

# Advantech - Analog Output

In this example, we illustrate how to produce analog outputs at a given AO channel. This process is divided into several steps. The first step is to configure the range of 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 line 1.

```
1 sel := "USB-4704,BID#0"  
2 AO := atconfig_form(0, "A011", sel)
```

Select Advantech Device

USB-4704,BID#0

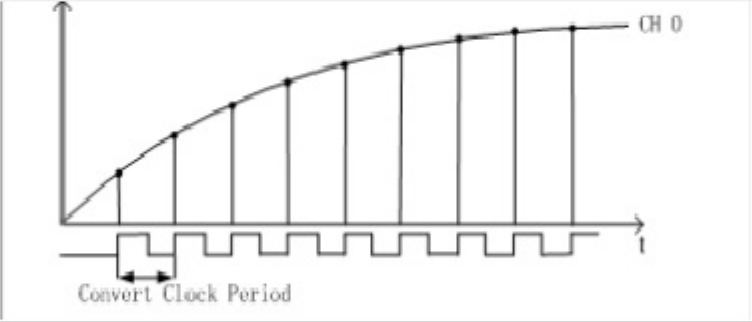
Selected Device Properties

Device Number:	1	Product Id:	0XE8	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	Base Address:	Port_#0004.Hub_#0001

Analog Input **Analog Output** Digital Input Digital Output

AI Channel **Conversion** Record

Schematic diagram of convert clock working principle



Start Channel To Scan: 0

Channel Count To Scan: 1

Convert Clock Source: Internal Clock

Convert Clock Rate: 1000.00

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

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

Next, we select the AO channel to write:

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

Everything is ready to output data, which is done in three steps.

Step 1: Open the AO device in non- buffered mode.

```
9 sel1 := 1 //Select device
10 // Open AO device in nonbuffered mode.
11 AOhandle := atdevice_ao_open(sel1, false)
12 AIhandle := atdevice_ai_open(sel1, false)
```

Step 2: Output the data

```
13 ai_var := vector_create(size(waveform), false, 0)
14 freq := 0.5
15 t0 := timenow()
16 for(i := 0; i < ONE_WAVE_POINT_COUNT; i += 1)
17 {
18     xaxis[i] = timenow() -t0
19     waveform[i] = 5 * sin(2 * cpi() * xaxis[i] * freq)
20     atdevice_ao_write(AOhandle, chanStart, waveform[i])
21     ai_var[i] = atdevice_ai_read(AIhandle,chanStart)
22 }
23 write_to_ao0 := join_mat_cols(xaxis, waveform)
24 ain0_read_ao0 := join_mat_cols(xaxis, ai_var)
```

Step 3: Close the device

```
25 atdevice_close(AOhandle)
26 atdevice_close(AIhandle)
```

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

Channel 1 Data

