

Advantech - Analog Output, Universal Example

In this example, we illustrate how to produce analog outputs at a given AO channel. The process is divided into several steps. The first step is to configure the range of the desired output channels. The AO range should be set to +/- 10V. The range can be set using a GUI, the device selection should be changed in the GUI.

```
1 A0 := atconfig_form(0, "A010", "")
```

The screenshot displays the Advantech configuration interface. At the top, there is a section for selecting a device, with a dropdown menu showing "USB-4704,BID#0" and a "Select" button. Below this, the "Selected Device Properties" section lists various attributes: Device Number (1), Product Id (0XEB), Dll Version (3, 1, 14, 0), Name (USB4704), Board Id (0), Board Version (1.0.19.1), Description (USB-4704,BID#0), Driver Version (3, 1, 13, 0), and Base Address (Port_#0004.Hub_#0001).

The interface has four tabs: "Analog Input", "Analog Output", "Digital Input", and "Digital Output". The "Analog Output" tab is currently selected. Within this tab, there are three sub-tabs: "AI Channel", "Conversion", and "Record". The "AI Channel" sub-tab is active, showing a table with columns for Channel, Connection Type, Value Range, and SCADA Tag.

Channel	Connection Type	Value Range	SCADA Tag
0	SingleEnded	+/- 10 V	<input type="checkbox"/>
1	SingleEnded	+/- 10 V	<input type="checkbox"/>
2	SingleEnded	+/- 10 V	<input type="checkbox"/>
3	SingleEnded	+/- 10 V	<input type="checkbox"/>
4	SingleEnded	+/- 10 V	<input type="checkbox"/>
5	SingleEnded	+/- 10 V	<input type="checkbox"/>
6	SingleEnded	+/- 10 V	<input type="checkbox"/>
7	SingleEnded	+/- 10 V	<input type="checkbox"/>

At the bottom right of the interface, there is a "Configure" button.

```
atconfig_form_configure(AO)
```

```
AOhandle := atconfig_form_device_handle(AO, "ao")
```

```
AIhandle := atconfig_form_device_handle(AO, "ai")
```

The waveform data is generated so that there is a single column vector which contains data to be written to AOs in parallel. The function GenrateWaveform() is given at the end of this document.

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

Next, we select the AO channel to write:

```
6 chanStart := 0
7 chanCount := 1
```

Everything is ready to output data, which is done in the remaining two steps.

Step 2: Output the data

```
8 ai_var := vector_create(size(waveform), false, 0)
9 freq := 0.5
10 t0 := timenow()
11 for(i := 0; i < ONE_WAVE_POINT_COUNT; i += 1)
12 {
13     xaxis[i] = timenow() - t0
14     waveform[i] = 5 * sin(2 * cpi() * xaxis[i] * freq)
15     atdevice_ao_write(AOhandle, chanStart, waveform[i])
16     ai_var[i] = atdevice_ai_read(AIhandle, chanStart)
17 }
18 write_to_ao0 := join_mat_cols(xaxis, waveform)
19 ain0_read_ao0 := join_mat_cols(xaxis, ai_var)
```

Step 3: Close the device

```
20 atdevice_close(AOhandle)
21 atdevice_close(AIhandle)
```

The data output at AO0 is displayed in the graph below

Channel 1 Data

