ICP DAS Devices and MatDeck

Unlike most other available software, MatDeck allows you to integrate and combine MatDeck's C++ style script, Python, Drawings, Diagrams, Mathematical functions, Documentations, Tables, Custom Graphical User Interfaces (MatDeck visual GUI designer) and other features in the same file - MatDeck document. MatDeck and its numerous different features are all now available to ICP DAS series 2000 and 7000 devices.

Uses of GUIs for ICP DAS Devices

Using MatDeck, it is possible to set up multiple ICP DAS devices. The devices can be set via code (Python or MatDeck C++ style script) or via a dedicated GUI for the ICP DAS devices. The settings GUI is integrated together with the user code which in turn means the GUI is directly combined with the user's code. This unique MatDeck feature will reduce the amount of code needed to be written by a factor of 2 to 5. Additionally, you can avoid device setting code mistakes as well as simplifying the process of writing code.

In MatDeck, there are two GUI forms for configuring ICP DAS USB Series 2000 devices, and six different forms for configuring Series 7000 devices.

Configuration of ICP M-7026 Devices

In this example, we will illustrate two alternative ways of configuring the channels for ICP DAS M-7026 devices. MatDeck provides ICP DAS functions in the group Devices - ICP, and configuration can be done directly in MatDeck's C++ style Script. Additionally, you can configure devices in Python. However, the most effective and intuitive way for configuring the devices is by using MatDeck's GUI configuration.

This document is "a live document" and simultaneously executes commands, which is one of MatDeck's unique advantages.

ICP M-7026 supports communication using either Modbus, or DCON protocol. In both cases, commands are sent using appropriate MatDeck functions. The device, M-7026, is connected to the PC using the RS485 interface via a ICP i-7561U device.

The first step is to access the device, M-7026, by opening the communications and getting the device handle:

```
dev := icpcom_opendevice("COM4,9600,N,8,1")
```

Here, the string contains information about the COM port, baud rate - 9600, and data format N,8,1. dev contains the device's handle after opening. For further commands, the device handle is used. All commands in DCON protocol are defined as strings according to the user manual for the device. For example, command $AA7CiRrr is used to set the Type Code for a specific Analog Input channel of a specified module. Here, $ is the delimiter character, AA is the address of the module to be set in hexadecimal format (00 to FF), 7 is the command used to set the channel Type Code, Ci specifies the Analog Input channel to be set, zero based (0-5), Rrr represents the Type Code to be used for the Analog Input channel. The connected device has the address, 01. We want to set the channel 0 to the input range -1V to +1V which is represented by the type code 0A:

```
icpcom_sendcmd(dev, "$017C0R0A")
```

The input value can be read using the standard read function:

```
aivalue := icpcom_ai_read(dev, 1, 0, 6)
```

After, we finish by using the device it is necessary to close device and release the handle

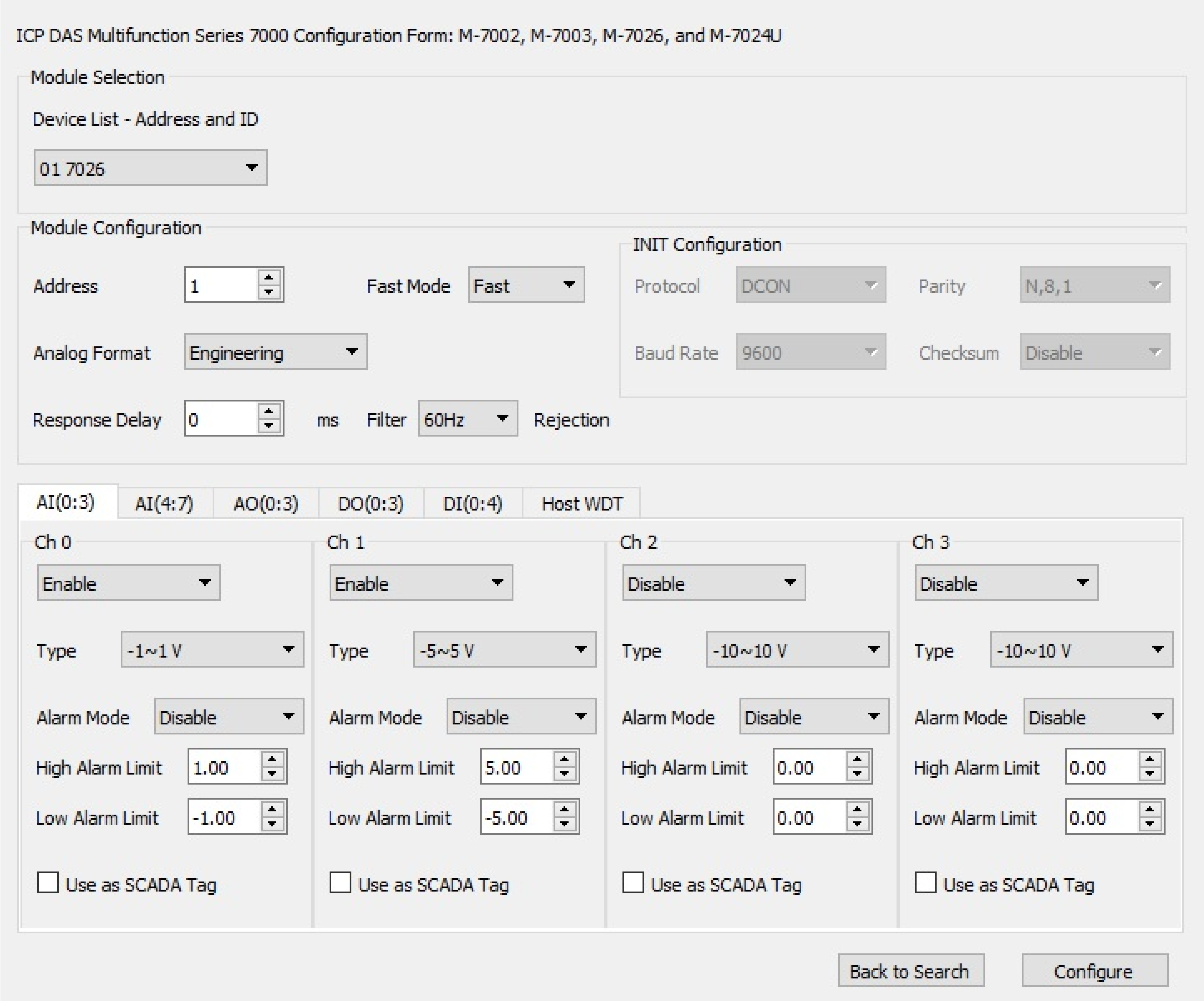
```
icpcom_closedevice(dev)
```

$$\text{aivalue} = 0.001$$

GUIs for Configuration

Configuration using DCON protocol requires knowledge of DCON commands, device configuration tables and programming to prepare all these scripts in a very intuitive manner. ICP devices can be easily set by using the GUI form, icpcom_multifunction7000_form. All lines of script code, and more important all DCON commands, are substituted by GUI settings. The same form can be used setup all the device pins.

form := icpcom_multifunction7000_form(0 , "Form1")



ICP DAS Multifunction Series 7000 Configuration Form: M-7002, M-7003, M-7026, and M-7024U

Module Selection

Device List - Address and ID: 01 7026

Module Configuration

Address: 1

Fast Mode: Fast

Analog Format: Engineering

Response Delay: 0 ms

Filter: 60Hz Rejection

INIT Configuration

Protocol: DCON

Parity: N,8,1

Baud Rate: 9600

Checksum: Disable

AI(0:3)

	Ch 0	Ch 1	Ch 2	Ch 3
	Enable	Enable	Disable	Disable
Type	-1~1 V	-5~5 V	-10~10 V	-10~10 V
Alarm Mode	Disable	Disable	Disable	Disable
High Alarm Limit	1.00	5.00	0.00	0.00
Low Alarm Limit	-1.00	-5.00	0.00	0.00
	□ Use as SCADA Tag	□ Use as SCADA Tag	□ Use as SCADA Tag	□ Use as SCADA Tag

icpcom_multifunction7000_form_configure(form) Configure function

After ICP device is set, analog and digital input are read using standard read functions, and analog and digital output are accessed using write functions:

```
dev1 := icpcom_opendevice("COM4,9600,N,8,1")
aivalue1 := icpcom_ai_read(dev1, 1, 0, 6)
icpcom_closedevice(dev1)
```

$$\mathrm{aivalue1} = 0.001$$