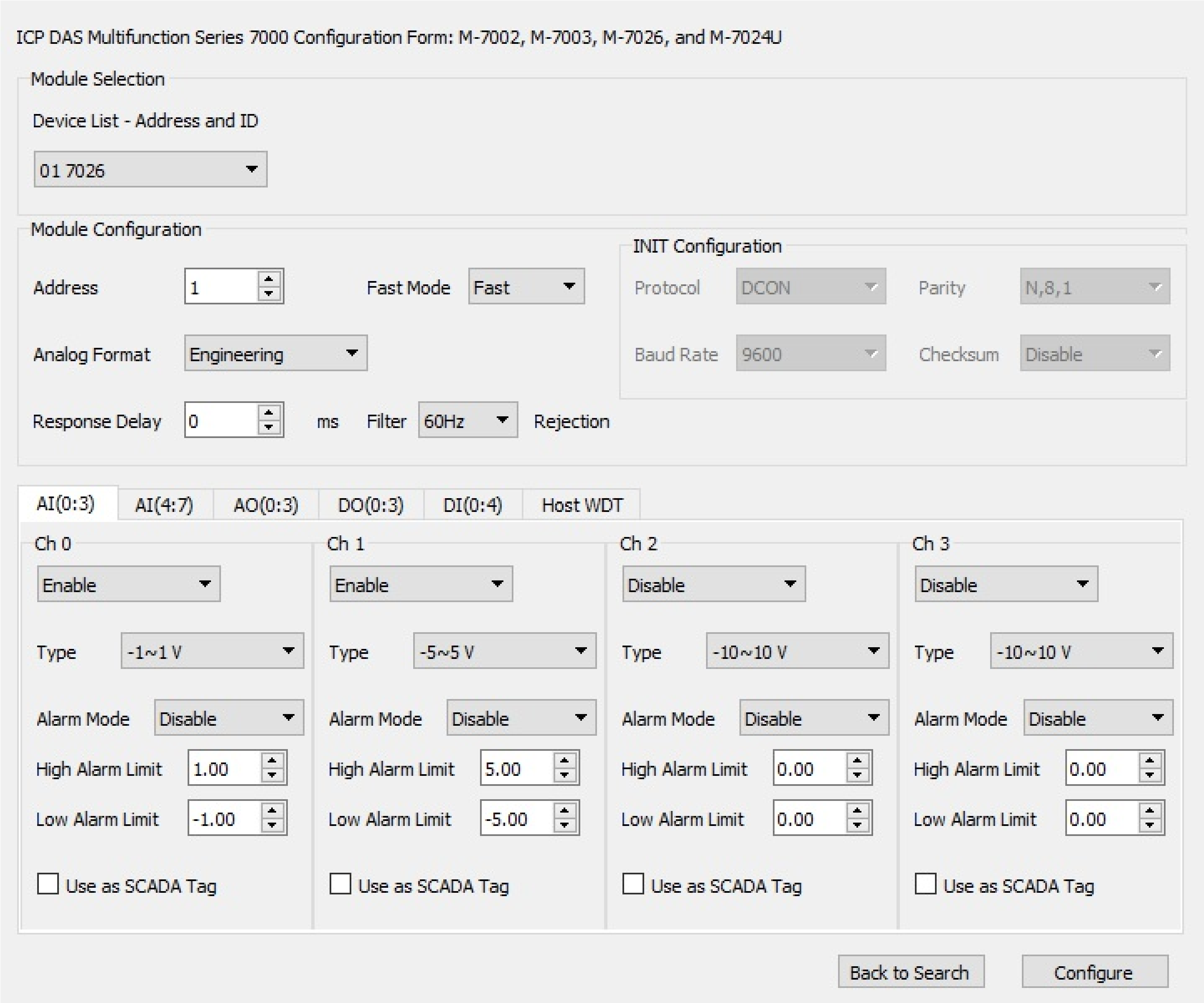
# ICP DAS Devices and MatDeck Features

MatDeck allows users to combine MatDeck's C++ style script, Drawings, Diagrams, Mathematical functions, Documentations, Tables, and other features all in the same file - MatDeck document. MatDeck and its numerous different features are all available on ICP DAS series 2000 and 7000 devices. In this example, we illustrate how ICP DAS Devices can be used with numerous MatDeck features. ICP device M-7026 is configured using MatDeck's GUI form which is designed for the purpose. Once configured, MatDeck's mathematical functions can be used to generate signals. The results are recorded in a xls file in the form of report.

## Configuring a ICP M-7026 Device using GUIs

ICP device M-7026 can be easily set by using GUI form, icpcom_multifunction7000_form. All the numerous lines of script code, and more importantly, all DCON commands are substituted by the GUI's settings. The same form can be used to setup all of the device pins. The document is "a live document" and simultaneously executes commands. This is one of MatDeck's unique advantages.

form := icpcom_multifunction7000_form(0 , "Form")



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

| | 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 the ICP device is set, analog and digital inputs are read using standard read functions. Analog and digital outputs are also accessed using write functions

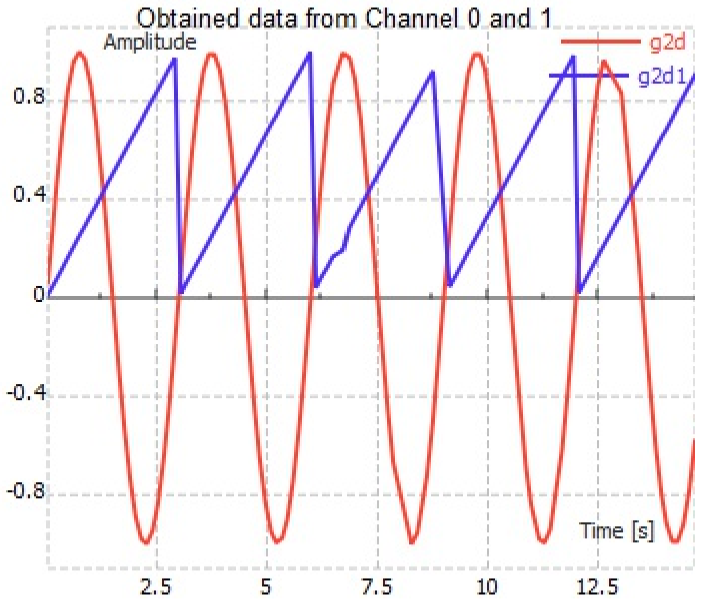
## Combining Mathematics with ICP Devices

We can generate sin functions with given frequencies based on the current time in the MatDeck script. First, the device, M-7026 should be opened.

```
dev1 := icpcom_opendevice("COM4,9600,N,8,1")
```

The handle, dev1, is used to access the devices further. Next, the signals are generated and sent to the analog output channels, 0 and 1. Finally, the signal is read at analog inputs 0 and 1. The readings are packed into vectors and displayed in a 2d graph.

```
T0 := timenow() // starting time
fre := 1 /3 // frequency of sinusoid in Hz
period := 3 // period of triangular in s (saw tooth function)
numit := 100 //nuber of iterations
val := vector_create(numit, false, 0)
st := vector_create(numit, false, 0)
tim := vector_create(numit, false, 0)
for(i := 0; i < numit; i += 1)
{
  ttemp :=(timenow() -T0) //current time
  temp := sin(2 * fre * cpi() * ttemp) //current sine value
  stemp := -mod(ttemp, period) / period //current saw tooth value
  icpcom_ao_write(dev1, 1, 0, 3, temp) //write ao value at ch 0
  icpcom_ao_write(dev1, 1, 1, 3, stemp) //write ao value at ch 1
  val[i] = icpcom_ai_read(dev1, 1, 0, 6) //read current ai value at ch 0
  tim[i] = ttemp
  st[i] = icpcom_ai_read(dev1, 1, 1, 6) //read current ai value at ch 1
}
g2d := join_mat_cols(tim, val) //prepare graph
g2d1 := join_mat_cols(tim, st) //prepare graph
```



After all operations are done, the device handle should be released by closing it.

```
icpcom_closedevice(dev1)
```

## Results

There will be N measurements, which is set above. The first 10 measurements will be automatically displayed in the table. After all N measurements, the data is exported to a xlsx file. We read the voltage at the analog input channel 0 and the analog input channel 1.

```
TableH := ["Time", "Voltage AIN 0", "Voltage AIN 1"]
Data := join_mat_cols(subset(tim, 0, 0, 9, 0),subset(val, 0, 0, 9, 0))
Data = join_mat_cols(Data, subset(st, 0, 0, 9, 0))
Table := table_create(Data, TableH)
```

Table =

| Time | Voltage AIN 0 | Voltage AIN 1 |
|---|---|---|
| 0.031 | 0.064 | 0.015 |
| 0.225 | 0.456 | 0.076 |
| 0.358 | 0.681 | 0.121 |
| 0.480 | 0.843 | 0.164 |
| 0.613 | 0.960 | 0.208 |
| 0.743 | 1 | 0.253 |
| 0.868 | 0.970 | 0.292 |
| 0.999 | 0.868 | 0.337 |
| 1.124 | 0.711 | 0.376 |
| 1.255 | 0.490 | 0.42 |

## Exporting Data to Excel Files

Here, the data obtained by the three measurements will be exported to a Excel file at appropriate positions. The variables are exported manually.

```
excel_write("measurements.xlsx","Sheet1","A1","Time")
excel_write("measurements.xlsx","Sheet1","B1",tim)
excel_write("measurements.xlsx","Sheet1","A2","Voltage AIN 0")
excel_write("measurements.xlsx","Sheet1","B2",val)
excel_write("measurements.xlsx","Sheet1","A3","Voltage at AIN 1")
excel_write("measurements.xlsx","Sheet1","B3",st)
```

## Exporting to Text Files

Measurement data can be exported to a txt file as well. For example, the txt file will be made if the maximal voltage at AIN0 is above 0.5V. The sentence "The measurement is successful" will be written in the test.txt file.

```
dat:= datef("d/m/y")
timc:= timef(":")
```

```
if(mat_max(val) > 0.5)
{
  file_name:= datef("d-m-y")+" "+ timef("-")
  F:= file_create(file_name,"text",true)
  file_write(F,"The measurement is successful\n")
  file_write(F, mat_max(val))
  file_write(F,"\n"+ dat +" "+ timc)
  file_close(F)
}
```