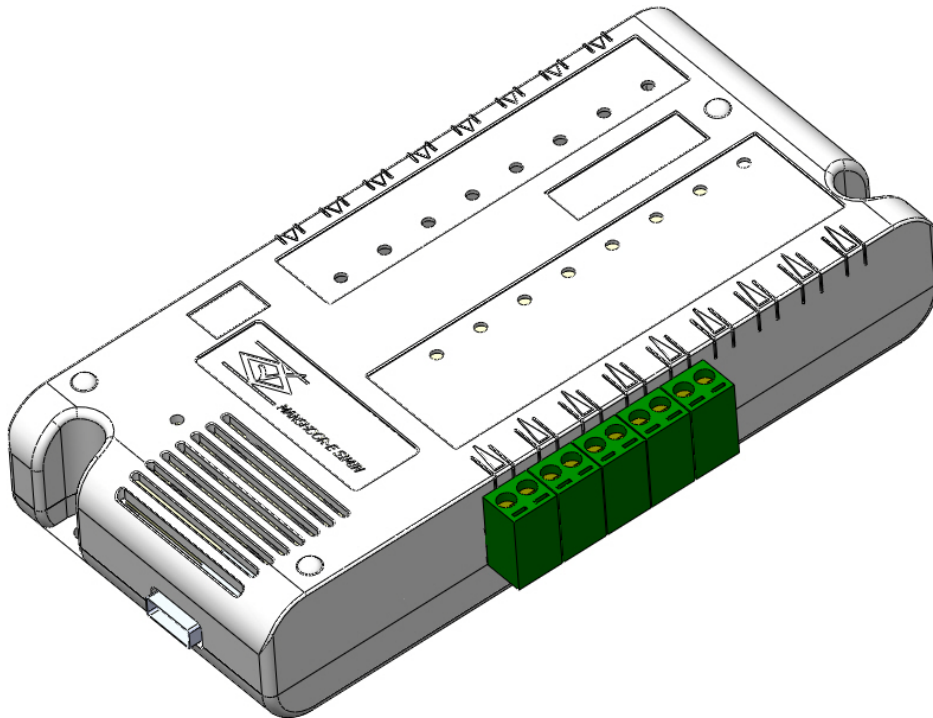


NANO Digital/Analog 8 Channels



**D/A Interface from a PC's USB port
Installation and Users Manual
&
Software Guide**

**Available exclusively from
Manshoore Simin Ltd Co.
www.msbbs.com**

Copyright ©2011 Manshoore Simin Ltd Co.

Contents

1. Introduction
2. Getting Started
3. Electronic Schematic
4. Software Sample
5. Software Options
 - 5.1. Set All Channels
 - 5.2. Set Channel 1 to 8
 - 5.3. Change USB ID
6. Writing your own software for NANO D/A Interface
 - 6.1. Using NANO D/A Interface by Delphi.

1. Introduction

The Nano Digital to Analog board kit enables your PC application to allowing you to Set 10 bit digital from PC application to 8 output channels to convert analog voltage. Outputs can be a voltage from 0-5VDC and used for, such as motor speed controlling, light controlling and any things that need to variable voltage range.

The logic circuits operate from 3.3V to 5V making them compatible with PC USB port variable voltage range.

The microcontroller in the kit, is used for industrial environments.

2. Getting Started

By connecting D/A Interface to USB PC port, the HID device driver will install automatically. In other word, Windows will automatically recognize it and configure it's driver accordingly. Windows 2000, XP, Vista and Windows 7, support the device.

After install, insert the device CD into CD drive and go to CD root, then copy the root folder in to your hard disk and execute the "File Register" exe file and click on "Register File" button and select "MSUSBI.ocx" then ok if the file registers successfully. Then execute "HID_Device_Sample.exe".

3. Electronic Schematic

The NANO D/A Interface has 8 regulated analog output channels with variable voltage range that for making all connections to the external device and has 2 pin for common grand for these 8 analog outputs.

Speed rate for this product is 18 ms.

This do not need any Power Supplier, rather supplied from PC USB Port; Designed into the interface that USB voltage first convert to 12V then regulate to 5V for reason, the voltage level do not dependent on USB voltage.

Figure1 illustrate the interface pin out.

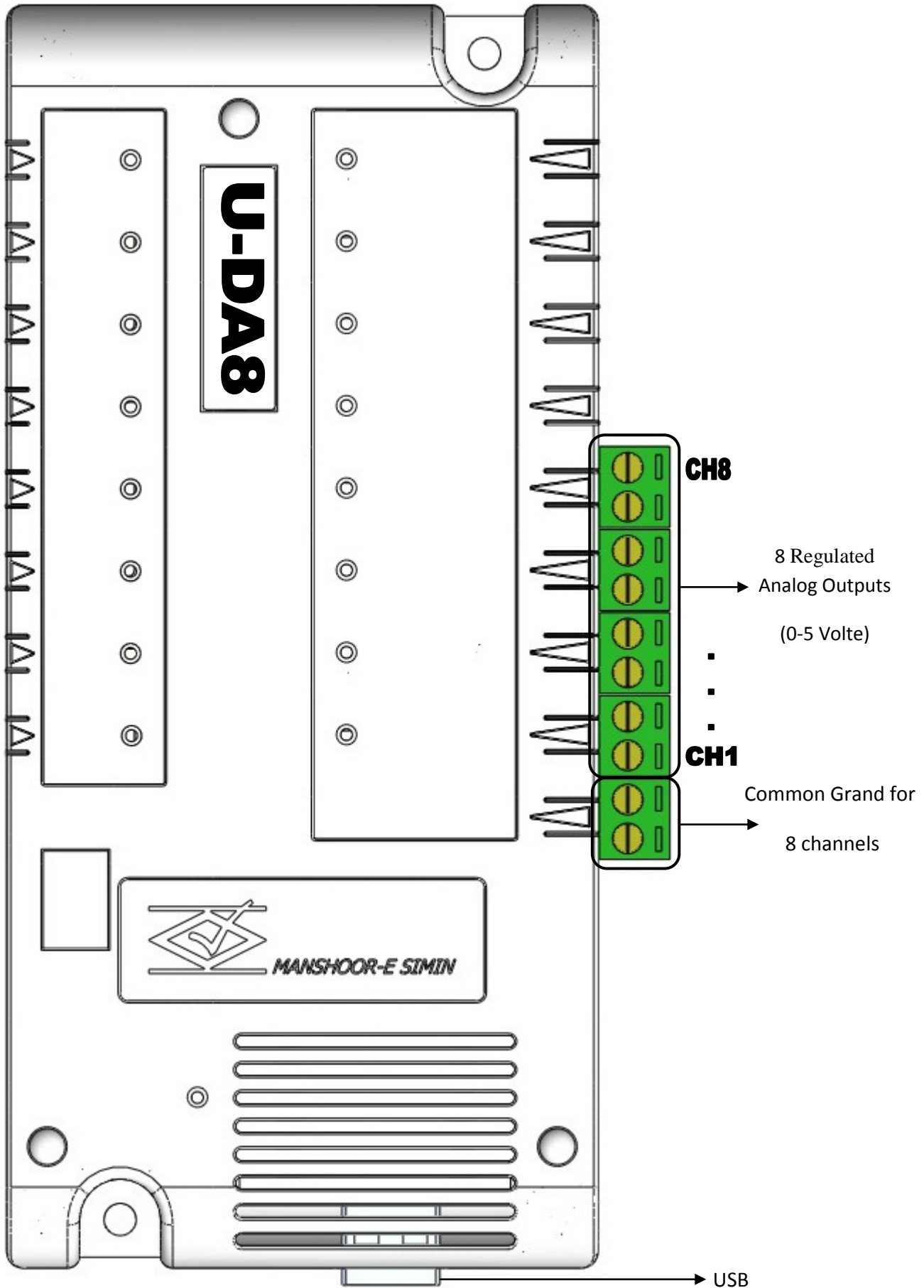


Figure 1

Pin	Description
Analog Output	8 Regulated Analog Output Channels. Voltage Range: 0-5 VDC. Digital value: 10 Bit.
Grand	2 Common Channels to Grand for 8 Analog Output Channels.

4. Software Sample

The Nano D/A Interface is supplied with this ready sample which makes it very easy for the beginner to get quickly up and running with device control.

Installation, as described above, is painless and easy requiring only a Windows 2000, XP, Vista or Windows7, computer with fairly modest specifications. To run the software double click on the "HID_Device_Sample.exe". The "HID_Device_Sample" environment screen will then appear providing the workspace for doing operation. By running the software, last interface will be selects automatically that indicated in the "Select Interface" combo box. User can selects the interface from combo box or insert device index and then click on "Select Device" button.

By click on "Device Name" button, after insert index of the device, the interface name show in the "Name" edit box.

To show number of interfaces that connect to USB port, click on "Device Count".

When the interface would be connect or disconnect on USB port, click on "Update Device" button to refresh list of interfaces. (figure2)

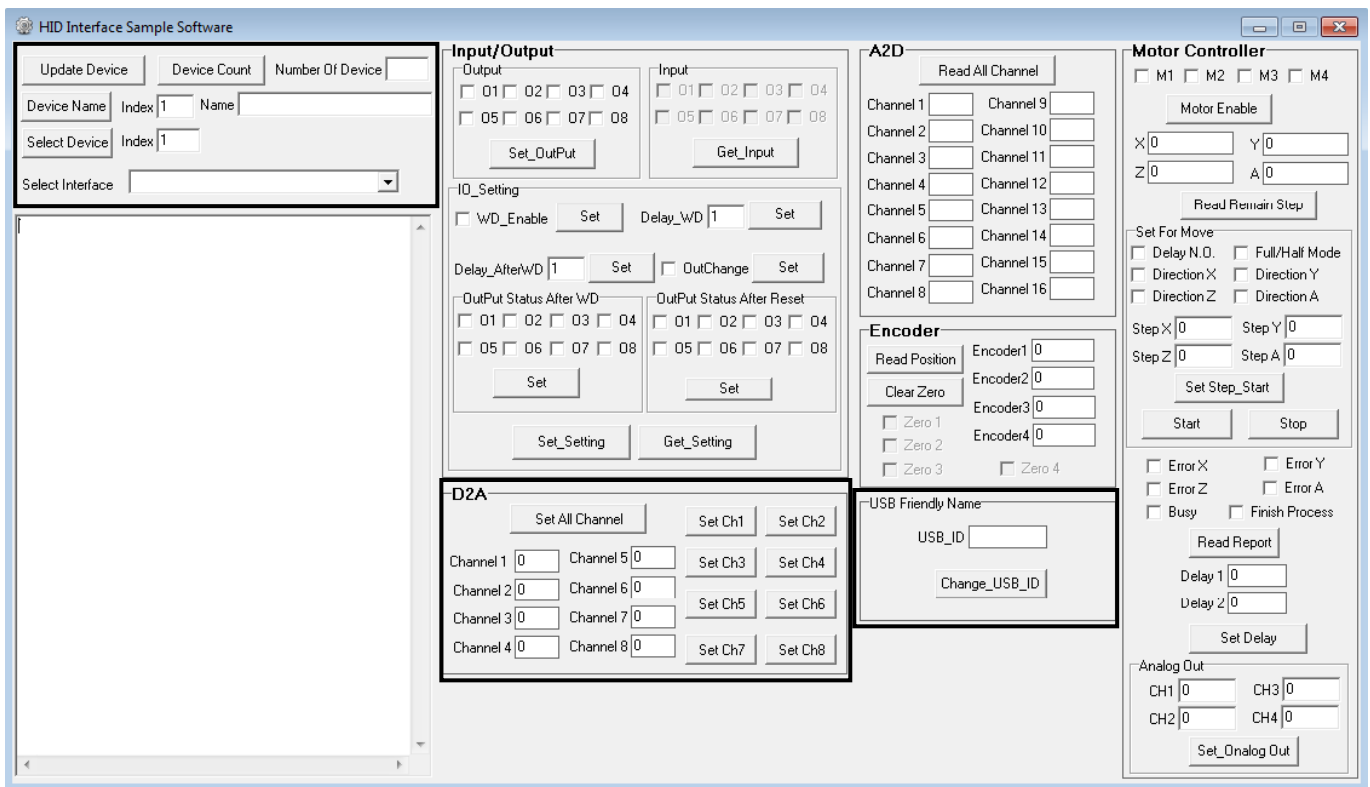


Figure 2

5. Software Options

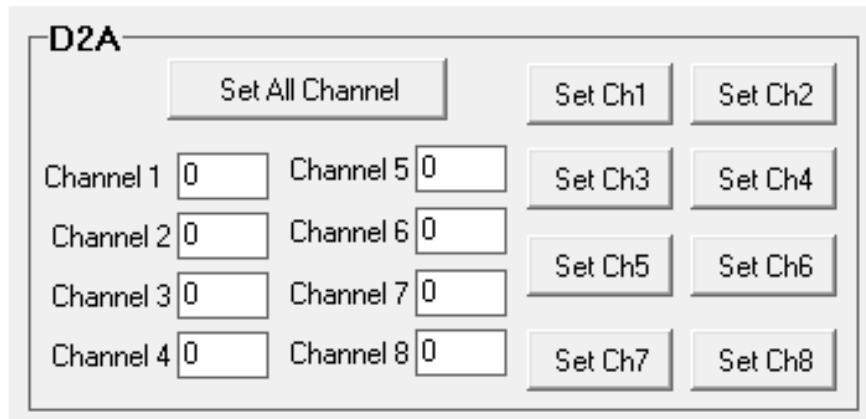


Figure 3

5.1. Set All Channels

To set all outputs at once, set "channel1" to "channel8" edit box values from 0 to 4095, and then click on "Set All Channel" button. Outputs will change from 0 to 5 volte linear.

5.2. Set Channel 1 to 8

To set outputs separately, set "channel(n)" edit box value from 0 to 4095, and then click on "Set Ch(n)" button. The output for this channel will change from 0 to 5 volte linear.

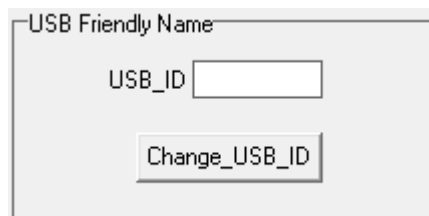


Figure 4

5.3. Change USB ID (Figure 4)

By connecting more than, one similar interface type into USB port, all of these interfaces have only one friendly name in the list, for distinct of these similar names, user can change ID of the interface, default ID for these interfaces is 0000. User can change these four characters. For example if USB friendly name in the device list is "M.S.D/A 0000", user can change "0000" to any character like "Test", after this change, the interface must be disconnect and then connect to the USB port, by connecting , USB friendly name in the device list change to "M.S.D/A Test". With this change user can distinct between two similar interfaces.

6. Writing your own software for NANO D/A Interface

Provided with this interface an ocx(Active X) file, called "MSUSBI.ocx". This file encapsulates the functions used by sample software in communicating with D/A interface across the USB interface into any simple functions easily understood and used in custom software. Although the ocx is written in "Delphi" it can be used (called) by programs written in a number of popular languages, the popular of which are Visual BASIC, C#, Visual C. Below described are the techniques to use the ocx in Delphi and

Visual Basic. If you program in another language please refer to your compiler manual on the details of importing and calling a ocx functions, which will be very similar to the techniques described below.

6.1. Using NANO D/A Interface by Delphi.

Using NANO D/A Interface with your own programs written in Delphi is very simple. At the head of your program, before using any of the NANO D/A Interface function, you must Import "MSUSBI.ocx" in your programming language and insert the ocx in your program form....

//If Connect or disconnect a interface to USB port call this function//

```
MSUSBDevice1.DeviceUpdate();  
////////////////////////////////////
```

//Check how many device is connect to USB port//

```
integer := MSUSBDevice1.DeviceCount();  
////////////////////////////////////
```

// Get friendly name of USB interface//

```
string := MSUSBDevice1.ListOfDevice(Device index:integer);  
////////////////////////////////////
```

//Select the interface via Device number before doing any work//

```
if not MSUSBDevice1.SelectDevice(Device NO:integer) then  
  ShowMessage('Device Could not Select');  
////////////////////////////////////
```

//Set All Outputs //

```
if not MSUSBDevice1.D2A_SetAllCh (Output1, Output2, Output3, Output4, Output5, Output6,  
Output7, Output8:Integer) then  
  Memo1.Lines.Add('Failed To Send.');
```

//Set Output1 //

```
if not MSUSBDevice1.D2A_SetCh1(O1:Integer) then  
  Memo1.Lines.Add('Failed To Send.');
```

//Set Output2 //

```
if not MSUSBDevice1.D2A_SetCh2(O2:Integer) then  
  Memo1.Lines.Add('Failed To Send.');
```

//Set Output3 //

```
if not MSUSBDevice1.D2A_SetCh3(O3:Integer) then  
  Memo1.Lines.Add('Failed To Send.');
```

//Set Output4 //

```
if not MSUSBDevice1.D2A_SetCh4(O4:Integer) then  
  Memo1.Lines.Add('Failed To Send.');
```

//Set Output5 //

if not MSUSBDevice1.D2A_SetCh5(O5:Integer) then
Memo1.Lines.Add('Failed To Send.')

////////////////////////////////////

//Set Output6 //

if not MSUSBDevice1.D2A_SetCh6(O6:Integer) then
Memo1.Lines.Add('Failed To Send.')

////////////////////////////////////

//Set Output7 //

if not MSUSBDevice1.D2A_SetCh7(O7:Integer) then
Memo1.Lines.Add('Failed To Send.')

////////////////////////////////////

//Set Output8 //

if not MSUSBDevice1.D2A_SetCh8(O8:Integer) then
Memo1.Lines.Add('Failed To Send.')

////////////////////////////////////

// Change_USB_ID //

if not MSUSBDevice1.Change_USB_ID(USB_ID:string[4])then
Memo1.Lines.Add('Failed To Send.');

////////////////////////////////////