COMPREHENSIVE SERVICES

We offer competitive repair and calibration services, as well as easily accessible documentation and free downloadable resources.

SELL YOUR SURPLUS

We buy new, used, decommissioned, and surplus parts from every NI series. We work out the best solution to suit your individual needs.

Sell For Cash Get Credit Receive a Trade-In Deal

OBSOLETE NI HARDWARE IN STOCK & READY TO SHIP

We stock **New, New Surplus, Refurbished,** and **Reconditioned** NI Hardware.

APEX WAVES

Bridging the gap *between the manufacturer and your legacy test system.*

1-800-915-6216

www.apexwaves.com

sales@apexwaves.com

Request a Quote CLICK HERE ***DAQCard-500***

DAQCard™-500
Register-Level Programmer Manual

Multifunction I/O Board for the PCMCIA Bus

July 1995 Edition

Part Number 340898A-01

National Instruments Corporate Headquarters
6504 Bridge Point Parkway
Austin, TX 78730-5039
(512) 794-0100
Technical support fax: (800) 328-2203
(512) 794-5678

Branch Offices:
Australia 03 9 879 9422, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Canada (Ontario) 519 622 9310,
Canada (Québec) 514 694 8521, Denmark 45 76 26 00, Finland 90 527 2321, France 1 48 14 24 24,
Germany 089 741 31 30, Hong Kong 2645 3186, Italy 02 48301892, Japan 03 5472 2970, Korea 02 596 7456,
Mexico 95 800 010 0793, Netherlands 03480 33466, Norway 32 84 84 00, Singapore 2265886, Spain 91 640 0085,
Swed 08 730 49 70, Swi nd 056 20 51 51, Tai 02 377 120 , U.K. 01635 523545

Limited Warranty

The DAQCard-500 is warranted against defects in materials and workmanship for a period of one year from the date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this manual is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Trademarks

LabVIEW®, NI-DAQ®, RTSI®, and DAQCard™ are trademarks of National Instruments Corporation.

Product and company names listed are trademarks or trade names of their respective companies.

WARNING REGARDING MEDICAL AND CLINICAL USE OF NATIONAL INSTRUMENTS PRODUCTS

National Instruments products are not designed with components and testing intended to ensure a level of reliability suitable for use in treatment and diagnosis of humans. Applications of National Instruments products involving medical or clinical treatment can create a potential for accidental injury caused by product failure, or by errors on the part of the user or application designer. Any use or application of National Instruments products for or involving medical or clinical treatment must be performed by properly trained and qualified medical personnel, and all traditional medical safeguards, equipment, and procedures that are appropriate in the particular situation to prevent serious injury or death should always continue to be used when National Instruments products are being used. National Instruments products are NOT intended to be a substitute for any form of established process, procedure, or equipment used to monitor or safeguard human health and safety in medical or clinical treatment.

Contents

Tables

About This Manual

This manual contains information about the internal operation and programming of the DAQCard-500. The DAQCard-500 is a low-cost, low-power analog input, digital, and timing I/O board for computers equipped with a Type II PCMCIA slot.

This manual assumes you are familiar with the *DAQCard-500 User Manual*. If you will be using National Instruments software with the DAQCard-500, you do not need to read this manual. For information on the DAQCard-500 installation, signal connections, and theory of operation, consult your user manual.

Organization of This Manual

The *DAQCard-500 Register-Level Programmer Manual* is organized as follows:

- Chapter 1, *General Description*, describes the general characteristics and gives a configuration overview of the DAQCard-500.
- Chapter 2, *Register Map and Descriptions*, describes in detail the address and function of each of the DAQCard-500 control and status registers.
- Chapter 3, *Programming*, contains programming instructions for operating the circuitry on the DAQCard-500.
- Appendix A, *MSM82C54 Data Sheet*, contains a manufacturer data sheet for the MSM82C54 CMOS programmable interval timer (OKI Semiconductor). This timer is used on the DAQCard-500.
- Appendix B, *Customer Communication*, has a form you can use to comment on the product documentation. This appendix also contains information on how to access technical assistance for your National Instruments product.
- The *Glossary* contains an alphabetical list and description of terms used in this manual, including abbreviations, acronyms, metric prefixes, mnemonics, and symbols.
- The *Index* contains an alphabetical list of key terms and topics covered in this manual, including the page where you can find each one.

Conventions Used in This Manual

The following conventions are used in this manual:

bold	Bold text denotes menus, menu items, or dialog box buttons or options.
italic	Italic text denotes emphasis, a cross reference, or an introduction to a key concept.
bold italic	Bold italic text denotes a note, caution, or warning.
`monospace`	Text in this font denotes text or characters that are to be literally input from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, variables, filenames, and extensions, and for statements and comments taken from program code.
< >	Angle brackets containing numbers separated by an ellipses represent a range of values associated with a bit, signal, or port (for example, ACH<0..7> stands for ACH0 through ACH7).
NI-DAQ	NI-DAQ refers to the NI-DAQ software for PC or Macintosh unless otherwise stated.

Abbreviations, acronyms, metric prefixes, mnemonics, and terms are listed in the *Glossary*.

National Instruments Documentation

The *DAQCard-500 Register-Level Programmer Manual* is one piece of the documentation set for your data acquisition (DAQ) system. You could have any of several types of manuals, depending on the hardware and software in your system. Use the different types of manuals you have as follows:

- *Getting Started with SCXI*—If you are using SCXI, this is the first manual you should read. It gives an overview of the SCXI system and contains the most commonly needed information for the modules, chassis, and software.
- Your SCXI hardware user manuals—If you are using SCXI, read these manuals next for detailed information about signal connections and module configuration. They also explain in greater detail how the module works and contain application hints.
- Your DAQ hardware user manuals—These manuals have detailed information about the DAQ hardware that plugs into or is connected to your computer. Use these manuals for hardware installation and configuration instructions, specification information about your DAQ hardware, and application hints.

- Software manuals—Examples of software manuals you may have are the LabVIEW and LabWindows/CVI manual sets and the NI-DAQ manuals. After you set up your hardware system, use either the application software (LabVIEW or LabWindows/CVI) manuals or the NI-DAQ manuals to help you write your application. If you have a large and complicated system, it is worthwhile to look through the software manuals before you configure your hardware.
- Accessory installation guides or manuals—If you are using accessory products, read the terminal block and cable assembly installation guides or accessory board user manuals. They explain how to physically connect the relevant pieces of the system. Consult these guides when you are making your connections.
- SCXI chassis manuals—If you are using SCXI, read these manuals for maintenance information on the chassis and installation instructions.

Related Documentation

As you read this manual, you may find it helpful to refer to the following documents:

- Your *DAQCard-500 User Manual*
- Your personal computer technical reference manual
- *PC Card Standard, Release 2.1* or later, Personal Computer Memory Card International Association (PCMCIA)
- *Card Services Specifications, Release 2.1* or later, Personal Computer Memory Card International Association (PCMCIA)
- *Socket Services Specifications*, *Release* 2.1 or later, Personal Computer Memory Card International Association (PCMCIA)

Customer Communication

National Instruments wants to receive your comments on our products and manuals. We are interested in the applications you develop with our products, and we want to help if you have problems with them. To make it easy for you to contact us, this manual contains a comment form for you to complete. This form is in Appendix B, *Customer Communication*, at the end of this manual.

Chapter 1
General Description

This chapter describes the general characteristics and gives a configuration overview of the DAQCard-500.

General Characteristics

Thank you for your purchase of the National Instruments DAQCard-500. The DAQCard-500 is a low-cost, low-power analog input, digital I/O, and timing I/O board for computers equipped with a Type II PCMCIA slot. The board contains a 12-bit, successive-approximation ADC with eight single-ended analog inputs, four lines of TTL-compatible digital input, and four lines of digital output. The optional 27-pin I/O connector for the DAQCard-500 enables you to easily connect all your signals directly to the card. The DAQCard-500 is fully software configurable and calibrated so that you can easily install the card and begin your acquisition without having to spend time calibrating the card.

The DAQCard-500 ships with NI-DAQ, National Instruments complete DAQ driver which handles every function listed on the data sheet for our DAQ hardware. Using NI-DAQ you can quickly and easily start your application without having to program the card on the register level.

The small size and weight of the DAQCard-500 coupled with its low-power consumption make this board ideal for use in portable computers, making remote data acquisition practical. The board requires very little power when operating and has a standby mode that uses even less power, thus extending the life of your computer batteries.

In addition, the low cost of a system based on the DAQCard-500 makes it ideal for laboratory work in industrial and academic environments. The multichannel analog input is useful in signal analysis and data logging. The 12-bit ADC is useful in high-resolution applications such as chromatography, temperature measurement, and DC voltage measurement. The eight TTL-compatible digital I/O lines can be used for switching external devices such as transistors and solid-state relays, and for reading the status of external digital logic. The DAQCard-500, used in conjunction with your computer, is a versatile, cost-effective platform for laboratory test, measurement, and control.

Board Configuration Overview

This section is intended as a reference to the DAQCard-500 configuration options. You should already have unpacked your DAQCard-500 and installed it in the computer. The DAQCard-500 is completely software configurable. Refer to your *DAQCard-500 User Manual* if you have not already done these tasks.

Analog Input Configuration

The DAQCard-500 is always configured as follows:

- Single-ended input mode referenced to analog ground
- ±5 V analog input range

Digital I/O Configuration

The DAQCard-500 always uses one 4-bit digital output port and one 4-bit digital input port.

Counter Configuration

You can use the MSM82C54 counter/timers for general-purpose applications, such as pulse and square wave generation, event counting, and pulsewidth, time-lapse, and frequency measurement. For information about configuring the MSM82C54, see the *Timing Connections* section of Chapter 3 in the *DAQCard-500 User Manual*.

Chapter 2
Register Map and Descriptions

This chapter describes in detail the address and function of each of the DAQCard-500 control and status registers.

Register Map

The register map for the DAQCard-500 is given in Table 2-1. This table shows the register name, the register offset address, the register type (read-only, write-only, or read-and-write), and the size of each register in bits.

Table 2-1. DAQCard-500 Register Map

Register Name	Offset Address (Hex)	Type	Size
Configuration and Status Register Group			
Command Register 1	00	Write-only	8 bit
Command Register 2	07	Read-and-write	8 bit
Status Register 1	00	Read-only	8 bit
Status Register 2	01	Read-only	8 bit
Analog Input Register Group			
A/D FIFO Most Significant Byte Register	03	Read-only	8 bit
A/D FIFO Least Significant Byte Register	02	Read-only	8 bit
A/D Clear Register	01	Write-only	8 bit
Counter/Timer (MSM82C54) Register Group			
Counter 0 Data Register	08	Read-and-write	8 bit
Counter 1 Data Register	09	Read-and-write	8 bit
Counter 2 Data Register	0A	Read-and-write	8 bit
Counter Mode Register	0B	Write-only	8 bit
Timer Interrupt Clear Register	06	Write-only	8 bit
Digital I/O Register Group			
Digital Output Register	04	Write-only	8 bit
Digital Input Register	05	Read-only	8 bit

To determine the actual address of these registers, add the offset address shown in Table 2-1 to the board base address. For information about how to determine the base address, see the *PCMCIA Card Initialization* section of Chapter 3, *Programming*.

Register Descriptions

Table 2-1 divides the DAQCard-500 registers into four groups. A bit description of each of the registers is included later in this chapter.

The Configuration and Status Register Group controls the overall operation of the DAQCard-500 and the A/D circuitry. The Analog Input Register Group handles output from the ADC. The Counter/Timer Register Group accesses the onboard MSM82C54 counter/timer integrated circuit. The Digital I/O Register Group consists of the digital output and input registers.

Register Description Format

The remainder of this chapter discusses each of the DAQCard-500 registers in the order shown in Table 2-1. Each register group section begins with a brief introduction followed by a detailed bit description of each register on the DAQCard-500. Each register description gives the address, type, word size, and bit map of the register, and a description of each bit.

The register bit map shows a diagram of the register with the MSB (bit 7 for an 8-bit register) shown on the left and the LSB (bit 0) shown on the right. Each bit is represented by a square with the bit name inside. An asterisk (*) after the bit name indicates that the bit is inverted (negative logic).

In some registers, several bits are labeled with an X, indicating *don't care* bits. When a register is read, these bits may appear set or cleared but should be ignored because they have no significance. When a register is written to, setting or clearing these bit locations has no effect on the DAQCard-500 hardware.

The bit map field for some write-only registers states *not applicable, no bits used*. Writing to these registers causes some event to occur on the DAQCard-500, such as clearing the analog input circuitry. The data is ignored when writing to these registers; therefore, any bit pattern will suffice.

For a detailed bit description of the MSM82C54 chip registers on the DAQCard-500, refer to Appendix A, *MSM82C54 Data Sheet*.

Configuration and Status Register Group

You can use the four registers in the Configuration and Status Register Group to control and monitor the DAQCard-500 A/D and interrupt-control circuitry. The two command registers contain bits that control the operation modes of the A/D circuitry and enable or disable the interrupt operations. The two status registers report the A/D conversion status, A/D conversion error, and the interrupt status. When you start up your computer or insert the card, all bits of the command registers are cleared.

Bit descriptions for the registers in the Configuration and Status Register Group are on the following pages.

Command Register 1

Command Register 1 indicates the input channel to be read, the interrupt enable bits, and whether multiple-channel scanning is enabled.

Address: 00 (hex)

Type: Write-only

Word Size: 8-bit

Bit Map:

7	6	5	4	3	2	1	0
SCANEN*	CNTINTEN	0	FIFOINTEN	0	MA2	MA1	MA0

Bit	Name	Description
7	SCANEN*	Scan Enable—This bit enables or disables multiple-channel scanning during data acquisition. If this bit is cleared, analog channels MA<2..0> are sampled alternately. If this bit is set, a single analog channel selected by MA<2..0> is sampled during the entire DAQ operation. To set up a scanning mode, you must perform two consecutive writes to this register. First, write MA<2..0> with SCANEN* set. This step loads the scan counter. Then, write MA<2..0> with SCANEN* cleared. This step enables scanning. For example, if MA<2..0> is 011 and SCANEN* is first set, then cleared, analog input channels 3 through 0 are sampled alternately during subsequent data conversions. If SCANEN* is set and does not become cleared (with MA<2..0> still set to 011), only analog input channel 3 is sampled during the subsequent data conversions. See *Programming Multiple A/D Conversions with Channel Scanning* in Chapter 3 for more information.
6	CNTINTEN	Counter Interrupt Enable—With this bit, the counter 2 output can cause interrupts. If this bit is set, an interrupt occurs when counter 2 output makes a low-to-high transition. Writing to the Timer Interrupt Clear Register clears this interrupt. If this bit is cleared, interrupts from counter 2 output are ignored.
5, 3	0	Reserved—These bits must be set to zero.
4	FIFOINTEN	FIFO Interrupt Enable—This bit enables and disables the generation of an interrupt when A/D conversion results are available. If FIFOINTEN is set, an interrupt is generated whenever the DAVAIL bit in Status Register 1 is set.

Bit	Name	Description (continued)
2–0	MA<2..0>	Multiplexer Address—These three bits determine which of the eight analog input channels are selected.

MA<2..0>	Selected Analog Input Channels
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

If SCANEN* is cleared, analog channels MA<2..0> are sampled alternately. If SCANEN* is set, a single analog channel specified by MA<2..0> is sampled during the entire DAQ operation. See *Programming Multiple A/D Conversions with Channel Scanning* in Chapter 3 for the correct sequence involved in setting the SCANEN* bit.

Command Register 2

Command Register 2 controls whether data acquisition is enabled.

Address: 07 (hex)

Type: Read-and-write

Word Size: 8-bit

Bit Map:

7	6	5	4	3	2	1	0
0	0	0	0	0	0	DISABDAQ	0

Bit	Name	Description
7–2, 0	0	Reserved—These bits must be set to zero.
1	DISABDAQ	Disable Data Acquisition—This bit disables the DAQ operation. When you start up your computer or insert the DAQCard-500, this bit is cleared and the DAQ operation is enabled. Writing a one to this bit disables both A/D conversion source signals OUT0 and EXTCONV*.

Status Register 1

Status Register 1 indicates the status of the current A/D conversion. The bits in this register indicate if a conversion is being performed, if data is available, if any errors have been found, and whether any counter or external interrupts are currently pending.

Address: 00 (hex)

Type: Read-only

Word Size: 8-bit

Bit Map:

7	6	5	4	3	2	1	0
1	1	1	CONVPROG	1	CNTINT	DATAERR	DAVAIL

Bit	Name	Description
7–5, 3	1	These bits are permanently set to one.
4	CONVPROG	Conversion in Progress—This bit reflects the status of A/D conversion. When an A/D conversion is in progress, this bit is set. Otherwise, it is cleared.
2	CNTINT	Counter Interrupt Status—This bit reflects the status of the interrupt caused by the counter 2 output signal. If the CNTINTEN bit in Command Register 1 is set, a low-to-high transition on the counter 2 output sets this bit and generates an interrupt request. To clear this bit, write to the Timer Interrupt Clear Register.
1	DATAERR	Data Error—This bit indicates if an overflow or overrun error has occurred. If this bit is cleared, no error was encountered. If this bit is set, the A/D FIFO has overflowed because the DAQ servicing operation could not keep up with the sampling rate, or an A/D conversion was initiated before the previous A/D conversion was complete. To distinguish between the overflow and overrun error conditions, examine the OVERFLOW and OVERRUN bits in Status Register 2. To clear this bit, write to the A/D Clear Register.
0	DAVAIL	Data Available—This bit indicates whether conversion output is available. If this bit is set, the ADC is finished with the last conversion and the result can be read from the FIFO. If the FIFO is empty, this bit is cleared. After writing to the A/D Clear Register, this bit is set. A FIFO reading is needed to clear this bit.

Status Register 2

Status Register 2 contains supplementary error information.

Address: 01 (hex)

Type: Read-only

Word Size: 8-bit

Bit Map:

7	6	5	4	3	2	1	0
X	X	X	X	X	X	OVERFLOW	OVERRUN

Bit	Name	Description
7–2	X	Don't care bits.
1	OVERFLOW	Overflow—This bit indicates if an overflow error has occurred. If this bit is cleared, no error was encountered. If this bit is set, the A/D FIFO has overflowed because the DAQ servicing operation could not keep up with the sampling rate. To clear this bit, write to the A/D Clear Register.
0	OVERRUN	Overrun—This bit indicates whether an A/D conversion was initiated before the previous A/D conversion was complete. OVERRUN is an error condition that will occur if the DAQ sample interval is too small (sample rate is too high). If OVERRUN is set, one or more conversions was skipped. If OVERRUN is cleared, no overrun condition has occurred. To clear this bit, write to the A/D Clear Register.

Analog Input Register Group

The two registers making up the Analog Input Register Group control the analog input circuitry and can be used to read the FIFO. Reading the FIFO Register returns stored A/D conversion results. Writing to the A/D Clear Register clears the DAQ circuitry.

The following pages contain bit descriptions for the registers making up the Analog Input Register Group.

A/D FIFO Registers

The 12-bit A/D conversion results are automatically sign-extended to 16-bit data, then stored in a 16-word deep A/D FIFO buffer in two's complement format. Two 8-bit A/D FIFO Registers must be read to return an A/D conversion value stored in the A/D FIFO. The A/D FIFO Least Significant Byte Register (offset 02 hex) must be read first, followed by a read of the A/D FIFO Most Significant Byte Register (offset 03 hex) to obtain an A/D conversion. The A/D FIFO Most Significant Byte Register contains the eight most significant bits of the A/D conversion. The A/D FIFO Least Significant Byte Register contains the eight least significant bits of the A/D conversion. The value read is removed from the A/D FIFO, thereby freeing space for another A/D conversion value to be stored.

The A/D FIFO is empty when all values it contains have been read. You should read the Status Register before reading the A/D FIFO Registers. If the A/D FIFO contains one or more A/D conversion values, the DAVAIL bit is set in the Status Register, and the A/D FIFO Registers can be read to retrieve a value. If the DAVAIL bit is cleared, the A/D FIFO is empty and reading the A/D FIFO Registers returns meaningless information.

The values returned by reading the A/D FIFO Registers are in two's complement binary format.

A/D FIFO Most Significant Byte Register

Address: 03 (hex)

Type: Read-only

Word Size: 8-bit

Bit Map: Two's complement binary mode

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8

{ Sign Extension Bits }

Bit	Name	Description
7–0	D<15..8>	Data Bit <15..8>—These bits contain the most significant byte of the 16-bit two's complement result of a 12-bit A/D conversion.

A/D FIFO Least Significant Byte Register

Address: 02 (hex)

Type: Read-only

Word Size: 8-bit

Bit Map: Two's complement binary mode

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Bit	Name	Description
7–0	D<7..0>	Data Bit <7..0>—These bits contain the least significant byte of the 16-bit two's complement result of a 12-bit A/D conversion.

A/D Clear Register

You can reset the ADC by writing to this register. This operation clears the FIFO, loads the last conversion value into the FIFO, and clears all error bits in the Status Registers. Notice that the FIFO contains one data word after reset; therefore, a FIFO reading is necessary after reset to empty the FIFO. Ignore the data that is read.

Address: 01 (hex)

Type: Write-only

Word Size: 8-bit

Bit Map: Not applicable; no bits used.

Counter/Timer (MSM82C54) Register Group

The five registers in the Counter/Timer Register Group access the onboard MSM82C54 counter/timer. The MSM82C54 has three counters—0, 1, and 2. Counter 0 controls onboard DAQ timing, and counters 1 and 2 are available for general-purpose timing functions.

The MSM82C54 has three independent 16-bit counters and one 8-bit mode register. The mode register sets the mode of operation for each of the three counters. Writing to the Timer Interrupt Clear Register clears the interrupt request asserted when a low pulse is detected on the output of counter 2.

The following pages contain bit descriptions for the registers in the Counter/Timer Register Group.

Counter 0 Data Register

The Counter 0 Data Register is used to load and read back the contents of MSM82C54 counter 0.

Address: 08 (hex)

Type: Read-and-write

Word Size: 8-bit

Bit Map:

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Bit	Name	Description
7–0	D<7..0>	Data Bit <7..0>—These bits are 8-bit counter 0 contents.

Counter 1 Data Register

The Counter 1 Data Register loads and reads back the contents of MSM82C54 counter 1.

Address: 09 (hex)

Type: Read-and-write

Word Size: 8-bit

Bit Map:

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Bit	Name	Description
7–0	D<7..0>	Data Bit <7..0>—These bits are 8-bit counter 1 contents.

Counter 2 Data Register

The Counter 2 Data Register loads and reads back the contents of MSM82C54 counter 2.

Address: 0A (hex)

Type: Read-and-write

Word Size: 8-bit

Bit Map:

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Bit	Name	Description
7–0	D<7..0>	Data Bit <7..0>—These bits are 8-bit counter 2 contents.

Counter Mode Register

The Counter Mode Register determines the operation mode for each of the three counters on the MSM82C54 chip. The Counter Mode Register selects the counter involved, the counter read/load mode, its operation mode (any of the six modes possible with this chip), and the counting mode (binary or BCD).

The Counter Mode Register is an 8-bit register. Bit descriptions for each of these bits are included in Appendix A, *MSM82C54 Data Sheet*.

Address: 0B (hex)

Type: Write-only

Word Size: 8-bit

Bit Map:

7	6	5	4	3	2	1	0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

Bit **Name** **Description**

7–6 SC<1..0> Counter Select—These bits select the counter on which the command operates.

SC1	SC0	Operation
0	0	Select counter 1
0	1	Select counter 2
1	0	Select counter 3
1	1	Read-back command

5–4 RL<1..0> Read/Write or Latch Select—These bits select data written to or read from a counter or send a Counter Latch command.

RL1	RL0	Operation
0	0	Counter Latch command
0	1	Read and write least significant byte only
1	0	Read and write most significant byte only
1	1	Read and write least significant byte then most significant byte

Bit	Name	Description (continued)
		The Counter Latch command latches the current count of the register selected by SC1 and SC0. The next read from the selected counter returns the latched data.
3–1	M<2..0>	Counter Mode Select—These bits select the counting mode of the selected counter. The following table lists six available modes and the corresponding bit settings. Refer to Appendix A, *MSM82C54 Data Sheet*, for additional information.

M2	M1	M0	Mode
0	0	0	Mode 0—interrupt on terminal count
0	0	1	Mode 1—hardware retriggerable one shot
0	1	0	Mode 2—rate generator
0	1	1	Mode 3—square wave mode
1	0	0	Mode 4—software retriggerable strobe
1	0	1	Mode 5—hardware retriggerable strobe

0	BCD	Binary Coded Decimal Select—If BCD is set, the selected counter keeps count in BCD format. If BCD is cleared, the selected counter keeps count in 16-bit binary format.

Timer Interrupt Clear Register

Writing to the Timer Interrupt Clear Register clears the interrupt request asserted when a low pulse is detected on the counter 2 output.

Address: 06 (hex)

Type: Write-only

Word Size: 8-bit

Bit Map: Not applicable; no bits used.

Digital I/O Register Group

The Digital I/O Register Group contains two registers—the Digital Output Register and the Digital Input Register. The Digital Output Register drives the four digital output lines of the I/O connector. The Digital Input Register returns the digital state of the four digital input lines of the I/O connector.

Digital Output Register

Writing to the Digital Output Register controls the four digital output lines of the I/O connector. The pattern contained in the Digital Output Register is driven onto the four digital output lines of the I/O connector.

Address: 04 (hex)

Type: Write-only

Word Size: 8-bit

Bit Map:

7	6	5	4	3	2	1	0
0	0	0	0	D3	D2	D1	D0

Bit	Name	Description
7–4	0	Reserved—These bits must be set to zero.
3–0	D<3..0>	Data Bit <3..0>—These bits are 4-bit output data. These four bits control the digital output lines DOUT<3..0>.

Digital Input Register

The Digital Input Register, when read, returns the logic state of the four digital input lines of the I/O connector.

Address: 05 (hex)

Type: Read-only

Word Size: 8-bit

Bit Map:

7	6	5	4	3	2	1	0
1	1	1	1	D3	D2	D1	D0

Bit	Name	Description
7–4	1	Reserved—These bits are permanently set to one.
3–0	D<3..0>	Data Bit <3..0>—These bits are 4-bit output data. These four bits control the digital input lines DIN<3..0>.

Chapter 3
Programming

This chapter contains programming instructions for operating the circuitry on the DAQCard-500. You can program the DAQCard-500 in two stages—by configuring the card using the Card and Socket Services from the PCMCIA system software, and by writing or reading the various registers on the board. The Card and Socket Services software determines the card base address and its interrupt level and places the card in unconfigured or powered-down states. You can use the registers described in Chapter 2, *Register Map and Descriptions*, to control and monitor DAQ operations following the procedures explained in this chapter.

PCMCIA Card Initialization

Before you can access the DAQ circuitry on the DAQCard-500, the card must be activated using Card Services. PCMCIA I/O cards are kept inactive until a program has requested that Card Services activate the card by assigning an interrupt level and an address space for the card I/O registers. The DAQCard-500 requires a 32-byte I/O address window and one interrupt level.

There are at least two different ways of activating the card:

- If you are using the DAQCard-500 with National Instruments software such as NI-DAQ, LabVIEW, and LabWindows/CVI, the NI-DAQ configuration programs request the activation of the card. For more information about this procedure, see the section on installation and configuration in your NI-DAQ manual.

- If the option above is not feasible for your application, you can develop your own program to activate the card. However, this is fairly complicated, and it requires significantly more programming. If you develop your own program, you should consult the documents, *Card Services Specifications* and *Socket Services Specifications* (PCMCIA) which explain how to activate a card using system-level calls. You will need to request an I/O window, an interrupt level, and a configuration. In the configuration, the configuration index should be set to 01 hex for normal operation. For more information about these operations, see the PCMCIA documents, *Card Services Specifications* and *Socket Services Specifications.*

After you activate the card, you are ready to configure the DAQCard-500 for the DAQ setup. The following pages explain how to set the registers for different operations.

Register Programming Considerations

The following programming instructions are language independent; that is, they tell you to write a value to a given register, to set or clear a bit in a given register, or to detect whether a given bit is set or cleared, without presenting the actual language-specific program code needed to accomplish this.

Several write-only registers on the DAQCard-500 contain bits that control multiple, independent pieces of the onboard circuitry. You should set or clear specific register bits, following the instructions in this chapter, without changing the current state of the remaining bits in the register. However, writing to these registers affects all register bits simultaneously. You cannot read these registers to determine which bits have been previously set or cleared; therefore, you should maintain a software copy of the write-only registers. You can determine the status of the write-only registers using this software copy. To change the state of a single bit without disturbing the remaining bits, set or clear the bit in the software copy and write the software copy to the register.

Initializing the DAQCard-500

You must initialize the DAQCard-500 hardware for the circuitry to operate properly. To initialize the DAQCard-500 hardware, complete the following steps:

1. Write 80 hex to Command Register 1.
2. Write 00 hex to Command Register 2.
3. Write 34 hex to the Counter Mode Register.
4. Write 00 hex to the Timer Interrupt Clear Register.
5. Write 00 hex to the A/D Clear Register.
6. Read the data from the A/D FIFO Registers. Ignore the data.

Steps 1 through 6 leave the DAQCard-500 circuitry in the following state:

- Data acquisition is enabled.
- Counter 0 output is high.
- Multiple-channel scanning is disabled.
- All interrupts are disabled.
- Analog input circuitry is initialized to channel 0.
- A/D FIFO is cleared.

In addition, see Appendix A, *MSM82C54 Data Sheet*, for more details concerning initialization of the MSM82C54 counter/timer.

Programming the Analog Input Circuitry

This section describes the analog input circuitry programming sequence, explains the A/D conversion results, and explains how to clear the analog input circuitry.

Analog Input Circuitry Programming Sequence

To program the analog input circuitry, perform the following steps:

1. Select the analog input channel by writing to Command Register 1. See the *Command Register 1* bit description in Chapter 2 for analog input channel bit patterns. Set up the bits as shown in the bit description, and only if the analog input channel or the scanning mode needs to be changed write to Command Register 1.

2. Initiate an A/D conversion with a low-to-high transition on the counter 0 output (OUT0) or on the EXTCONV* line.

   When an A/D conversion is initiated, the ADC stores the result in the A/D FIFO at the end of its conversion cycle. When EXTCONV* initiates the conversion, set OUT0 high.

3. Obtain the A/D conversion result by reading the A/D FIFO Registers. Before you read the A/D FIFO, however, you must read Status Register 1 to determine whether the A/D FIFO contains any results.

   To read the A/D conversion results, complete the following steps:

   a. Read Status Register 1 (8-bit read).

   b. If the DAVAIL bit is set, read the A/D FIFO Least Significant Byte Register followed by a read of the A/D FIFO Most Significant Byte Register to obtain the result. Reading the two 8-bit A/D FIFO Registers removes the A/D conversion result from the A/D FIFO.

The DAVAIL bit indicates whether one or more A/D conversion results are stored in the A/D FIFO. If the DAVAIL bit is cleared, the A/D FIFO is empty, and reading the A/D FIFO Registers returns meaningless data. When an A/D conversion is initiated, the DAVAIL bit will be set (by the card) within 20 µs. If EXTCONV* is being used for A/D timing, the DAVAIL bit will be set within 20 µs after a rising edge on EXTCONV*.

The DATAERR bit in Status Register 1 indicates if a data error has occurred during a conversion. This error can be either an overflow error or an overrun error.

An A/D FIFO overflow condition occurs if more than 16 conversions are initiated and stored in the A/D FIFO before the A/D FIFO Register is read. If this condition occurs, the OVERFLOW bit is set in Status Register 2 to indicate that one or more A/D conversion results have been lost because of FIFO overflow. Writing to the A/D Clear Register resets this error flag. Perform a FIFO dummy read after an A/D Clear write to reset the FIFO.

An A/D overrun condition occurs if a conversion is initiated before the last one has been completed. If this condition occurs, the OVERRUN bit is set in Status Register 2 to indicate that one or more A/D conversion results have been lost because of overrun. Writing to the A/D Clear Register resets this error flag. Once you have written to the A/D Clear Register, perform a FIFO dummy read to clear the FIFO.

A/D FIFO Output Binary Modes

The A/D conversion result stored in the A/D FIFO is a 16-bit two's complement value. It is composed of an 11-bit magnitude plus a 1-bit sign and a 4-bit sign extension.

Table 3-1 shows the input voltage for specified A/D conversion values for each voltage range, assuming there are no ADC offset or gain errors.

Table 3-1. A/D Conversion Values

A/D Conversion Result		**Input Voltage**
(Decimal)	**(Hex)**	**±5 V Range**
-2,048	F800	-5.0
-1,024	FC00	-2.5
0	0000	0
1,024	0400	2.5
2,047	07FF	4.9976

Clearing the Analog Input Circuitry

You can clear the analog input circuitry by writing to the A/D Clear Register, which leaves the analog input circuitry in the following state:

- Analog input error flags DATAERR, OVERFLOW, and OVERRUN are cleared.
- Pending interrupt requests are cleared.
- A/D FIFO has one useless word of data.

Before starting any A/D conversions, empty the A/D FIFO by reading the two A/D FIFO Registers and ignoring the data. This guarantees that the A/D conversion results read from the A/D FIFO are the results from the initiated conversions rather than from previous conversions.

To clear the analog input circuitry and the A/D FIFO, complete these steps:

1. Write 0 to the A/D Clear Register (8-bit write).

2. Read the A/D FIFO Registers and ignore the data.

Software Calibration

Because the ADC on the DAQCard-500 is not calibrated in hardware, the values returned from an A/D FIFO read must be scaled to correct for gain and offset errors. Otherwise, 12-bit accuracy will not be achieved and the calibrated gain and offset error specifications from the *DAQCard-500 User Manual* will not be valid.

To correct for offset error, you must subtract a fixed value from every sample. This value is the average value an A/D FIFO read returns when the input channel is tied to AGND. The actual offset error varies between boards.

To correct for gain errors, you must multiply the offset-corrected value by the gain-scaling factor. The gain-scaling factor is defined as follows:

Gain-scaling factor = (desired ADC reading)/(actual ADC reading) at full scale

For example, the gain-scaling factor for the ±5 V range would be:

Gain-scaling factor = (2,047)/(actual ADC reading for a 4.9995 input)

where all values are in decimal format.

National Instruments measures these two constants after the board has been manufactured and stores the appropriate values onboard the DAQCard-500. You may use these preset values in your own calibration process, or you may measure them yourself, thereby correcting for any errors introduced by external input buffering or conditioning circuitry.

The following sections describe how to read the calibration constants stored on your DAQCard-500 by National Instruments. If you will not be writing a program to determine these constants, you do not need to read these sections.

Reading the Onboard Calibration Constants

When the DAQCard-500 is calibrated, the calibration constants are stored in the Card Information Structure (CIS), a memory area used by the PCMCIA system software to determine the card characteristics. Information in this area is stored as tuples, which are distinct packets of different types of data. The calibration constants are stored in copies of a special tuple, `CISTPL_ADC_CALIBRATION`, which has been defined by National Instruments.

To read the information in these tuples, you must make system-level calls to the PCMCIA Card Services software. The document *PCMCIA Card and Socket Services Specifications* explains how these calls are performed for particular systems.

To obtain the data from the first tuple, call the Card Services routine `GetFirstTuple`, with the `Desired Tuple` field set to 81 hex. Examine the remaining tuples by calling `GetNextTuple` with the same parameters until you get the message `NO_MORE_TUPLES`. When you have located a tuple, use `GetTupleData` to obtain the tuple data fields (bytes 2 and higher).

Structure of the Calibration Constants Tuple

Table 3-2 lists information on fields present in each tuple.

Table 3-2. ADC Calibration Tuple

Byte	Field Name	Field Description
0	`TPL_CODE`	ADC Calibration tuple code (`CISTPL_ADC_CALIBRATION`, 81 hex)
1	`TPL_LINK`	Number of bytes following this one
2	`ADC_GAIN_CODE`	(see below)
3	`ADC_OFFSET_ERROR`	2 times ADC offset error
4	`ADC_GAIN_ERROR`	-2 times full-scale gain error

The gain codes indicate which voltage range and input mode you have selected. The proper gain code byte for the various options available on the DAQCard-500 is as follows:

Table 3-3. CIS Gain Codes on the DAQCard-500

Input Mode	Input Voltage Range	ADC_GAIN_CODE (Hex)
Single-ended	±5 V	02
Default	Default	FF

If the appropriate tuple for a range is not present, use the values from the default tuple.

The offset error value encoded in byte 3 is twice the average ADC reading for a grounded (0 V) input, represented as a two's complement number. Thus, to correct for offset error, divide this number by two and subtract it from each sample.

The gain error value encoded in byte 4 is the negative of twice the full-scale gain error in LSBs, represented as a two's complement number. To correct for gain error, divide this number by 4,096 (which is twice the full-scale reading), multiply the result by a particular sample value, and add the result to that sample.

Programming Multiple A/D Conversions on a Single Input Channel Using Counter 0

This manual refers to a sequence of timed A/D conversions as a *DAQ operation*. Counter 0 of the MSM82C54 is used as the sample-interval counter. In a DAQ operation, counter 0 continuously generates the conversion pulses. The software tracks the number of conversions that have occurred and turns off counter 0 after the required number of conversions has been obtained. The number of conversions in a single DAQ operation is unlimited. Counter 0 always uses a 1 MHz base clock.

To program your DAQCard-500 for multiple A/D conversions using channel 0, complete the following steps:

1. Select the analog input channel by writing to Command Register 1. The SCANEN* bit must be set for DAQ operations on a single channel. For analog input channel bit patterns, see the *Command Register 1* bit descriptions in Chapter 2, *Register Map and Descriptions*.

   Command Register 1 needs to be written to only when the analog input channel, scanning mode, or interrupt mode needs to be changed.

2. Program the sample-interval counter (counter 0). The sample interval is the time between successive A/D conversions. You can program counter 0 to generate a pulse on OUT0 once every *N* CLK0 pulses, where *N* can be between 2 and 65,535. A low-to-high transition on OUT0 (counter 0 output) initiates a conversion. Because the 1 MHz clock is internally connected to CLK0 (the clock that counter 0 uses), counter 0 will initiate a conversion every *N* µs.

   Use the following programming sequence to program the sample-interval counter. All writes are 8-bit write operations. All values given are hexadecimal.

   a. Write 34 to the Counter Mode Register (select counter 0, mode 2).

   b. Write the least significant byte of the sample interval to the Counter 0 Data Register.

   c. Writing 34 to the Counter Mode Register (step a) forces OUT0 to high. To finish programming counter 0, you must also write the most significant byte. However, because this writing starts the counting, this writing is performed later in step 4.

3. Clear the A/D circuitry. Before starting the DAQ operation, empty the A/D FIFO to clear out any old A/D conversion results. You must do this after the counters are programmed in case any spurious edges were caused while programming the counters. Write 0 to the A/D Clear Register to empty the FIFO (8-bit write), and follow this write with a read from the A/D FIFO Registers. Ignore the data obtained in the read.

4. Start and service the DAQ operation by writing the most significant byte of the sample interval to the Counter 0 Data Register. This enables counter 0 to start counting.

   After starting the DAQ operation, service the operation by reading the A/D FIFO Registers every time an A/D conversion result becomes available. To service the data acquisition, perform the following sequence until the desired number of conversion results has been read:

   a. Read Status Register 1 (8-bit read).

   b. If the DAVAIL bit (bit 0) is set, read the A/D FIFO Least Significant Byte Register followed by a read of the A/D FIFO Most Significant Register to obtain the result.

You can also use interrupts to service the DAQ operation. This topic is discussed in the *A/D Interrupt Programming* section later in this chapter.

Overflow or overrun error conditions may occur during a DAQ operation. These error conditions are reported through the Status Register 1 bit DATAERR. Every time you read Status Register 1 to check the DAVAIL bit, you should also check the DATAERR bit to determine if an error occurred. When the DATAERR bit is set, either an OVERFLOW or an OVERRUN error has occurred. You can determine which error you received by subsequently reading Status Register 2 as described later in this chapter.

An overflow condition occurs if more than 16 A/D conversions have been stored in the A/D FIFO since the A/D FIFO was last read; that is, the A/D FIFO is full and cannot accept any more data. This condition occurs if the software loop reading the A/D FIFO Register is not fast enough to keep up with the A/D conversion rate. When an overflow occurs, at least one A/D conversion result has been lost. An overflow condition has occurred if the OVERFLOW bit in Status Register 2 is set.

An overrun condition occurs if an A/D conversion was initiated before the previous A/D conversion was complete. Overrun is an error condition that may occur if the DAQ sample interval is too small (in other words, if the sample rate is too high.) If OVERRUN is set, one or more conversions was skipped. An overrun condition has occurred if the OVERRUN bit in Status Register 2 is set.

Clear the DATAERR, OVERFLOW, and OVERRUN bits by writing to the A/D Clear Register.

To stop the A/D conversion sequence, write 34 to the Counter 0 Mode Register to stop the generation of pulses on OUT0.

Programming Multiple A/D Conversions Using External Timing

You can use the external timing signal EXTCONV* for multiple A/D conversions. A low-to-high transition of EXTCONV* initiates an A/D conversion. The software can initiate a DAQ operation. Setting the DISABDAQ bit in Command Register 2 disables both the EXTCONV* signal and OUT0 on the counter. Clearing the DISABDAQ bit in Command Register 2 enables the EXTCONV* signal and starts the DAQ operation. In addition, to use the EXTCONV* signal, the OUT0 of counter 0 must be driven high. Otherwise, EXTCONV* is disabled.

Follow these steps to program multiple A/D conversions using external timing:

1. To disable the A/D conversion, write 2 to Command Register 2 to set the DISABDAQ bit. Any pulse on the EXTCONV* line is ignored.

2. Program counter 0. The high output of counter 0 enables the EXTCONV* signal. Write 34 to the Counter Mode Register to force OUT0 high (enable EXTCONV*). Writing 30 to the Counter Mode Register forces OUT0 low, disables the EXTCONV*, and stops the DAQ operation.

3. Select the analog input channel by writing to Command Register 1. The SCANEN* bit must be set for DAQ operation on a single channel. See the *Command Register 1* bit description in Chapter 2 for analog input channel bit descriptions.

4. Clear the A/D circuitry. Before starting the DAQ operation, empty the A/D FIFO to clear any old A/D conversion results. Write 0 to the A/D Clear Register and read the A/D FIFO Registers to empty the FIFO. Ignore the data.

5. Start and service the DAQ operation by clearing the DISABDAQ bit in Command Register 2.

    a. Write 0 to Command Register 2 to enable the A/D conversion.

    b. The next EXTCONV* signal initiates an A/D conversion. The operation must be serviced by reading the A/D FIFO Registers every time an A/D conversion result becomes available. To service the data acquisition, perform the following sequence until the desired number of conversion results has been read:

        1. Read Status Register 1.

        2. If the DAVAIL bit is set, read the A/D FIFO Least Significant Byte Register followed by a read of the A/D FIFO Most Significant Byte Register to obtain the result.

You can also use interrupts to service the DAQ operation. This topic is discussed in the *A/D Interrupt Programming* section later in this chapter.

Overflow or overrun error conditions may occur during a DAQ operation. These error conditions are reported through the Status Register 1 bit, DATAERR. Every time you read Status Register 1 to check the DAVAIL bit, you should also check the DATAERR bit to determine if an error occurred. When the DATAERR bit is set, either an OVERFLOW or an OVERRUN error has occurred. You can determine which of the errors you received by subsequently reading Status Register 2 as described later in this chapter.

An overflow condition occurs if more than 16 A/D conversions have been stored in the A/D FIFO since the A/D FIFO was last read; that is, the A/D FIFO is full and cannot accept any more data. This condition occurs if the software loop reading the A/D FIFO Register is not fast enough to keep up with the A/D conversion rate. When an overflow occurs, at least one A/D conversion result is lost. An overflow condition has occurred if the OVERFLOW bit in Status Register 2 is set.

An overrun condition occurs if an A/D conversion was initiated before the previous A/D conversion was complete. Overrun is an error condition that may occur if the DAQ sample interval is too small (in other words, if the sample rate is too high.) If OVERRUN is set, one or more conversions was skipped. An overrun condition has occurred if the OVERRUN bit in Status Register 2 is set.

Clear the DATAERR, OVERFLOW, and OVERRUN bits by writing to the A/D Clear Register.

Programming Multiple A/D Conversions with Channel Scanning

You can use the DAQ programming sequences given earlier in this chapter to program the DAQCard-500 for multiple A/D conversions on a single input channel. You can also program the DAQCard-500 to scan analog input channels during the DAQ operation. Analog channels *N* through 0 can be scanned, where *N* can be 1 through 7. Programming scanned multiple A/D conversions involves the same sequence of steps as single-channel DAQ operations except that the SCANEN* bit is cleared in Command Register 1. When the SCANEN* bit is cleared in Command Register 1, the analog channel select bits MA<2..0> select the highest numbered channel in the scan sequence. For example, if MA<2..0> is 011 (binary)—that is, channel 3 is selected and the SCANEN* bit is cleared—the following scan sequence is used:

> channel 3, channel 2, channel 1, channel 0, channel 3, channel 2, channel 1, channel 0, channel 3, and so on

Note: ***To select the analog input channel, follow these steps in this order:***

1. ***Write the configuration value indicating the highest channel number in the scan sequence to Command Register 1. The SCANEN* bit must be set to Command Register 1 during this first write.***

2. ***Write the same configuration value again to Command Register 1. The SCANEN* bit, however, must be cleared during the second write to Command Register 1.***

Use either counter 0 or EXTCONV* to control the scanning interval.

A/D Interrupt Programming

You can use a conversion interrupt to service the DAQ operation. To use the interrupt, set the FIFOINTEN bit in Command Register 1. If this bit is set is cleared, an interrupt is generated whenever the DAVAIL bit in Status Register 1 is set. This interrupt condition will be cleared when the FIFO is empty. To empty the FIFO, read all of its contents.

Programming the Digital I/O Circuitry

DIN0 through DIN3 (pins 12 through 15) of the I/O connector are dedicated digital input lines that are monitored by the Digital Input Register. An 8-bit read of the Digital Input Register returns the current state of these digital input lines. DOUT0 through DOUT3 (pins 16 through 19) of the I/O connector are dedicated digital output lines. The Digital Output Register always drives these lines. An 8-bit write to the Digital Output Register drives the new digital value to these lines. At startup, all of the digital output lines are initialized to zero.

Programming the MSM82C54 Counter/Timer

Counters 1 and 2 of the MSM82C54 counter/timer are available for general-purpose timing applications. Counter 0 has a fixed 1 MHz clock input and is used as the sample interval counter for A/D conversions. In addition, the same 1 MHz clock is available on the I/O connector (pin 20) for use with counters 1 and 2.

Write and read operations to the MSM82C54 are 8-bit operations. For general programming details, refer to Appendix A, *MSM82C54 Data Sheet*.

Appendix A
OKI MSM82C54 Data Sheet*

This appendix contains a manufacturer data sheet for the MSM82C54 CMOS programmable interval timer (OKI Semiconductor). This timer is used on the DAQCard-500.

This data not available in electronic format.

Appendix B
Customer Communication

For your convenience, this appendix contains forms to help you gather the information necessary to help us solve technical problems you might have as well as a form you can use to comment on the product documentation. Filling out a copy of the *Technical Support Form* before contacting National Instruments helps us help you better and faster.

National Instruments provides comprehensive technical assistance around the world. In the U.S. and Canada, applications engineers are available Monday through Friday from 8:00 a.m. to 6:00 p.m. (central time). In other countries, contact the nearest branch office. You may fax questions to us at any time.

Corporate Headquarters
(512) 795-8248
Technical support fax: (800) 328-2203
(512) 794-5678

Branch Offices	**Phone Number**	**Fax Number**
Australia	03 9 879 9422	03 9 879 9179
Austria	0662 45 79 90 0	0662 45 79 90 19
Belgium	02 757 00 20	02 757 03 11
Canada Ontario)	519 622 9310	519 622 9311
Canada (Quebec)	514 694 8521	514 694 4399
Denmark	45 76 26 00	45 76 71 11
Finland	90 527 2321	90 502 2930
France	1 48 14 24 24	1 48 14 24 14
Germany	089 741 31 30	089 714 60 35
Hong Kong	2645 3186	2686 8505
Italy	02 48301892	02 48301915
Japan	03 5472 2970	03 5472 2977
Korea	02 596 7456	02 596 7455
Mexico	95 800 010 0793	05 404 0890
Netherlands	03480 33466	03480 30673
Norway	32 84 84 00	32 84 86 00
Singapore	2265886	2265887
Spain	91 640 0085	91 640 0533
Sweden	08 730 49 70	08 730 43 70
Switzerland	056 20 51 51	056 20 51 55
Taiwan	02 377 1200	02 737 4644
U.K.	01635 523545	01635 523154

Documentation Comment Form

National Instruments encourages you to comment on the documentation supplied with our products. This information helps us provide quality products to meet your needs.

Title: **DAQCard™-500 Register-Level Programmer Manual**

Edition Date: **July 1995**

Part Number: **340898A-01**

Please comment on the completeness, clarity, and organization of the manual.

If you find errors in the manual, please record the page numbers and describe the errors.

Thank you for your help.

Name _____

Title _____

Company _____

Address _____

_____

Phone (_____) _____

Mail to: Technical Publications
National Instruments Corporation
6504 Bridge Point Parkway, MS 53-02
Austin, TX 78730-5039

Fax to: Technical Publications
National Instruments Corporation
MS 53-02
(512) 794-5678

Glossary

The following metric system prefixes are used with abbreviations for units of measure:

Prefix	Meaning	Value
p-	pico-	10^{-12}
n-	nano-	10^{-9}
µ-	micro-	10^{-6}
m-	milli-	10^{-3}
k-	kilo-	10^{3}
M-	mega-	10^{6}
G-	giga-	10^{12}

A/D	analog-to-digital
ADC	analog-to-digital converter
BCD	binary-coded decimal/select bit
CIS	Card Information Structure
CMOS	complementary metallic oxide semiconductor
CNTINT	counter interrupt status bit
CNTINTEN	counter interrupt enable bit
CONVPROG	conversion in progress bit
D	data bit
DAQ	data acquisition
DATAERR	data error bit
DAVAIL	data available bit
DC	direct current
DIN	digital input bit/signal
DISABDAQ	disable data acquisition bit
DOUT	digital output bit/signal
EXTCONV*	external conversion signal
FIFO	first-in-first-out
FIFOINT	FIFO interrupt bit
FIFOINTEN	FIFO interrupt enable bit
hex	hexadecimal
Hz	hertz
I/O	input/output
LSB	least significant bit
M	counter mode select bit
MA	multiplexer address bit
MB	megabytes of memory
MSB	most significant bit
OUT	counter output
OVERFLOW	overflow error status bit

OVERRUN	overrun error status bit
PCMCIA	Personal Computer Memory Card International Association
RL	read/load select bit
RAM	random-access memory
rms	root mean square
SC	counter select bit
SCANEN*	scan enable bit
TTL	transistor-transistor logic
V	volts
X	don’t care bit

Index

C

D

E

F

I

M

O

P

R

T