ADwin

LabVIEW Driver

for ***ADwin*** measurement data acquisition boards

Version 1.1

November 1996

Measurement data acquisition with *ADwin* boards in LabVIEW

1. Installing the ADwin driver

Please insert the ***ADwin-CD*** into the CD-ROM drive of your PC, the installation programm starts automatically. It is recommended install the driver to the directory C:\ADBASIC3 (if this has not been done during the instalation of ADbasic).

ADWIN4.BTL	driver program for the T400 processor
ADWIN5.BTL	driver program for the T450 processor
ADWIN8.BTL	driver program for the T805 processor
TESTVE16.EXE	displays the version of the 16-bit ADwin DLLs installed on your system
TESTVE32.EXE	displays the version of the 32-bit ADwin DLLs installed on your system
ADTEST.EXE	Windows program for testing the ***ADwin*** PC plugin boards
ADWINSET.EXE	program for registering the link address (only for Windows NT)

Moreover, according to the operating system in use, the following DLLs (Dynamic Link Libraries) are copied to the Windows directory:

Windows 3.X	**Windows 95**	**Windows NT**
ADWIN.DLL	ADWIN.DLL	ADWIN.DLL
ADWIN32.DLL	ADWIN32.DLL	ADWIN32.DLL
ADPOINT.DLL	ADPOINT.DLL	ADPOINT.DLL
	ADWIN95.DLL	ADWINNT.DLL

If using *Windows NT* you must have administraor access and you have to restart the computer after having installed the driver.

2. Loading the driver

Only after loading the ***ADwin*** driver will the computer be able to communicate with the ***ADwin*** board. The process of loading clears the whole memory of the ***ADwin*** board so that all program codes on the card are lost. The driver is loaded in Labview or in ADbasic.

With the function VI: **Adboot.VI** from your LabVIEW program. Depending on the type and version of the ***ADwin*** board the following parameters should be used for the inputs `dateiname` and `iboardsize`, respectively:

ADwin-type	*dateiname*
ADwin-4	*adwin4.btl*
ADwin-5	*adwin5.btl*
ADwin-8	*adwin8.btl*
ADwin-9	*adwin9.btl*

version	*iboardsize*
1 MB	*100000*
2 MB	*200000*
4 MB	*400000*
8 MB	*800000*
16 MB	*1000000*
32 MB	*2000000*

3. Calling of measurement functions

Note! These functions are only available for ADwin-4, -5, -8 PC boards

In LabVIEW programs VI instructions help to send commands and data to the ***ADwin*** board as well as to pick up data from the card. In this case the Vls are using functions which are included in the DLLs. With regards to 16-bit systems (Windows 3.1x) it is **ADwin.dll**, with regards to 32-bit systems (Windows 95, Windows NT) it is **ADwin32.dll**. In order to access the DLLs under LabVIEW you have to make the following changes before starting the program first:

Important !

Please**, copy** the DLL used by your program (ADwin.dll or ADwin32.dll) to another directory. **Rename** it **ADview.dll.** Then **move** this file (ADview.dll) back to your Windows directory. Now two DLL-files with the same contents but with different names do exist.

Attention !

After switching on the computer or respectively before starting your measurement tasks the ***ADwin*** driver has always to be loaded according to one of the instructions described under paragraph 2. Otherwise your computer will not be able to communicate with the ***ADwin*** board!

4. LabVIEW VIs

All LabVIEW Vls which have access to the ***ADwin*** board call the functions of the ADview.dll file. The following functions are available:

4.1 ADbasic-functions

4.1.1 ADboot.VI

Booting of the ADwin board with the file ADwin8.btl (default). If using different ADwin board you have to set the input "file name" to the correct value (e.g. ADwin4.btl).

Note !

When opening the Vls you will get more information about the settings.

4.1.2 ADbload.VI

This Vl loads the binary file (e.g. *.t81) of an ADbasic-program to the ***ADwin*** board.

4.1.3 ADbstart.VI

This Vl starts an ADbasic process already loaded.

4.1.4 ADbstop.VI

This Vl stops a running ADbasic process.

4.1.5 Dout_bit.VI (for ADwin-4, -5, -8 PC boards only)

This Vl sets the digital outputs of the ***ADwin*** board. The data input is made in **binary code** (0=clear, 1=set, for 16 bit [15.....0]).

4.1.6 Dout_val.VI (for ADwin-4, -5, -8 PC boards only)

This Vl sets the digital outputs of the ***ADwin*** board. The data input is made in **decimal code** (but evaluation is made in binary code).

4.1.7 SetPar_i.VI

The SetPar_i.VI gives a **long integer** value to an ADbasic parameter variable (Par_1...Par_80).

4.1.8 SetPar_f.VI

The SetPar_i.VI gives a **floating point** value to an ADbasic float parameter. (FPar_1...FPar_80).

4.1.9 GetPar_i.VI

This VI reads out one of the parameter variables (Par_1...Par_80). The return value is a **long integer** type.

4.1.10 GetPar_f.VI

This VI reads out one of the parameter variables (FPar_1...FPar_80). The return value is a **floating point** type.

Attention !

This function is possibly **not** available under Windows 3.1x.

4.1.11 SetiData.VI

This VI gives an array of **integer** values to one of the ADbasic data arrays (data_1...data_200).

4.1.12 SetfData.VI

This VI gives an array of **floating point** values to one of the data arrays (data_1...data_200).

4.1.13 GetiData.VI

The GetiData.VI reads out the values of an ADbasic **integer** data array (data_1 data_200).

4.1.14 GetfData.VI

The GetfData.VI reads out the values of an ADbasic **floating point** data array (data_1 data_200).

4.1.15 Get_adc.VI (for ADwin-4, -5, -8 PC boards only)

This VI reads out an analog input of the ***ADwin*** board. The return value is an **integer** type. The conversion will be made in ADbasic.

4.1.16 Set_dac.VI (for ADwin-4, -5, -8 PC boards only)

This VI sets an analog output to an **integer** value.

4.1.17 GetDigin.VI (for ADwin-4, -5, -8 PC boards only)

This VI returns the momentary status of the digital inputs.The return value is a **long integer** type in decimal code.

4.1.18 Workload.VI

This VI returns the workload (in percent) of the ***ADwin*** board processor.

4.1.19 Memory.VI

The memory.VI displays the free memory (in byte) of the ***ADwin*** board.

4.2 Measurement process (for ADwin-4, -5, -8 PC boards only)

The access to the **analog inputs** (ADC 1-12) with Vls is described in this paragraph.

4.2.1 Acquire.VI

The Acquire.VI carries out a series of A/D-conversions on one or more analog input channels with a fixed clock rate. With this Vl all 12 analog input channels can be measured cyclically.

4.3 Process for analog output

The Vls described in this paragraph are suitable for direct access to the analog outputs (DAC 1-6) and for a cyclic output of the values. It is possible to run two different processes with different clock rates simultaneously on the ***ADwin*** board**.**

4.3.1 StartDAC.VI

The StartDAC.VI starts the cyclic output with a specified output rate on up to 6 analog channels simultaneously.

4.3.2 StopDAC.VI

This Vl stops an already started but not yet finished output process prematurely.

4.4 Digital PID-controller

The following Vls describe the usage of a digital PID-controller. The PID-controller ist not programmed in ADbasic. (Note: In this case the opportunities to register actuating values or control deviations are limited by a buffer). Up to four control processes can run simultaneously.

4.4.1 Conf_PID.VI

The Conf_PID.VI (configure PID-controller) has to be carried out before starting the controlling task for every process. It sets the input for measurements, the output channel, and the size of the

buffer where the measurement values are registered. The buffer contains the actuating values and the control deviations which are registered after the start.

4.4.2 StartPID.VI

This Vl starts one of the control processes. At the same time the buffer will be cleared and reset.

4.4.3 StopPID.VI

Stops the control process.

4.4.4 Set_PID.VI

This Vl changes the settings (k, To,Td, Ti) of a control process without stopping it.

4.4.5 Get_PID.VI

Reads out the buffer contents with the actuating values and the control deviations which are registered after start of the controller.

4.5 Support

Because some of the input/output Vls need values in raw data format (integer values from 0-4095), there are two Vls which convert volt into value or value into volt, respectively. The inputs are meant for discrete values, arrays and 2 dimensional arrays and are used according to the requirements. (Note: If processes need a longer time to be carried out in some cases, the suitable conversion sequences can easily be copied from the Vls and used directly in the program).

4.5.1 V_to_Val.VI

Conversion of volt to values (0 - 4096), depending on the range set on the ***ADwin*** board (0...+10V, -5...+5V, -10...+10V).

4.5.2 Val_to_V.VI

Conversion of values (0 - 4096) into volt values, depending on the range set on the ***ADwin*** board (0...+10V, -5...+5V, -10...+10V).

Note !

If one of your VI's cannot be carried out (to be seen at the cursor which will appear interrupted) the VI does probably not find the DLL. In this case, please check first if the **ADview.dll** can be found in your Windows directory. If so, please open the VI that has the error. Then doubleclick the DLL-function in the diagram. After this, a window opens so that editing will be possible. In the first line you will find at *„Library name or path"* the entry ADview.dll. Please enter here the complete path of the ADview.dll (e.g. C:\Windows\ADview.dll) and save the VI with this setting. It should now work properly.

4. Sample program „Bas_dmo1.VI"

The Bas_dmo1.VI uses the binary file of the BAS_DMO1 of your ADbasic sample program. Please, make sure that the name of the binary file, its path as well as the process number correspond to the parameter used in the program, so that your program works properly. If necessary adapt these parameters to the Lab VIEW program.

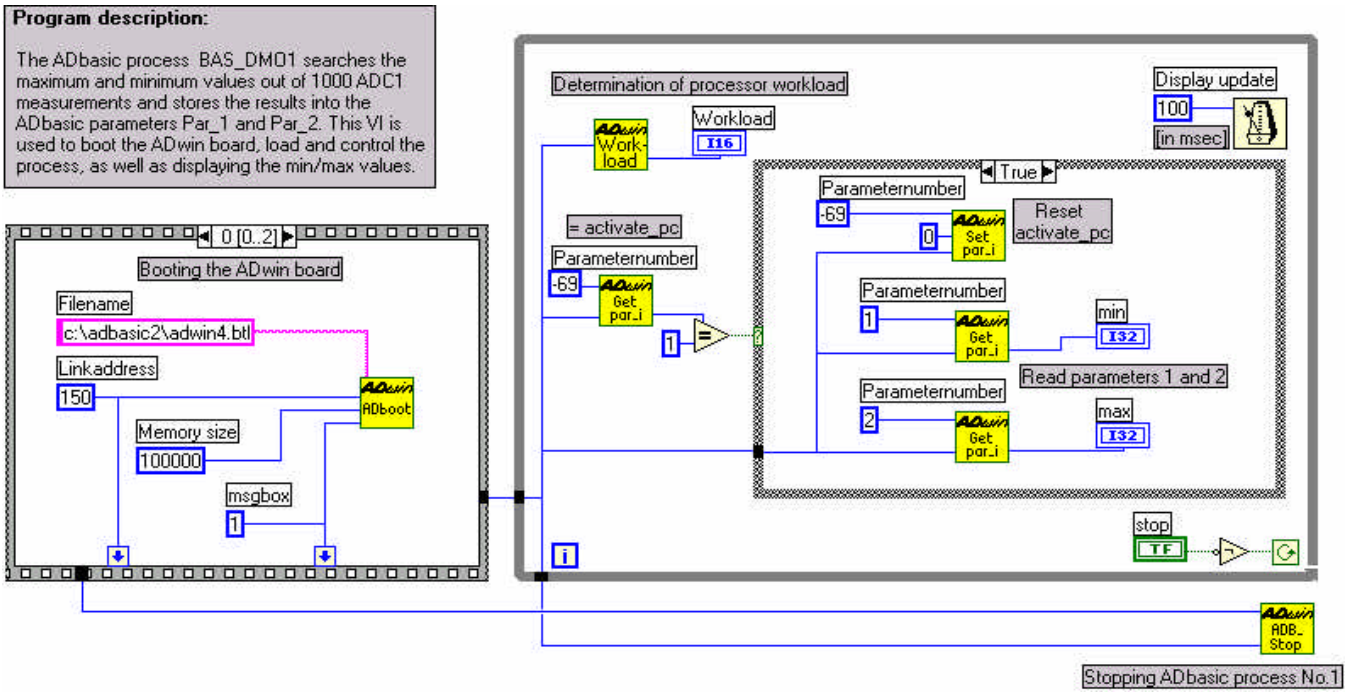
4.1 Program structures

On the following page you will find an illustration of the icon, the front-panel as well as of the diagram.The program includes a boot process so that you do not have to load the file ADwin4.btl to your ***ADwin*** board**.**

Icon:

Front-Panel:

Diagram:


Program description:
The ADbasic process BAS_DMO1 searches the maximum and minimum values out of 1000 ADC1 measurements and stores the results into the ADbasic parameters Par_1 and Par_2. This VI is used to boot the ADwin board, load and control the process, as well as displaying the min/max values.
0 [0..2]
Booting the ADwin board
Filename
c:\adbasic2\adwin4.btl
Link.address
150
Memory size
100000
msgbox
1
ADboot
Determination of processor workload
Workload
I16
Work-load
= activate_pc
Parameternumber
-69
Get par_i
1
Display update
100
[in msec]
True
Reset activate_pc
0
Set par_i
Parameternumber
1
min
I32
Read parameters 1 and 2
Parameternumber
2
max
I32
stop
TF
i
ADB_ Stop
Stopping ADbasic process No.1
1 [0..2]
Loading ADbasic process no. 1
Filename
c:\adbasic2\samples\bas_dmo1.t41
ADB_ Load
2 [0..2]
Starting ADbasic process no.1
Processnumber
1
ADB_ Start