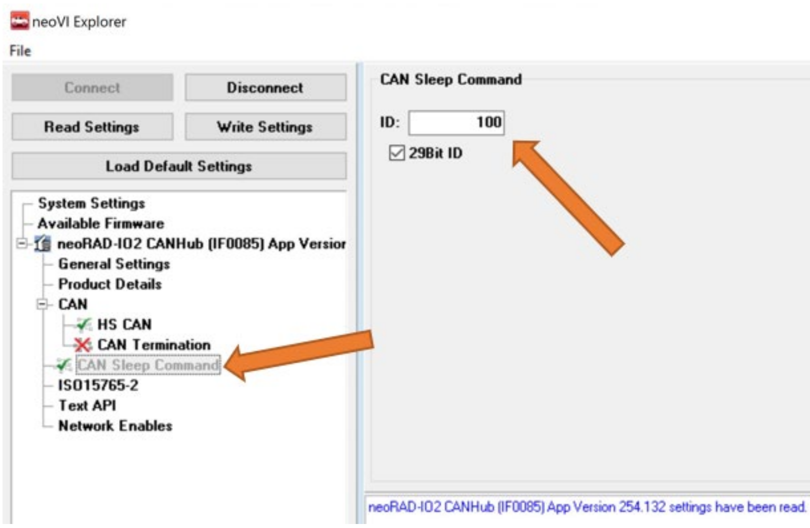
Here you will be able to change the CAN and CAN-FD speed and the termination, and the CAN Sleep command, which is detailed on the next page.



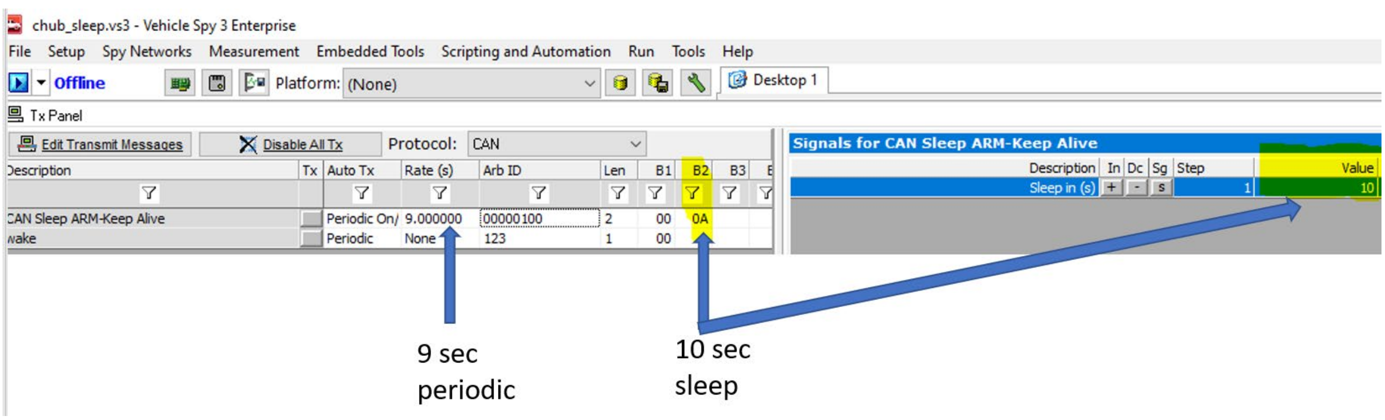
CAN-SLEEP via CAN-HUB

One of the unique features of the CAN-HUB is to stop all outbound CAN communication in a programable amount of time. This will keep a CAN data logger such as an ION or Fire series from waking up and potentially draining the battery. It should be noted that the RAD-IO2 hardware will still be powered from the CAN-HUB, so powering this from a key-on circuit might be best.

The CAN Sleep ARB-ID of the CAN-HUB can be changed in this screen, the default is 100h 29 bit extended ID.



To invoke the CAN sleep action, simply transmit the sleep ARB-ID (in this case 100h is the default) with the second byte filled with the amount of seconds in which to time out. To keep the unit from going to sleep, just keep seeding the message in less time than the expiration as a periodic keep alive message. If the time expires before the keep alive/arm is sent the CAN-HUB will stop sending RAD-IO2 messages. To wake the CAN-HUB back up send it any message and the sleep mode will be cleared until invoked again.



In the above example, the CAN-HUB can be sent a periodic message every 9 seconds with a 10 second expiration time. Therefore it will never go to sleep. If you were to stop the periodic message it will sleep 10 seconds after the last message is received.

Using the RAD-IO2 with CAN-HUB/FIRE2/ION/IND-VCAN

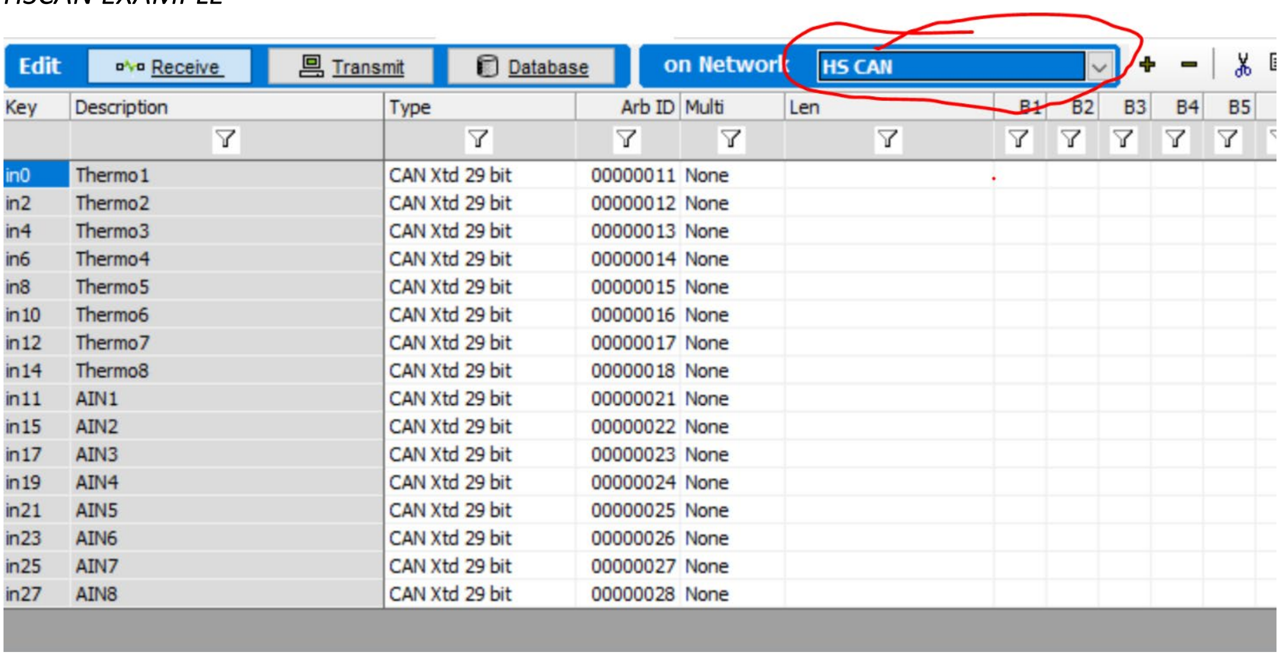
The RAD-IO2 communicates via a UART/USB signal. This can be connected directly to the USB port on a FIRE2, ION or VCAN-Industrial. You can also convert the native UART/USB signal via the CAN-HUB to CAN messages. The Arbitration CAN ID's for the messages can be assigned individually for each channel of the RAD-IO2 module or selected via group. You can use the free JAVA program to accomplish this, or write custom PYTHON code. Both are on GIT-HUB.

6.0 If you are using Vehicle Spy or another program, you can load in the DBC file (generated from the JAVASCRIPT program) with the associated channel. If you do not load in the DBC file, you must configure the receive channels. The example below is for Vehicle Spy; if you are using the CAN-HUB then select the appropriate "HSCAN" channel, if you are attaching to the Fire2, ION or Industrial VCAN, select the "NEOVI" channel.

For INPUT DEVICES such as Thermocouple and Analog Input

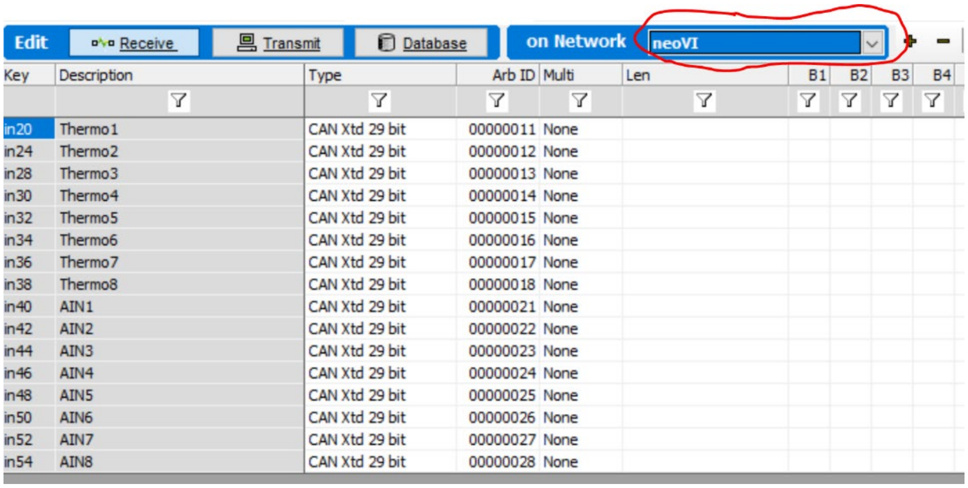
6.1 Create a Receive message from Messages Editor by clicking on "Receive", and then IF you are using CAN-HUB and converting to CAN, select the appropriate HS-CAN channel that you have physically wired to, or connected the DB9 connector on the CAN-HUB output to, you will also need to supply 5-40VDC to the CAN-HUB either via your custom cable or via the cable sold by Intrepid. If you are connecting via USB cable directly to the FIRE2/ION/Plasma or Industrial Value CAN, select the NEOVI channel.

HSCAN EXAMPLE



Key	Description	Type	Arb ID	Multi	Len	B1	B2	B3	B4	B5
in0	Thermo1	CAN Xtd 29 bit	00000011	None						
in2	Thermo2	CAN Xtd 29 bit	00000012	None						
in4	Thermo3	CAN Xtd 29 bit	00000013	None						
in6	Thermo4	CAN Xtd 29 bit	00000014	None						
in8	Thermo5	CAN Xtd 29 bit	00000015	None						
in10	Thermo6	CAN Xtd 29 bit	00000016	None						
in12	Thermo7	CAN Xtd 29 bit	00000017	None						
in14	Thermo8	CAN Xtd 29 bit	00000018	None						
in11	AIN1	CAN Xtd 29 bit	00000021	None						
in15	AIN2	CAN Xtd 29 bit	00000022	None						
in17	AIN3	CAN Xtd 29 bit	00000023	None						
in19	AIN4	CAN Xtd 29 bit	00000024	None						
in21	AIN5	CAN Xtd 29 bit	00000025	None						
in23	AIN6	CAN Xtd 29 bit	00000026	None						
in25	AIN7	CAN Xtd 29 bit	00000027	None						
in27	AIN8	CAN Xtd 29 bit	00000028	None						

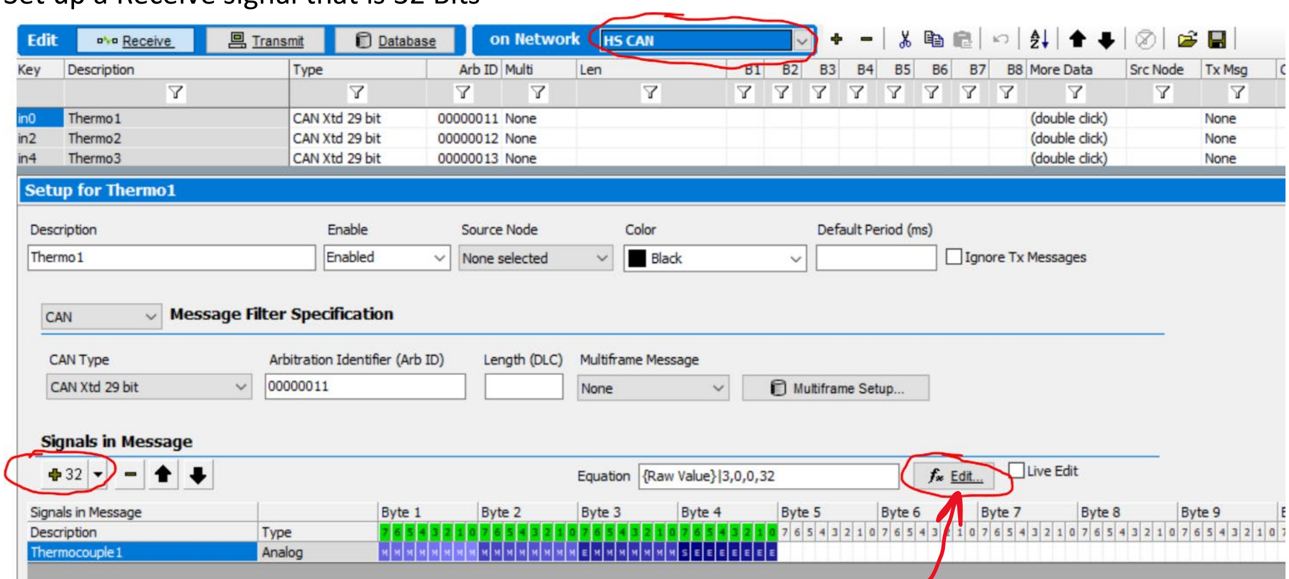
NEOVI EXAMPLE



Key	Description	Type	Arb ID	Multi	Len	B1	B2	B3	B4
in20	Thermo1	CAN Xtd 29 bit	00000011	None					
in24	Thermo2	CAN Xtd 29 bit	00000012	None					
in28	Thermo3	CAN Xtd 29 bit	00000013	None					
in30	Thermo4	CAN Xtd 29 bit	00000014	None					
in32	Thermo5	CAN Xtd 29 bit	00000015	None					
in34	Thermo6	CAN Xtd 29 bit	00000016	None					
in36	Thermo7	CAN Xtd 29 bit	00000017	None					
in38	Thermo8	CAN Xtd 29 bit	00000018	None					
in40	AIN1	CAN Xtd 29 bit	00000021	None					
in42	AIN2	CAN Xtd 29 bit	00000022	None					
in44	AIN3	CAN Xtd 29 bit	00000023	None					
in46	AIN4	CAN Xtd 29 bit	00000024	None					
in48	AIN5	CAN Xtd 29 bit	00000025	None					
in50	AIN6	CAN Xtd 29 bit	00000026	None					
in52	AIN7	CAN Xtd 29 bit	00000027	None					
in54	AIN8	CAN Xtd 29 bit	00000028	None					

Notice that the only thing different when setting up VSPY between a CAN wired channel or a NEOVI channel that is connected via USB, is the Network Channel; the drop down box **"on Network"** selection.

6.2 Set up a Receive signal that is 32 Bits



6.3 Then press the "fx Edit"

6.4 Select "32 Bit Floating IEEE Float", "Little End First …" you can also give it some Mins and Max values but this will not affect the output measurements, just future Graphical Panels if you use them. Your signals will now appear as decimal measurements.

RAD-IO2-AOUT - Analog Output

6.5 The Analog output module has 8 banks with 3 channels each.

6.6 If you are using the AOUT with the CAN-HUB you would select the appropriate HSCAN bus and not neoVI- if you are connected via USB directly to a FIRE2 or ION/Plasma and Industrial VCAN use NEO-VI.

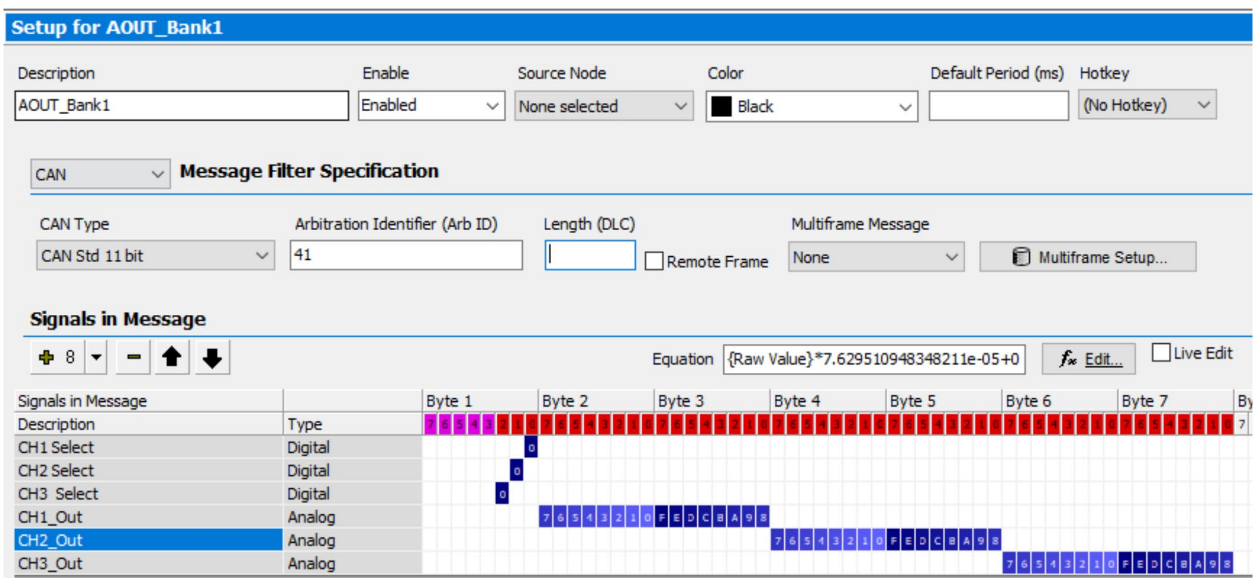
6.7 To understand how the system works, if you wanted to generate 2.5 volts using hex you would need to send 7F,FF. Desired voltage divided by Full scale voltage multiplied by 16 bits. Therefore 2.5 / 5 * 65535 = , or 32767.5; truncating to 32767 is 7FFF hex. 1 Volt would be 1 / 5 * 65535 = 13,107 = 3333 hex

6.7.1 In the first byte, the LSB bits 1,2,3 are the enable selector for each AOUT, then the subsequent 2 bytes (16 bits) are for the actual analog output.

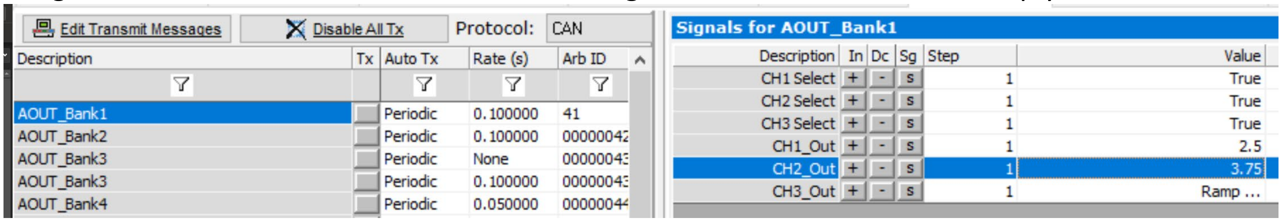
6.7.2 Keep in mind the Analog Output has three 0-5VDC 16 Bit DAC channels for each bank, for a total of 24 channels, and each bank is isolated from each other bank.

6.7.3 In this example the CH1,CH2 and CH3 are the selection mask. CH1_Out, CH2_Out, CH3_Out are the data bytes.

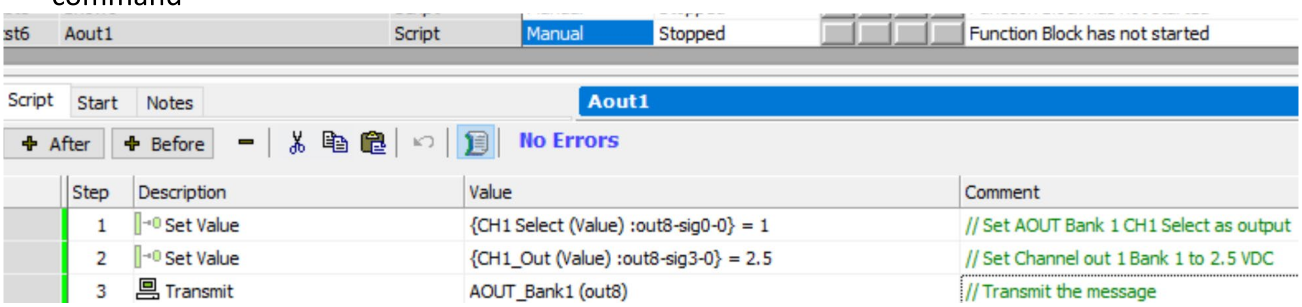
6.7.4 FOR EXAMPLE: If you wanted to send 2.5 VDC to Bank1 channel 1 you would need to transmit ARB-ID 0041 Byte1=01, Byte2=7F Byte3=FF.



6.8 Using the generated DBC file or your own Tx messages, if you wanted to use signals and generate 2.5 volts, 3.75 VDC and have VSPY generate a RAMP function simply use the Tx Panel



6.9 If you wanted to use function blocks you would of course use the SET VALUE and Transmit command

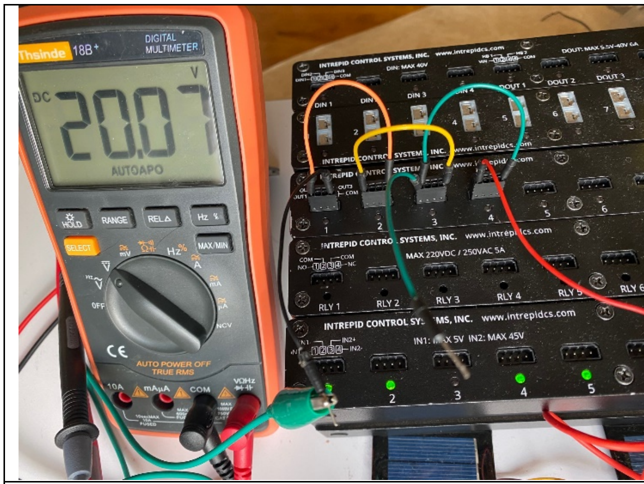
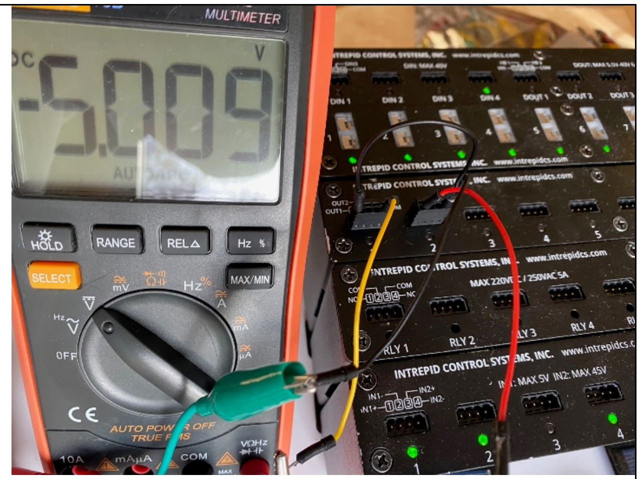


Analog Output Example

This example assume ARB ID 41h-48h

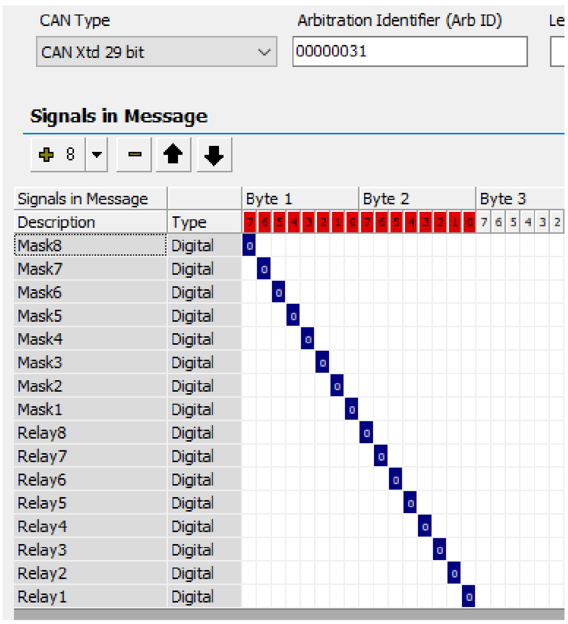
	CH Select	Channel 1		Channel 2		Channel 3		
ARB ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte3	
41	0	7F	FF					No Channel selected, Error response
41	1	7F	FF					Bank1, CH1, 2.5 VDC
41	1	FF	FF					Bank1, CH1, 5 VDC
41	1	0	0					Bank1, CH1, 0 VDC
41	2	0	0	7F	FF			Bank1, CH2, 2.5VDC
41	4	0	0	0	0	FF	FF	Bank1, CH3, 2.5VDC
42	1	7F	FF					Bank2, CH1, 2.5VDC
45	3	33	33	7F	FF			Bank5, CH1, 1VDC; CH2 2.5 VDC
48	7	33	33	7F	FF	FF	FF	Bank8, CH1, 1VDC; CH2 2.5 VDC, CH3 5VDC
48	7	0	0	0	0	0	0	Bank5, CH1, 1VDC

6.10 It is easy to generate more than 5VDC or generate negative polarity by wiring the signal and ground pins of the isolated banks together so that the individual voltages equal the sum.

	
Wiring up for Max 20VDC Bank1 Com to DMM minus, Bank1 CH1 to Bank2 Com, Bank2 CH1 to Bank3 Com, Bank3 CH1 to Bank4 Com, Bank4 to DMM plus	Wiring up for +/- 5VDC Bank1 CH1 to DMM Plus will show max -5VDC Bank1 Com to Bank2 Com; DMM minus for -5VDC Bank2 Sig to DMM Plus will show max +5VDC

RELAY MODULE

6.11 For the RELAY module, similar logic applies. Byte 1 contains the selector or mask, and Byte 2 the data. For the example sending a HI ("1") in data bit 1 on data byte 1 enables the Mask for relay 1, AND if in the same message you send HI ("1") in databit 1 on databyte 2, you will turn the Relay1 "ON". Assuming ArbID h0031 for the Relay module, sending a Transmit h0031, 01, 01, will turn relay 1 ON. Transmitting a h0031, 01, 00 will turn it OFF. Transmitting a h0031, 00, 00 will have no effect and will actually create a response back from the Relay module.



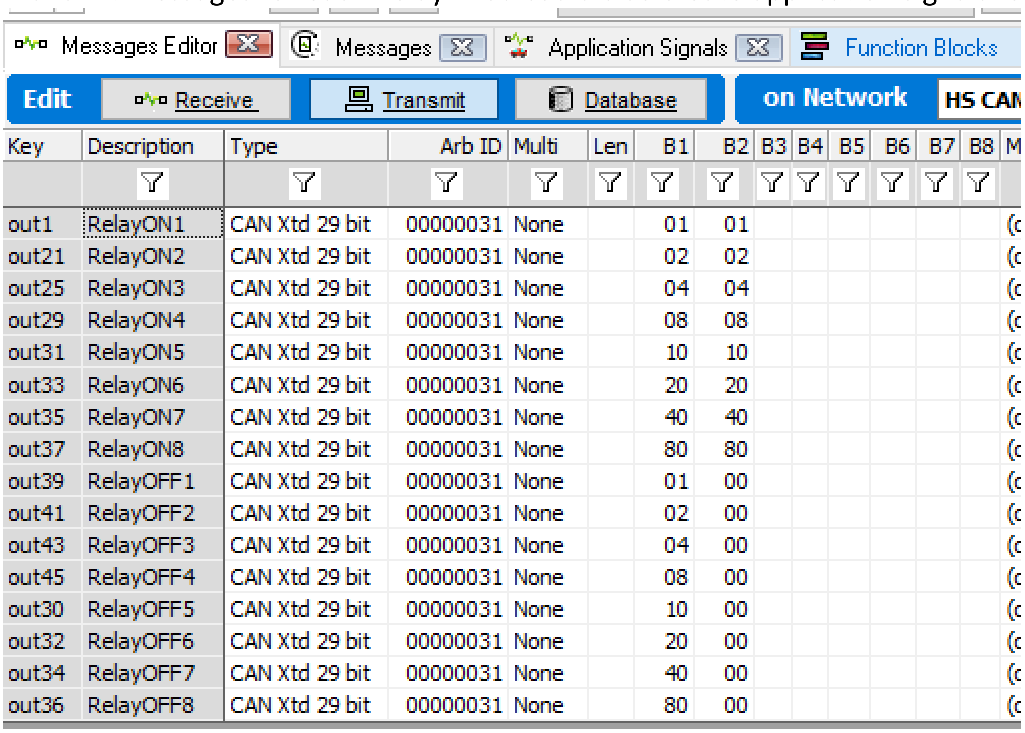
Relay Output example.

Relay Output Example;
Assume ARBID 0031h

	Byte1	Byte2		
Transmit 31h	0	0		You will receive a message back from the module, nothing will turn ON
Transmit 31h	1	1		Relay 1 will turn "ON"
Transmit 31h	1	0		Relay 1 will turn "OFF"
Transmit 31h	80	80		Relay 8 will turn "ON"
Transmit 31h	80	0		Relay 8 will turn "OFF"
Transmit 31h	1A	1A		Relay 2,4,5 will turn on
Transmit 31h	1A	0A		Relay 5 will turn OFF and relays 2,4 ON
Transmit 31h	1	FF		Relay 1 will turn ON, there will be no effect on the other relays

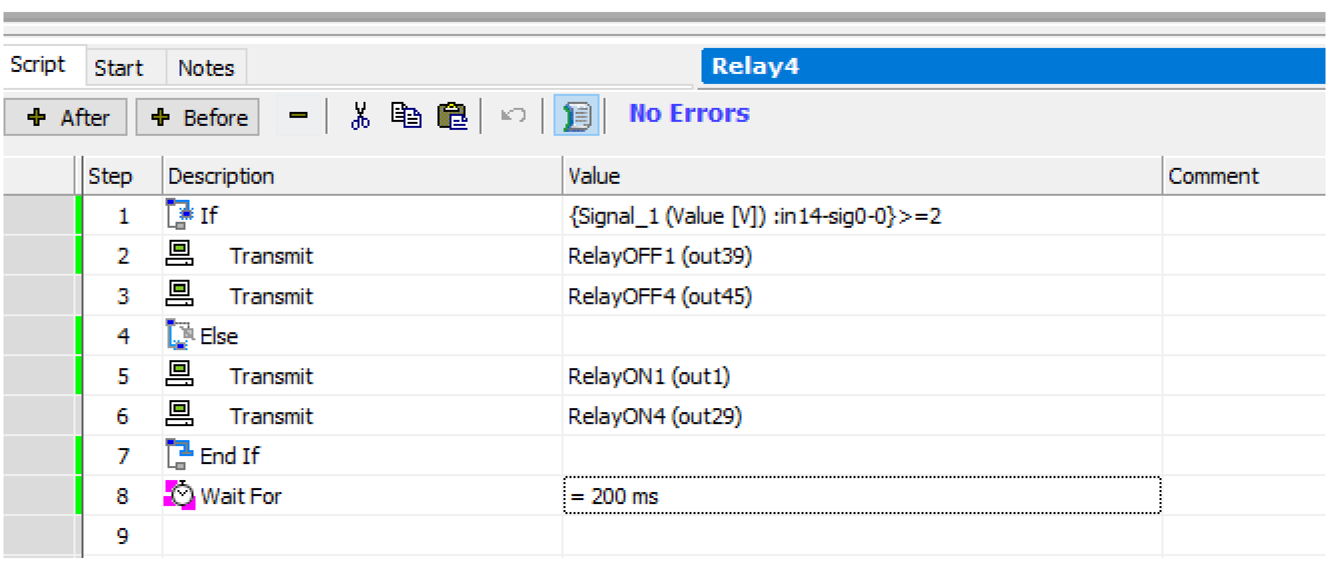
6.12 FOR EXAMPLE: we created 16 signals to illustrate the control over the relays.

One way to make function blocks to be more readable for yourself and future users is to create Transmit Messages for each Relay. You could also create application signals for each value.



Key	Description	Type	Arb ID	Multi	Len	B1	B2	B3	B4	B5	B6	B7	B8	M
out1	RelayON1	CAN Xtd 29 bit	00000031	None		01	01							(c
out21	RelayON2	CAN Xtd 29 bit	00000031	None		02	02							(c
out25	RelayON3	CAN Xtd 29 bit	00000031	None		04	04							(c
out29	RelayON4	CAN Xtd 29 bit	00000031	None		08	08							(c
out31	RelayON5	CAN Xtd 29 bit	00000031	None		10	10							(c
out33	RelayON6	CAN Xtd 29 bit	00000031	None		20	20							(c
out35	RelayON7	CAN Xtd 29 bit	00000031	None		40	40							(c
out37	RelayON8	CAN Xtd 29 bit	00000031	None		80	80							(c
out39	RelayOFF1	CAN Xtd 29 bit	00000031	None		01	00							(c
out41	RelayOFF2	CAN Xtd 29 bit	00000031	None		02	00							(c
out43	RelayOFF3	CAN Xtd 29 bit	00000031	None		04	00							(c
out45	RelayOFF4	CAN Xtd 29 bit	00000031	None		08	00							(c
out30	RelayOFF5	CAN Xtd 29 bit	00000031	None		10	00							(c
out32	RelayOFF6	CAN Xtd 29 bit	00000031	None		20	00							(c
out34	RelayOFF7	CAN Xtd 29 bit	00000031	None		40	00							(c
out36	RelayOFF8	CAN Xtd 29 bit	00000031	None		80	00							(c

This function block example turns OFF Relay 1 and Relay 4 if the voltage on the AIN module measured is greater than or equal to 2 volts and ELSE turns Relay 1 and Relay 4 ON if the voltage is less than 2 volts. We then delay for 200 mS. It is good to delay for some time after changing the relays state or they may chatter if a noisy input signal is used as the action item.



Step	Description	Value	Comment
1	If	{Signal_1 (Value [V]) :in14-sig0-0}>=2	
2	Transmit	RelayOFF1 (out39)	
3	Transmit	RelayOFF4 (out45)	
4	Else		
5	Transmit	RelayON1 (out1)	
6	Transmit	RelayON4 (out29)	
7	End If		
8	Wait For	= 200 ms	
9			

RAD-IO2-DIO

7.0 RAD DIO

7.1 Digital inputs have 2 shared settings Pre-scaler and Trip voltage. All channels use the settings found in channel 1.

- 7.1.1 Pre-scaler is a clock divider on the 1.2MHz main clock
- 7.1.2 Trip voltage is the Threshold voltage that separates high values from low.
- 7.1.3 Each channel has an invert bit that will cause the resulted input to be inverted. For example if the PWM output is 20% and you set the Invert bit, the PWM will be inverted so it will then be 80%.

7.2 There are 6 possible modes for DI, all of these modes can be changed "on the fly" while the unit is under operation. This allows the same physical input to sequence through the modes to take different measurements, such as Analog In, PWM and frequency.

- 7.2.1 RADIO2DIN_MODE_DISABLE:
  - 7.2.1.1 This disables the channel
  - 7.2.1.2 No values when this mode is set.
- 7.2.2 RADIO2DIN_MODE_DIGITAL:
  - 7.2.2.1 This mode returns the digital state of the input as a single byte
  - 7.2.2.2 0x00 = input low
  - 7.2.2.3 0x01 = input high
  - 7.2.2.4 These are reversed if invert = 1
- 7.2.3 RADIO2DIN_MODE_PWM:
  - 7.2.3.1 This mode returns the PWM duty cycle of the input as a single byte from 0 to 100 (%).
  - 7.2.3.2 The result will be 100 - duty cycle if invert set.
- 7.2.4 RADIO2DIN_MODE_PERIOD:
  - 7.2.4.1 Returns the high period of the last pulse in as a 16bit int
  - 7.2.4.2 The prescaler value can be used here to measure longer pulses
  - 7.2.4.3 The invert bit will switch to measuring the low time period
  - 7.2.4.4 Period = (value * prescaller) / 1200000 (seconds)
- 7.2.5 RADIO2DIN_MODE_FREQ:
  - 7.2.5.1 Returns the frequency of the given signal as 16 bit value
  - 7.2.5.2 Frequency = value / prescaler (Hz)
- 7.2.6 RADIO2DIN_MODE_ANALOG:
  - 7.2.6.1 This returns a 12bit values for the input voltage
  - 7.2.6.2 You can convert to voltage by (value * 40) / 4096 (Volts)

7.3 Digital Output (DO)

- 7.3.1 The digitals outputs are two half bridges
- 7.3.2 They can be separated or joined using the half bridge bit in channel 1, If the half bridge bit is set then channel 2 settings are ignored.
- 7.3.3 Frequency is a 16 bit value in Hz, this gives a range of 0Hz to 65535Hz
- 7.3.4 There are 4 modes for Digital Output (DO)

7.3.4.1 **Disabled**; which is the same as setting RADIO2DOUT_SET_HIZ, output is disabled. Digital Outputs do not change and are There is no value for digital mode

7.3.4.2 **DO**: This is a digital output.

7.3.4.3 **PWM**: PWM mode sets a % duty cycle (0-100%) and a frequency (0Hz to 65535Hz).

7.3.4.4 **Oneshot**: This sends a single pulse. The value sets the length of the pulse

7.3.5 There are 4 output states for the H-Bridge

7.3.5.1 RADIO2DOUT_SET_HIZ = 0; This is the equivalent to disabled, Outputs are not driven.

7.3.5.2 RADIO2DOUT_SET_LOW_REV. In this mode the H-bridge mode OUT1 is low and OUT2 is high: this would cause a motor hooked up to spin in reverse.

7.3.5.3 RADIO2DOUT_SET_HIGH_FWD. In this mode the H-bridge mode OUT1 is high and OUT2 is low: this would cause a motor hooked up to spin forwards.

7.3.5.4 RADIO2DOUT_SET_BRAKE. This sets all outputs low: in HBridge mode this would cause a motor to brake. If you turn a motor it will act as a generator. BRAKE mode shorts the tw windings of a motor together. This will stop a spinning motor much from faster than just setting HiZ the shorting action causes the magentinc fields to collapse immediately.

8.0 Support Contact Information

8.1 ICS United States Headquarters and West Coast office

Intrepid Control Systems, Inc.
31601 Research Park Drive
Madison Heights, MI 48071

Phone: (800) 859-6265 or (586) 731-7950
Fax: (586) 731-2274
Email: Email ICS

Intrepid Control Systems, Inc.
1925 Winchester Blvd, Ste 200
Campbell, CA 95008 USA

Phone: 1-800-859-6265 or (586) 731-7950
Fax: 586-731-2274
Email: Email Sales

8.3 ICS International Offices

European Union

Haid-und-Neu-Straße 7
76131 Karlsruhe
Germany
Phone: +49 721 1803083-0
Fax: +49 721 1803083 -9
Email: Email Sales

United Kingdom

Serviced Office Suite 1.03
MIRA Technology Park
Technology Centre NW05
Watling Street
Nuneaton
Warwickshire
CV10 0TU
Phone: +44 24 7718 0296
Email: Email Sales

China (Beijing)

https://www.intrepidcs.net.cn

Room 214, No.45 Chengfu Road,
Zhongguancun Zhizao Street
Block C, Haidian District,
Beijing P.R.China 100083
Phone: +86 176 8014 3205
Email: Email Sales

China (Shanghai)

https://www.intrepidcs.net.cn

Room 902, Building 16
No. 1000 Jinhai Road
City of Elite
Pudong, Shanghai
P.R.China 201206
Phone: 021-61637366
Fax: 021-61637366 * 600
Email: Email Sales

China (Shenzhen)

https://www.intrepidcs.net.cn

Room 22-YZ, Block A,
Che Kung Temple Fortune Plaza
No.5 Xianglin Road
Shenzhen, 518040 CHINA
Phone: +86 0755 82723212
Email: Email Sales

India

Office 306, B Building, Third Floor,
GO Square,
Wakad,
Pune, MH, 411057 India
Phone:
Sales: +91 77 55 99 00 70
Tech Support: +91 77 55 99 00 64
All Others: +91 77 55 99 00 74
Email: Email Sales

South Korea

http://www.intrepidcs.co.kr/

#1310, Keurancheu Techno B/D,
388, Dunchon-daero, Jungwon-gu,
Seongnam-si, Gyeonggi-do, 13403,
South Korea
Phone/전화: +82 31 698 3460
Fax/팩스: +82 31 698 3461
Email: Sales

Japan

http://www.intrepidcs.jp

164-0003
Shinryou Building 3F
Higashinakano 1-59-6
Nakano-Ku: Tokyo
電話番号: +81 03-5937-1523
ファックス: +81 03-5937-2524
営業 / 一般的なお問い合わせ: icsjapan@intrepidcs.com
技術サポート: icsjapansupport@intrepidcs.com

Australia

67A Hardiman St
Kensington, VIC 3031
Australia
Phone: 03 9466 4948
(International callers: +61 3 9466 4948)
Email: Julian Merritt

