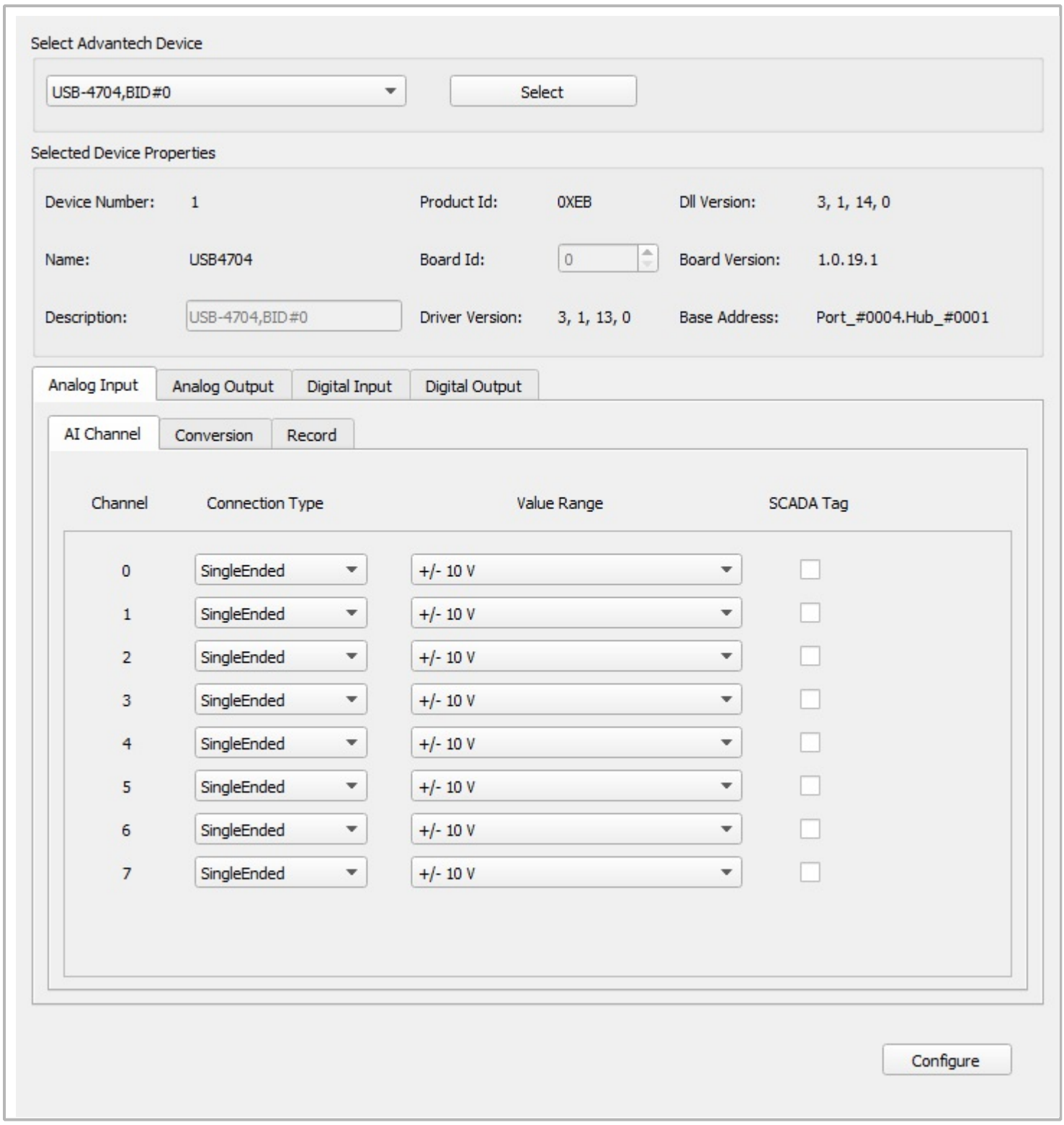
Advantech - Analog Output

In this example, we illustrate how to produce analog outputs at a given AO channel. There are several different ways to produce the output. The first step is to configure the Advantech device using an Advantech GUI.

```
AO := atconfig_form(0, "AO1", "")
```



The easiest way to use the device further on, is to export the device handle from the form, which is done in the canvas below. In the given scenario, we are using the Advantech USB-4704 device. In order to test and display the Analog Output, we have connected AO0 to AI0. We need two handles for AO and AI.

```
atconfig_form_configure(AO)
AOhandle := atconfig_form_device_handle(AO , "ao")
AIhandle := atconfig_form_device_handle(AO , "ai")
```

Different options for Analog Output generation

Predefined Output Signal

The waveform data is generated in such a manner that there is a single column vector which contains data to be written to AOs in parallel. Here, the main interest is the signal shape and not timing. We start by defining the length of the signal, x axis, and waveform.
Define the x axis and sine waveform

```
ONE_WAVE_POINT_COUNT := 512
xaxis := ynodes(x, 0, 511, 512)
waveform := sin(2 * cpi() * xaxis / 100)
```

Creating data for the graph

```
AO1 := join_mat_cols(xaxis, waveform)
//Vector to read AI
AIread := vector_create(size(waveform), false, 0)
```

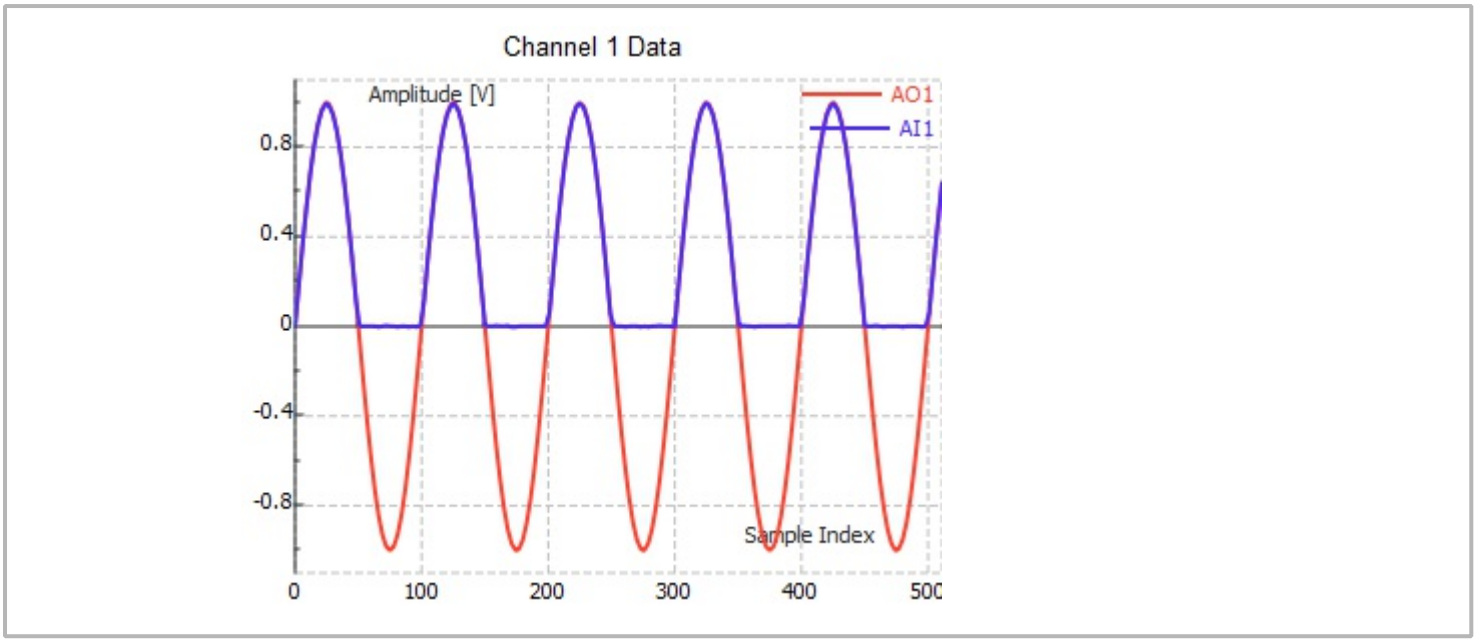
Next, we select the AO channel to write:

```
chanStart := 0
```

Loop for writing to AO0 and reading from AI0 which is connected to AO0.

```
for(i := 0; i < ONE_WAVE_POINT_COUNT; i += 1)
{
  atdevice_ao_write(AOhandle, chanStart, waveform[i])
  AIread[i] = atdevice_ai_read(AIhandle, chanStart)
}
AI1 := join_mat_cols(xaxis, AIread)
```

In graph below, the x axis is sample index, and it is not related to time.

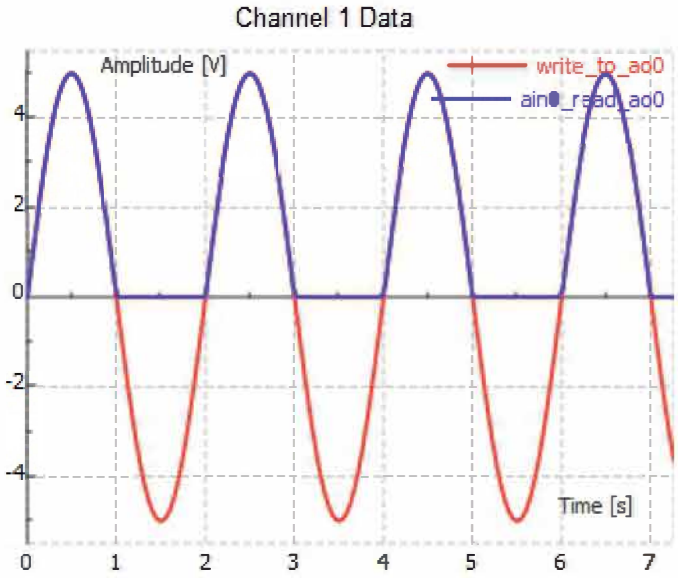


Output Signal in Real Time

In real time, the signal which we generate will be related to current time. The frequency of the signal is defined manually. Again, we will generate the same number of samples. The time of execution is measured and it depends on the device which is used. Here, we used the Advantech USB-4704 device.

```
freq := 0.5 //Frequency in Hz
t0 := timenow()
for(i := 0; i < ONE_WAVE_POINT_COUNT; i += 1)
{
  xaxis[i] = timenow() - t0
  waveform[i] = 5 * sin(2 * cpi() * xaxis[i] * freq)
  atdevice_ao_write(AOhandle, chanStart, waveform[i])
  AIread[i] = atdevice_ai_read(AIhandle,chanStart)
}
write_to_ao0 := join_mat_cols(xaxis, waveform)
ain0_read_ao0 := join_mat_cols(xaxis, AIread)
```

The data output at AO0 is displayed in the graph below.


Channel 1 Data
Amplitude [V]
write_to_ao0
ain0_read_ao0
4
2
0
-2
-4
Time [s]
0
1
2
3
4
5
6
7