Innovative Integration

ADC64 DASYLab Driver Software Supplement

First Edition

The ADC64 Developer's Package Software Manual was prepared by the technical staff of Innovative Integration, September 1996.

For further assistance contact

Innovative Integration

31352 Via Colinas #101

Westlake Village, CA 91362

PHONE:	(818) 865-6150
FAX:	(818) 879-1770
BBS:	(818) 879-1729
email:	techsprt@innovative-dsp.com
FTP:	ftp.innovative-dsp.com
WWW:	http://www.innovative-dsp.com

Document: \\p60\winword\manuals\adc64\software\adc64dsy.doc

Table of Contents

List of Figures

1. Introduction

This document describes the driver required to support the Innovative Integration (I.I.) ADC64 PCI bus digital signal processor (DSP) card in the DASYTEC DASYLab software environment . DASYLab provides an easy to use yet powerful environment for data acquisition, and this driver allows the use of the full power of the ADC64 in this environment.

The ADC64 DASYLab driver consists of several components:

- the DASYLab DLL File (`adc64dsy.dll`)
- the DASYLab Help File (`adc64dsy.hlp`)
- the ADC64 Target COFF File (`adc64dsy.out`)
- the ADC64 driver INI file (`adc64.ini`)
- the ADC64 Windows 95 device driver installation disk
- a program to detect ADC64 configuration settings (`adc64.exe`)
- sample DASYLab worksheets

This manual discusses installation issues and features specific to the I.I. ADC64 DASYLab driver. Please see the DASYLab documentation for information about the DASYLab application or for information about installing DASYLab.

Installation of the II ADC64 DASYLab driver is discussed first, followed by a description of configuring the driver for use with the ADC64. This is followed by notes on usage on the ADC64 under DASYLab and a description of the sample worksheets.

1.1 A Note about this Manual

Certain typography conventions are used in this manual to indicate user operations, file types, etc., as follows:

- Windows 95 application menu commands are identified and presented as pipe-delimited strings indicating the menu entries which are being discussed. For example, the Load Program menu item under the File menu in the Code Composer package would be named by the following string:

  File | Load Program

- Computer readable files and keyboard input/output are represented in Courier font, with user input in bold. For example, a program file will be referred to by name as

  `C:\ADC64\TALKER\TALKER.OUT`

  while user input and commands look like

  `ROM MYPROG.OUT`

2. Installation

Installation of the ADC64 DASYLab driver consists of both hardware and software installation procedures. This document describes the software installation process and details the features of the driver. Refer to the *ADC64 Hardware Manual* for a discussion of ADC64 hardware installation and configuration. It also assumes that the DASYLab software package is already installed on the system.

This document is intended to augment, not replace, the documentation provided with the DASYLab software package. Refer to the documentation provided with DASYLab for a complete discussion of their features and use.

2.1 Hardware and Software Requirements

2.1.1 Host Hardware Requirements

The DASYLab driver requires an IBM or 100% compatible 386 or higher machine for proper operation (Pentium-class machines are highly recommended). The host must have at least 16 Mbytes of memory and a 3.5" floppy drive.. Windows 3.1 is required for DASYLab, and Windows 95 is suggested.

2.1.2 Target Hardware Requirements

The DASYLab driver requires an ADC64 board with 128 KB of RAM installed. Any of the speed options are acceptable, but the faster CPUs will give better performance in data acquisition.

2.1.3 DASYLab Requirements

This driver requires DASYLab Version 3.0.

2.2 Software Installation

2.2.1 Loading the ADC64 Device Driver

2.2.1.1 Windows 95

After installing the ADC64 card in the system as described in the *ADC64 Hardware Manual*, the Windows 95 Device Driver should be loaded. Reboot the PC into Windows '95. Windows 95 will automatically detect the ADC64 and display:

"New Hardware Found - PCI Card"

On the display screen, choose the radio button

"Driver from disk provided by hardware manufacturer"

Then click on "OK"

Insert the ADC64 Windows 95 Device Driver Install disk and press "OK".

Restart your computer when prompted to do so.

To verify correct installation, check under the Device Manager. Right-click on My Computer and select Properties. Then select the Device Manager tab. Select the ADC64 Supercontroller category.

The device status should report "This device is working properly."

Under the resources tab, the Interrupt Request and I/O Range should be displayed. Under the Conflicting device category, the list should state "No conflicts".

Provided that all is well, the DASYLab driver can now be loaded.

2.2.1.2 Windows 3.1

In Windows 3.1, no board device driver needs to be loaded. Since the Windows 95 device manager is not available to display the PCI Plug and Play I/O Port and IRQ assignments, a separate program is provided on the Device Driver disk, ADC64.EXE, to display the settings.

2.2.2 Loading the DASYLab Device Driver

Installation of the DASYLab driver is a simple operation. An installation program, **SETUP.EXE**, is provided to load the driver.

If for some reason the installation program is unsuitable, a description of how to manually install the device driver follows.

The three files that compose the driver need to be copied from the distribution disk to the DASYLab home directory by entering:

```
COPY A:ADC64DSY.DLL C:\DLAB3
COPY A:ADC64.INI    C:\DLAB3
COPY A:ADC64DSY.OUT C:\DLAB3
```

The sample worksheets are kept in a subdirectory on the distribution disk set, and are to be loaded into the DASYLab worksheet directory:

```
COPY A:\WORKSHEE\*.* C:\DLAB3\WORKSHEE
```

The above commands assume that DASYLab is installed in its default location of `C:\DLAB3.` If the program is installed in another directory, substitute the directory for `C:\DLAB3` above.

2.2.3 Editing the `ADC64.INI` File

When the DASYLab home directory is not `C:\DLAB3`, `ADC64.INI` must be edited for the proper operation of the driver. This needs to be done since the driver loads a program to the DSP on the ADC64 board on every experiment run. The absolute path name of the program COFF file (i.e., the full name with drive and path included) is saved in the INI file for the driver's use. This needs to always refer to the file `ADC64DSY.OUT` in the DASYLab home directory.

Using Notepad or another text editor, open `ADC64DSY.OUT` and change the value of the file name in the Common Section "COFF File" Key to be the current path. Below is an illustration of what this section of the INI File should look like. See the Appendix I for more details on the INI File format.

```
;
;   ADC64 DASYLab Configuration File
;
[Common]
COFF File=c:\dlab3\adc64dsy.out

[ADC64 Target 0]
...
<< rest of file >>
...
```

3. Configuring the ADC64 DASYLab Driver

3.1 Loading the Driver into DASYLab

Before configuring the driver can begin, DASYLab needs to load the ADC64 driver as its working driver. From the DASYLab main menu select:

Experiment | Select Driver...

When the dialog box appears, select the ADC64 driver, file name `ADC64DSY.DLL.`

3.2 Opening the Hardware Setup Dialog

The configuration of the ADC64 driver and board is performed by a dialog box that is called by DASYLab. To post this dialog box the user can either select

Experiment | Hardware Setup

or select the toolbar button with the circuit-board icon.

3.3 The Hardware Setup Dialog

The purpose of the Hardware Setup dialog is to describe to the ADC64 Driver the state of the ADC64 board. With this information the driver can inform DASYLab of the data it requires and communicate with the board to send and receive data.

Because the results of this dialog are the only source of information about the state of these parameters, it is essential that the values in this box exactly duplicate the true settings of the board or unpredictable results will occur. It is recommended that the settings be programmed in the dialog first, and then the hardware set to match by powering down the system and removing the board.

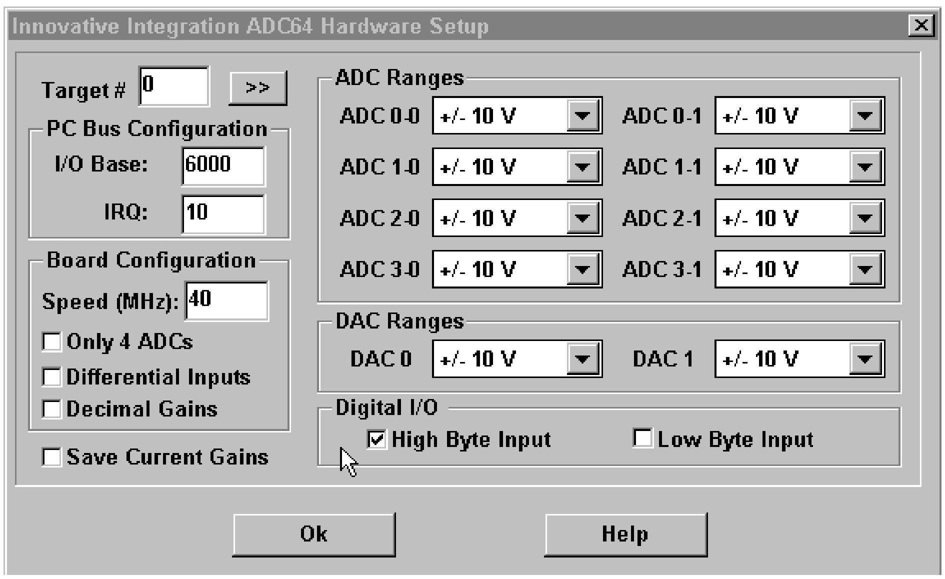

Figure 1: ADC64 DASYLab Hardware Setup Dialog

Figure 1 shows the actual dialog with some sample values. The following sections describe in detail the setting of each section of this dialog box

3.3.1 Target

Normally, only one ADC64 is in use. The Target Specifier allows configurations for multiple cards or for multiple configurations of a single ADC64 card to be saved in the ADC64.INI file. These multiple configurations are referred to as Target #s.

The Initial Target # that the driver uses is Target 0.. This configuration is loaded when DASYLab first starts up. Other Target configurations are loaded by altering the tag string in the edit control and pressing the adjacent button marked with the “>>“ string.. The configuration indicated is loaded into the Setup Dialog for viewing and editing..

Any three letter tag string can be used to distinguish a setup definition.

3.3.2 PC Bus Configuration

This section defines the location of the ADC64 on the PC Bus. If these values are not accurate, the driver will not be able to communicate with the card and will not run.

On Windows 95, the values for these entries can be found by looking at the resources used by the installed ADC64 device driver by using the Control Panel's System applet. Select the Device Manager Tab, Open the entry labeled "Other" in the tree and select the ADC64 Supercontroller Driver.

Next, in the new window, select the Resources Tab and read off the IRQ and IO Base address.

For Windows 3.1 users, the program ADC64.EXE can be run to obtain the IRQ and Port Addresses. Enter

```
ADC64
```

which displays a window showing the information. If the system has multiple ADC64 boards, enter

```
ADC64 -T <target #>
```

where "target #" ranges from 0 to the number of boards installed in the system minus one.

The current driver does not actually use the interrupt IRQ value.

3.3.3 Board Configuration

The Board Configuration Section contains several switches that define which optional components of the ADC64 are installed on the target.

3.3.3.1 Speed

The speed entry holds the CPU speed of the board in megahertz.

The current driver does not use this value.

3.3.3.2 Only 4 ADCs

Check this box if the ADC64 was installed with only 4 ADCs on the board. Leave unchecked if all 8 ADCs are installed.

3.3.3.3 Differential Inputs

Check this box if the ADCs are set for differential input. Leave unchecked if the ADCs are set for single-ended inputs.

Note that the driver does not support mixing single ended and differential on different ADCs. All ADCs must be set for either single ended or differential operation.

3.3.3.4 Decimal Gains

Check this box if the optional Gain Amplifiers giving gains of 1, 2, 5, and 10 are installed. Leave the box unchecked if the standard gains of 1, 2, 4, and 8 are installed.

3.3.4 ADC Ranges

These combo boxes allow the selection of ADC ranges for each of the 8 ADCs on the board. See Section 4 for mapping the DASYLab ADC numbering scheme to that used in the *ADC64 Hardware Manual*.

Each ADC may have its own range selected by jumper on the card. The driver will support separate ranges for each ADC.

3.3.5 DAC Ranges

These combo boxes allow the selection of DAC ranges for each of the 2 DACs on the board.

Each DAC may have its own range selected by jumper on the card. The driver will support separate ranges for each DAC.

3.3.6 Digital I/O

Digital I/O can be configured by software to allow either half of a 16 bit register to be separately configured as an output or an input. Checking the box for each byte selects it as an input.

Be sure that if you remove all channels of Digital Input, there are Digital Input blocks placed on the current worksheet. Doing so can produce errors in the worksheet. In addition, restoring a saved worksheet using Digital input channels no longer available should also be avoided.

The same situation exists for Digital Output, if all channels are configured to be input.

3.3.7 Save Current Gains

When a worksheet is saved to disk, the gain settings applied are stored with it. Even so, if the standard configuration used for an experimental setup has many non-default gain settings, it is still a tiresome task to continually set each of the 64 gains to the same value.

The Save Current Gains option saves the current gain settings with the configuration in the ADC64.INI. When this configuration is loaded, the gains in it will also be loaded as the initial gain settings for a new worksheet.

4. Driver User Notes

The ADC64 DASYLab driver provides a number of block modules for use in worksheets:

- 4 or 8 ADC blocks with 4 or 8 multiplexed inputs each
- 1 DAC block with 2 available channels
- 1 Digital In block with 8 or 16 lines
- 1 Digital Out block with 8 or 16 lines

The variable number of ADC channels and Digital channels must be set at configuration time to match the hardware on the board.

4.1 ADC and Channel Mapping

The ADC64 differs from many data acquisition boards in that it has multiple ADCs on the board. In order to handle this situation smoothly in DASYLab, each ADC on the board is given a separate module block with up to eight channels available for use.

To support the board configuration that has only 4 ADCs smoothly, the first four modules dropped on the worksheet correspond to the even ADC numbers as used in the ADC64 board naming. This means that there are different numbering orders for a channel in DASYLab's scheme and for a channel in the *ADC64 Hardware Manual*. Figure 2 describes the correspondence between the two sets of names.

DASYLab ADC Name	DASYLab Channel Number	A/D Reference Designator	Innovative ADC Number	Innovative Channel Number
ADC 0-0	0-7	U17	0	0-7
ADC 1-0	8-15	U41	2	16-23
ADC 2-0	16-23	U30	4	32-39
ADC 3-0	24-31	U51	6	48-55
ADC 0-1	32-39	U18	1	8-15
ADC 1-1	40-47	U42	3	24-31
ADC 2-1	48-55	U31	5	40-47
ADC 3-1	56-63	U52	7	56-63

Figure 2: DASYLab ADC Mapping to ADC64 ADCs

This table will be useful during the configuration of the ADC64 when changing the Range Selection jumpers for each ADC. The DASYLab configuration dialog displays the DASYLab name, while the ADC64 board documentation shows the ADC number.

Another time when this difference comes into play is when designing the input connector for the experiment. To aid in this, Figure 3 duplicates the connector pinout table from the *ADC64 Hardware Manual*, but with the DASYLab channel numbers substituted for the Innovative channels.

Pin	Signal Name	Description	Pin	Signal Name	Description
1	-12V	-12V power from PC bus	51	+12V	+12V from PC bus
2	TCLK1	Timer/counter I/O from DSP	52	TCLK0	Timer/counter 0 on DSP
3	EXT TRIG3	External trigger input 3	53	EXT TRIG2	External Trigger Input 2
4	EXT TRIG1	External trigger input 1	54	EXT TRIG0	External Trigger Input 0
5	GATE5*	Gate to 82C54 timer counter 5	55	-	Unused
6	TMR5	Output of Timer 5	56	TMR_CLK5	Clock input to 82C54 timer counter 5
7	DX15	Digital I/0 bit 15	57	DX14	Digital I/0 bit 14
8	DX13	Digital I/0 bit 13	58	DX12	Digital I/0 bit 12
9	DX11	Digital I/0 bit 11	59	DX10	Digital I/0 bit 10
10	DX9	Digital I/0 bit 9	60	DX8	Digital I/0 bit 8
11	DX7	Digital I/0 bit 7	61	DX6	Digital I/0 bit 6
12	DX5	Digital I/0 bit 5	62	DX4	Digital I/0 bit 4
13	DX3	Digital I/0 bit 3	63	DX2	Digital I/0 bit 2
14	DX1	Digital I/0 bit 1	64	DX0	Digital I/0 bit 0
15	DVCC	+5V from PC bus	65	DGND	Digital Ground
16	-	Unused	66	-	Unused
17	AGND	Analog Ground	67	DAC0	Output from D/A channel 0
18	AGND	Analog Ground	68	DAC1	Output from D/A channel 1
19	IN1	ADC 0-0 Channel 1	69	IN0	ADC 0-0 Channel 0
20	IN3	ADC 0-0 Channel 3	70	IN2	ADC 0-0 Channel 2
21	IN5	ADC 0-0 Channel 5	71	IN4	ADC 0-0 Channel 4
22	IN7	ADC 0-0 Channel 7	72	IN6	ADC 0-0 Channel 6
23	IN9	ADC 0-1 Channel 1	73	IN8	ADC 0-1 Channel 0
24	IN11	ADC 0-1 Channel 3	74	IN10	ADC 0-1 Channel 2
25	IN13	ADC 0-1 Channel 5	75	IN12	ADC 0-1 Channel 4
26	IN15	ADC 0-1 Channel 7	76	IN14	ADC 0-1 Channel 6
27	IN17	ADC 1-0 Channel 1	77	IN16	ADC 1-0 Channel 0
28	IN19	ADC 1-0 Channel 3	78	IN18	ADC 1-0 Channel 2
29	IN21	ADC 1-0 Channel 5	79	IN20	ADC 1-0 Channel 4
30	IN23	ADC 1-0 Channel 7	80	IN22	ADC 1-0 Channel 6
31	IN25	ADC 1-1 Channel 1	81	IN24	ADC 1-1 Channel 0
32	IN27	ADC 1-1 Channel 3	82	IN26	ADC 1-1 Channel 2
33	IN29	ADC 1-1 Channel 5	83	IN28	ADC 1-1 Channel 4
34	IN31	ADC 1-1 Channel 7	84	IN30	ADC 1-1 Channel 6
35	IN33	ADC 2-0 Channel 1	85	IN32	ADC 2-0 Channel 0
36	IN35	ADC 2-0 Channel 3	86	IN34	ADC 2-0 Channel 2
36	IN37	ADC 2-0 Channel 5	87	IN36	ADC 2-0 Channel 4
38	IN39	ADC 2-0 Channel 7	88	IN38	ADC 2-0 Channel 6
39	IN41	ADC 2-1 Channel 1	89	IN40	ADC 2-1 Channel 0
40	IN43	ADC 2-1 Channel 3	90	IN42	ADC 2-1 Channel 2
41	IN45	ADC 2-1 Channel 5	91	IN44	ADC 2-1 Channel 4
42	IN47	ADC 2-1 Channel 7	92	IN46	ADC 2-1 Channel 6
43	IN49	ADC 3-0 Channel 1	93	IN48	ADC 3-0 Channel 0
44	IN51	ADC 3-0 Channel 3	94	IN50	ADC 3-0 Channel 2
45	IN53	ADC 3-0 Channel 5	95	IN52	ADC 3-0 Channel 4
46	IN55	ADC 3-0 Channel 7	96	IN54	ADC 3-0 Channel 6
47	IN57	ADC 3-1 Channel 1	97	IN56	ADC 3-1 Channel 0
48	IN59	ADC 3-1 Channel 3	98	IN58	ADC 3-1 Channel 2
49	IN61	ADC 3-1 Channel 5	99	IN60	ADC 3-1 Channel 4
50	IN63	ADC 3-1 Channel 7	100	IN62	ADC 3-1 Channel 6

Figure 3: Table of P1 pins, with DASYLab Driver channel equivalents

4.2 Analog to Digital Converters Modules

4.2.1 Configuration

The number and distribution of channels on a block of ADCs are dependent on the choices made in the Configuration Dialog box. If the selection of only 4 ADCs is made, the last 4 ADCs will not be released onto the worksheet. If the Differential Option is selected, each ADC only will have 4 available channels. The range for each ADC must also be set to match the board configuration for proper operation.

Gains for each channel available may be set as DASYLab normally allows..

4.2.2 Channel Selection and Acquisition Rate

Each of the ADCs on the ADC64 has a maximum clock rate of 100 kHz. If multiple channels are to be multiplexed off a single ADC, the effective maximum rate is divided by the number of channels read on the ADC. For example, if 4 channels are multiplexed on a single ADC:

Theoretical Maximum Rate = 100 kHz / 4 channels = 25 kHz.

However, on the ADC64, there are multiple ADCs. If you take these same 4 inputs and route them to 4 different ADCs, the maximum rate remains at the full 100 kHz.

In DASYLab worksheet terms, this is equivalent to the difference between these 2 configurations:

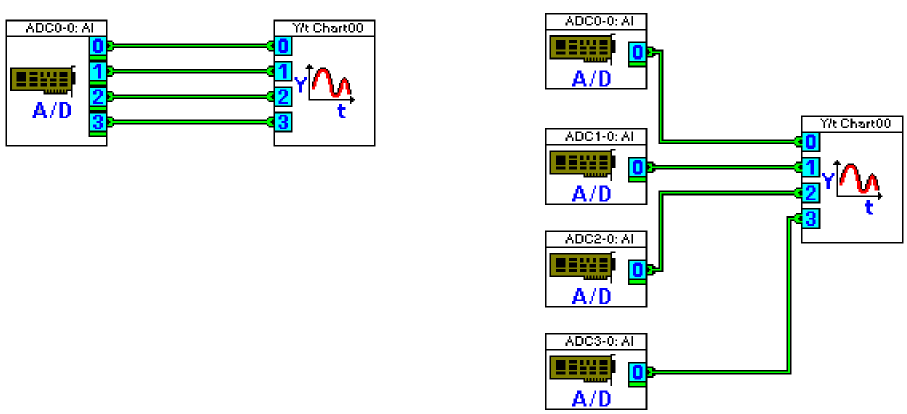

Figure 4: Multiplexed and Parallel Configurations

The configuration on the left will multiplex the four channels on a single ADC, thus limiting the rate to at most 25 kHz. The configuration on the right will acquire from all four ADCs in parallel, allowing the full rate to be achieved.

In the case where a mix of multiplexed and parallel channels are used, the driver attempts to read the data as close to simultaneously as it can. Consider a case where 5 channels are read off of 1 ADC, 2 off of a second, and 1 off a third. On the first clock, the first channel is read from all 3 ADCs in parallel. The second clock reads the second line from the first 2 ADCs, and the remaining 3 clocks read only from the first ADC. The maximum rate achievable is 100 kHz divided by the "fullest" ADC's count, or 20 kHz.

The order the channels are multiplexed on an ADC is in strict channel order. That is, if channels 2, 5, and 7 on an ADC are read out, channel 2 is read on the first clock, followed by 5 and 7 on the next 2 clocks..

4.3 Digital to Analog Converter Modules

The DAC module supports a channel for each DAC on the ADC64. Each can have its range individually set in the configuration. These DACs work in both synchronous and asynchronous modes.

The asynchronous mode sends 1 point to the ADC64 at a time. The overhead for this transfer limits this mode to very low rates. Asynchronous mode should be used when only low rates are required or when maximum ADC performance is required.

In synchronous mode, the DAC rate is matched to the rate at which data is acquired, so that 1 DAC data point is sent for each event. DASYLab allows you to select a rate even lower, sending DAC data for every N events. This capability is not supported by the driver, which will always send 1 DAC data point for each event read back in.

4.4 Digital Input and Output Modules

The ADC64 board has a single 16 bit Digital I/O port, which can be configured to allow either byte to act as input or output. The driver allows any of the 4 arrangements to be set in its configuration dialog.

Digital Input is processes like the analog input, while Digital Output is handled similarly to the way the DACs are handled.

As with the DACs, the Digital Out can be processed synchronously or asynchronously. Asynchronous mode is very limited as to the rates that can be supported. Synchronous output is slaved to the Analog Input rate, and is currently limited to exactly that rate.

4.5 Performance Issues

4.5.1 Maximizing Data Rates

Data rate is maximized by careful planning your experiment to partition fast and slow events. Inputs should be spread over as many ADCs as possible. DACs and Digital Output should be run asynchronously if they are not critically timed.

The DSP on the ADC64 is responsible for acquiring the data from the ADCs, packetizing it, and transmitting it back to the host computer. A timer driven interrupt controls the input rate, while the remaining CPU cycles are dedicated to packetizing and transferring packets to the DASYLab host.

The more work that DSP interrupt handler needs to do, the less time is left to transmit the data. This means that the system will tend to lose ground against the input at a lower rate, and thus drop data in continuous mode. In fact, if all the CPU time is taken up in interrupt processing, the system may appear to freeze, as there are no cycles to process and transmit the acquired data.

The DSP interrupt handler has four components: the Analog input, Digital input, Digital or Analog output, and Multiplexing. If the experiment consists only of analog input and is spread across ADCs so that no multiplexing is required, the interrupt response handler can run at its fastest speed. Adding non-analog input or any output will reduce the throughput more, and an experiment requiring multiplexing the analog input will reduce the rate even further.

Of course, the more data that an experiment requires to be transferred, in any arrangement, the lower the maximum sustainable rate will be.

DASYLab allows burst acquisition of data which might allow higher rates than the continuous model for short periods, but if the buffers on the ADC64 are exhausted in the burst, data may be lost even in this mode.

4.5.2 Benchmarks

All benchmarks were taken with an ADC64 running at 50 MHz, on a Pentium 120 MHz host machine running Windows 95. The DASYLab buffers were 32k in size. These are the maximum speeds before dropped data was evident:

- One ADC Channel, 8k block size: 100 kHz.
- Two channels, non-multiplexed, 4k block size: 90 kHz.
- Four channels, non-multiplexed, 2k block size: 57.5 kHz
- Four channels, multiplexed on 1 ADC: 19.5 kHz. (Interrupt rate: 78 kHz)
- Analog out to Analog in, 1 channel: 47.5 kHz.

5. Sample Worksheets for the ADC64 Driver

The following section details the sample worksheets included with the DASYLab driver. These worksheets are provided as models for custom user software, and it is highly recommended that the user examine these examples before beginning a first development effort for the target.

5.1 Analog Input Benchmark Worksheets

5.1.1 AI_TEST

AI_TEST is a very simple worksheet that has 1 analog input going to a graphing module. A separator module discards most of the data before display to avoid allow the maximum throughput.

5.1.2 AI_TEST2

This worksheet is similar to AI_TEST, but has 2 unmultiplexed analog inputs.

5.1.3 AI_TEST4

This worksheet is similar to AI_TEST, but has 4 unmultiplexed analog inputs.

5.1.4 AI_TEST41

This worksheet demonstrates the performance of 4 multiplexed analog inputs.

5.2 Other Benchmark Worksheets

5.2.1 AO_IO

This worksheet demonstrates the DAC output, which is then read back by the ADC as input. The output of DAC 0 must be looped back to ADC Channel 0 to complete the test.

5.2.2 DO_IO

This worksheet demonstrates digital output looped back to digital input. The high and low bytes of digital input need to be connected together for this test.

6. Appendix I: `ADC64.INI`

The following is a sample ADC64.INI, with 2 parallel sets of settings. Note that Target 0 has had the gains saved in it. Target 1 has never had the gains saved.

```
;
;   ADC64 DASYLab Configuration File
;
[Common]
COFF File=c:\dlab3\adc64dsy.out

[ADC64 Target 0]
;
;   Fixed Board Settings
;
IRQ=10
IO Base=0x6000
MHZ=40

Decimal Gains=0
;
;  Range Jumper Settings
ADC 0 Range=0
ADC 1 Range=0
ADC 2 Range=0
ADC 3 Range=0
ADC 4 Range=0
ADC 5 Range=0
ADC 6 Range=0
ADC 7 Range=0

DAC 0 Range=0
DAC 1 Range=0
;
;   Default Configurable Settings
;
DIO_Lo_Read=0
DIO_Hi_Read=1
Analog Clock=250
;
;  ADC Gains
;
Channel 0 Gain=1
Channel 1 Gain=0
Channel 2 Gain=0
Channel 3 Gain=0
Channel 4 Gain=0
Channel 5 Gain=0
Channel 6 Gain=0
  <<  MANY OTHER GAIN SETTINGS >>
Channel 58 Gain=0
Channel 59 Gain=0
Channel 60 Gain=0
Channel 61 Gain=0
Channel 62 Gain=0
Channel 63 Gain=0
Differential Inputs=0
Only 4 ADCs=0
```

```
[ADC64 Target 1]
IRQ=10
IO Base=0x6000
MHZ=40
Decimal Gains=0
DIO_Lo_Read=0
DIO_Hi_Read=1
Analog Clock=100
ADC 0 Range=0
ADC 1 Range=0
ADC 2 Range=0
ADC 3 Range=0
ADC 4 Range=0
ADC 5 Range=0
ADC 6 Range=0
ADC 7 Range=0
DAC 0 Range=1
DAC 1 Range=0
Differential Inputs=0
Only 4 ADCs=1
```