Publication Number 29-485

OS/32 MINI I/O SYSTEM

USER MANUAL

PERKIN-ELMER

Computer Systems Division
2 Crescent Place
Oceanport, N.J. 07757

Table of Contents

TABLES

ILLUSTRATIONS

APPENDICES

CHAPTER 1
SYSTEM OVERVIEW

INTRODUCTION

The Mini I/O Software (Figure 1-1) enables the OS/32 user to use the Mini I/O System Controllers for performing analog to digital conversion, digital to analog conversion, and digital input/output transfers. It includes a set of FORTRAN RTL routines to allow FORTRAN user's convenient access to the Mini I/O System equipment. The hardware associated with the Mini I/O System is described in the *Mini I/O System Instruction Manual*, Publication Number 29-443, and the user can identify the equipment suitable for his application needs.

RELATED DOCUMENTS

In reading this manual it is assumed that the reader is familiar with the following documents:

Analog Input Controller Program Manual, Publication Number 29-475
Analog Output Controller Program Manual, Publication Number 29-476
Digital Input/Output Controller Program Manual, Publication Number 29-477
OS/32 MT Program Configuration Manual, Publication Number 29-389
FORTRAN V Level 1 User's Guide, Publication Number 29-361
OS/32 MT Program Reference Manual, Publication Number B29-390
OS/32 Source Updater Publication Number 03-090A15
Mini I/O System Instruction Manual, Publication Number 29-443

SOFTWARE PACKAGE DESCRIPTION

The software portion of the Mini I/O System consists of three drivers which control the operation of the Analog Output, Analog Input and Digital Input/Output Controllers as shown in Figure 1. The Mini I/O FORTRAN RTL routines form the remaining portion of the software package and replace three routines in the existing Run Time Library.

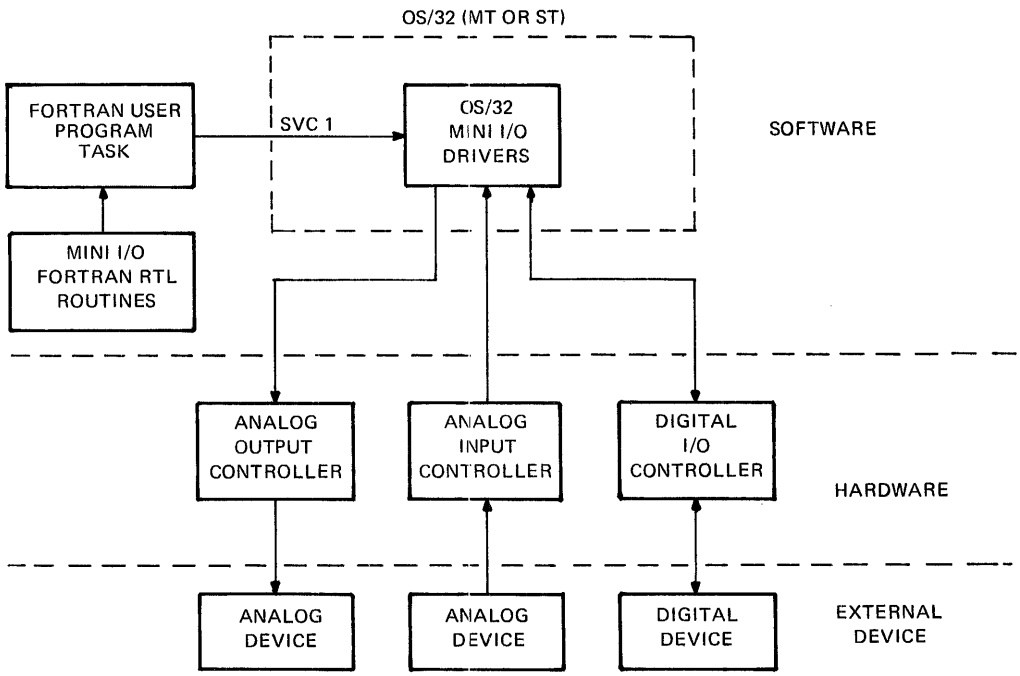

Figure 1-1. Mini I/O System

The software package that supports the Mini I/O System consists of the following:

1. Source and Object Code of 3 drivers and 3 DCBs:

Module name	Driver Name
DCB/CCB for Analog Input Controller	DCB138
Analog Input Controller Driver	INITAIDV
DCB for Analog Output Controller	DCB139
Analog Output Controller Driver	INITAODV
DCB/CCB for Digital I/O Controller	DCB140
Digital I/O Controller Driver	INITDIOV

2. Object Code of 7 FORTRAN RTL Routines:

Routine Description	RTL Library Name
Analog Inputs in a Sequential Order	AISQW
Analog Inputs in Any Sequence	AIRDW
Analog Output	AOW
Digital Input Non-Handshaking Mode	DIW
Digital Input Handshaking Mode	DIWH
Digital Output Non-Handshaking Mode	DOLW
Digital Output Handshaking Mode	DOLWH

NOTE

The source and object code of the drivers and RTL routines are shipped on magnetic media. See Appendix 3 for magnetic media and disc table of contents.

MANUAL ORGANIZATION

The remaining portions of this manual discuss the following: in Chapter 2, System Operation, how to edit the FORTRAN RTL to incorporate the Mini I/O RTL routines and the device codes for the various controllers; in Chapter 3, System Description, driver descriptions demonstrating how to set up an SVC 1 parameter block and description of the FORTRAN RTL routines; and in Chapter 4, System Application, examples of analog and digital applications showing Mini I/O FORTRAN RTL routine usage.

CHAPTER 2
SYSTEM OPERATION

INTRODUCTION

To incorporate the software portion of the Mini I/O System into OS/32, the operating system must be SYSGENed to include the Mini I/O drivers and DCBs. This procedure requires the use of the Configuration Utility Program (CUP/MT). The inclusion of the Mini I/O drivers and DCBs is accomplished in exactly the same manner as for any other device. For a detailed description of this procedure, refer to the *OS/32 MT Program Configuration Manual,* Publication Number 29-389.

DEVICE CONFIGURATION

The Mini I/O equipment should not be configured on a Selector Channel. The Level field should always contain 1 and the Flags field should not be used.

The following codes are used by the Mini I/O System devices:

	Device Code	Device Number
Analog Input Controller	138	88
Analog Output Controller	139	98-9B
Digital I/O Controller	140	A8-A9

Example:

```
DEVICES

1    AIC:088,138          Analog Input Controller
1    AOC1:098,139         Analog Output Controller  1
1    AOC2:099,139         Analog Output Controller  2
1    AOC3:09A,139         Analog Output Controller  3
1    AOC4:09B,139         Analog Output Controller  4
1    DIC:0A9,140          Digital Input Controller
1    DOC:0A8,140          Digital Output Controller
ENDD
```

FORTRAN RTL FILE

To replace the three subroutines AISQW, AIRDW and AOW in the existing FORTRAN Run Time Library with the new set of Mini I/O routines:

1. Load the OS/32 Library Loader utility program.
2. Assign LU 1 to the old RTL file/device.
3. Allocate and assign LU 2 to a new RTL file.
4. Assign LU 3 to some print device.
5. Assign LU 4 to the Mini I/O RTL routines file/device.
6. Assign LU 5 to the command input device.
7. Start the Utility Program.

The command sequence is:

```
DU 1,2 AISQW      Programs copies from LU 1 to LU 2 until AISQW is reached
DU 4,2            Programs copies from LU 4 to LU 2
FI 1 START        Find START on LU 1
DU 1,2            Programs copies from LU 1 to LU 2
RW 2              Rewind LU 2
TA 2,3            LU 2 program labels listed on LU 3
END
```

A new RTL file is generated and LU 3 lists all subroutine labels in the new RTL which include:

```
AISQW
AIRDW
AOW
DIW
DIWH
DOLW
DOLWH
```

To use the OS/32 Library Loader utility, refer to the *32-Bit Family Loader Description Manual.* To compile a FORTRAN program, refer to the *FORTRAN V Level 1 User's Guide.* To establish a FORTRAN program and to edit an RTL routine, refer to *TET/32 User's Manual.*

CHAPTER 3
SYSTEM DESCRIPTION

INTRODUCTION

This chapter describes the OS/32 DCB/Drivers and the FORTRAN RTL routines used in the Mini I/O System. The DCB/Drivers are described by the following program functional characteristics:

- Program Identification
- Supported Devices
- Supported Attributes
- Functional Description
- Status Definition
- Size Information

ANALOG INPUT CONTROLLER DRIVER

Program Identification

The Analog Input Controller Driver, Program Number 07-083F01 is part of the Mini I/O System, Program Number 04-059. For further information, refer to the *Analog Input Controller Program Manual,* Publication Number 29-475.

Supported Devices

Device	Product Number	OS/32 Device Code
Analog Input Controller	M48-212	138
	M48-213	138
	M48-214	138
	M48-215	138

Supported Attributes

Read, Wait/Proceed, Unconditional Proceed, Binary (no ASCII), Random (no Sequential), Format/Image (ignored), variable record length, command functions ignored.

Functional Description

The random address (RAD) field of the standard OS/32 MT or OS/16 MT2 SVC 1 parameter block contains the gain and address of the first channel to be sampled, in the format defined in Figure 3-1. The number of channels to be sampled is determined by dividing the length of the user's buffer (END − START + 1) by two. The digitized data is stored in the user buffer sequentially, one halfword per channel.

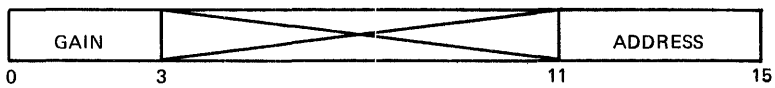

Figure 3-1. Random Field Format

Status Definition

Status	Meaning
X'00'	Normal Completion
X'C0'	Illegal Function Code
X'A0'	Device Unavailable (DU)
X'82'	Recoverable Error, Device Timeout
X'84'	Unrecoverable Device Error (DU During Transfer)
X'86'	Incorrect Address Alignment (START address odd, END address even)

Size Information

Driver Library

Name	Module	Size Bytes
INITAIDV	Analog Input Controller Driver	330
DCB138	DCB/CBB for Analog Input Controller	104

NOTE

This driver only accepts "Random" calls. In keeping with standard INTERDATA driver philosophy, a Random call means that the first address is selected at random and that further addresses are sequential (in the same call). However, this exactly matches Instrumental Society of America's definition of "sequential" analog input. To be consistent with INTERDATA driver philosophy, a Sequential call (if the driver were to accept such a call) means that the Random Address field of the SVC 1 parameter block is unused, and that the system should continue to sample from wherever it left off the previous time. ISA does not require such a mode of operation; therefore, it is not within the scope of this driver. The START address must be on an even address boundary and the END address must be on an odd address boundary, since the analog input controller is a halfword device.

ANALOG OUTPUT CONTROLLER DRIVER

Program Identification

The Analog Output Controller Driver, Program Number 07-083F03 is a part of the Mini I/O System. For further information see the *Analog Output Controller Program Manual,* Publication Number 29-476.

Supported Devices

Device	Product Number	OS/32 Device Code
Analog Output Controller	M48-353	139
	M48-354	139
	M48-355	139

Supported Attributes

Write, Wait/Proceed, Unconditional Proceed, Binary, Image/Format, Sequential/Random, all command functions ignored.

Function Description

One halfword of data is obtained from the user buffer in the format specified in Figure 3-2 and written to the device for conversion, and this procedure is repeated continuously as many times as is necessary to output all the halfwords in the user buffer. The number of halfwords to be output is computed by dividing the length of the user buffer (END − START + 1) by two.

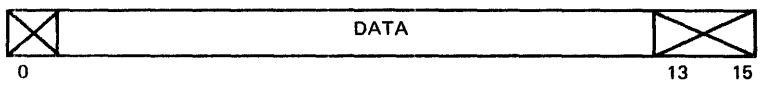

Figure 3-2. Analog Output Data Format

Status Definition

Status	Meaning
X'00'	Normal Completion
X'C0'	Illegal Function
X'86'	Incorrect Address Alignment (START address odd, END address even)

Size Information

Driver Library

Name	Module	Size Bytes
INITAODV	Analog Output Controller Driver	94
DCB139	DCB for Analog Output Controller	78

NOTE

Binary Image is treated identically to Binary Format (i.e., the Image bit is ignored). The Sequential/Random bit is also ignored. The START address must be aligned on an even boundary whereas the END address must be on an odd boundary, since the Analog Output Controller is a halfword device.

DIGITAL INPUT/OUTPUT CONTROLLER DRIVER

Program Identification

The Digital Input/Output Controller Driver, Program Number 07-083F05, is a part of the Mini I/O System. For further information, see *Digital Input/Output Controller Program Manual,* Publication Number 29-477.

Supported Devices

Device	Product Number	OS/32 Device Code
Digital Input/Output Controller	M48-450	140

Supported Attributes

Read (input side only), Write (output side only), Wait/Proceed, Unconditional Proceed, Sequential/Random Binary, Format/Image, all command functions ignored.

Functional Description

The number of transfer is computed using the Start and End Address fields: (END – START +1)/2. Resetting the Sequential/Random bit in the Function Code field causes transfers to continuously occur, without interruption. This is referred to as a non-handshaking transfer mode. In the handshaking transfer mode, the Sequential/Random bit is set and each transfer only occurs after the internal strobe line is pulsed. A time out rate for each transfer is set at a constant of 4 seconds.

Read Binary – The Start Address of the SVC 1 instruction points to a buffer that sequentially stores inputs from the Digital Input card, a halfword per input.

Write Binary – The Start Address points to a buffer (K1) of halfwords consisting of image halfwords for output transfer. The Random Address field of the SVC 1 parameter block points to another buffer (K2) of halfwords designating masks that define what corresponding bit position in K1 is to be changed. The length of K2 must be the same as that of K1. A bit set in K2 indicates that the digital output will be changed to the state defined by the corresponding bit position in K1. The halfwords transferred out to the Digital Output card are computed by the following logical expression:

$$(K1 \cdot K2) + (\overline{K2} \cdot R)$$

where:
. means logical AND
+ means logical OR
$\overline{K2}$ means one's complement of K2
R is the last known content of the output register

Status Definition

Status	Meaning
X'00'	Normal Completion
X'C0'	Illegal Function
X'A0'	Device Unavailable (DU)
X'82'	Recoverable Error, Time Out
X'84'	Unrecoverable Error, DU occured during transfer
X'86'	Incorrect Address Alignment (START address odd, END address even, Random address odd)

Size Information

Driver Library

Name	Module	Size Bytes
INITDIOV	Digital I/O Controller Driver	658
DCB140	DCB/CCB for Digital I/O Controller	104

NOTE

Binary Image is treated identically to Binary Format (i.e., the Formatted/Image bit of the SVC 1 Function Code is ignored by the driver). Since the DCB Record Length field DCB.RECL retains the last known value of the Digital Output Controller output register, it is necessary to reset DCB.RECL to 0 at system initialization (SYSINIT). To do this, Bit 13 of the DCB Flag field has to be set to indicate that DCB.RECL is to be reset at SYSINIT. At any time, if DCB.RECL is to be known, an SVC 7 fetch attributes can be executed to find out the last known value of the Digital Output Controller output register. Although only a single driver exists for Digital Input and Digital Output functions, separate device mnemonics are required for the Input and Output sides of the card since each side is addressable and programmable. Thus a common busy condition between the two is not required. Furthermore, both the input and output sides can be used in either the handshaking or non-handshaking transfer modes. The START and RANDOM addresses must be aligned on the even boundaries, whereas the END address must be aligned on an odd boundary, since the Digital I/O Controller is a halfword device.

MINI I/O FORTRAN RTL ROUTINES

The following subprograms enable the FORTRAN user to access the Mini I/O equipment. These routines are based on the Instrumental Society of America (ISA) Standards for Analog and Digital I/O. However, to simplify their use within the context of the INTERDATA operating systems, there are some deviations from the standards. In particular, an extra parameter is required to pass the Logical Unit to which the Mini I/O equipment is assigned. Furthermore, the returned status of each request is defined for compatability with OS/SVC 1.

Analog Inputs in a Sequential Order

This subroutine inputs halfword data from any number of analog channels in a sequential order. The array j specifies the first channel address to be read and the gain of all channels in the sequence. The form of this call is:

```
CALL AISQW (l,i,j,k,m)
```

where:

l Specifies the integer LU assigned to the analog input controller. This argument is either an INTERGER*4 variable name, array element name, or constant.

i Specifies the integer number of analog channels for input. This argument is either an INTEGER*4 variable name, array element name, or constant.

j Specifies an INTEGER*2 initial channel address of the first sample to be read and the gain of all channels in the sequence. This argument is a variable name or an array element name.

k Designates an INTEGER*2 array into which the converted values are stored. The i input values are sequentially stored into k, one halfword per input. This argument is either an array name, an array element name, or a variable name.

m In hexadecimal, designates an INTEGER*2 status of the request as follows:

0000	Normal Completion
8200	Device Timed-out
A000	Device Unavailable (DU)
8400	Unrecoverable Device Error (DU during Transfer)
8600	Incorrect Address Alignment

This argument is either a variable name or an array element name.

Analog Inputs in Any Sequence

This routine inputs analog data in a random order. The channel addresses to be sampled and the gain for each are specified in an INTEGER*2 array j. These addresses need not be sequential. The form of this call is:

CALL AIRDW (l,i,j.k,m)

where:

l Specifies the integer LU assigned to the analog input controller. This argument is either an INTEGER*4 variable name, array element name, or constant.

i Specifies the integer number of analog channels to be read. This argument is either an INTEGER *4 variable name, array element name, or constant.

j Is an INTEGER*2 array which comprises a list of channel addresses to be sampled together with their respective gains. See Figure 3-1. This argument is an array name, an array element name, or a variable name.

k Designates an INTEGER*2 array into which the converted values are stored. The order of the elements in k corresponds to the order of j. This argument is either an array name, an array element name, or a variable name.

m In hexadecimal, indicates the INTEGER*2 status of the request as follows:

0000	Normal Completion
8200	Device Timed-out
A000	Device Unavailable (DU)
8400	Unrecoverable Error (DU during Transfer)
8600	Incorrect Address Alignment

This argument is either a variable name or an array element name.

Analog Output

This subroutine outputs analog halfword data in the specified sequence. The form of this call is:

CALL AOW (l,i,k,m)

where:

l Specifies the integer LU assigned to the analog output controller. This argument is either an INTEGER*4 variable name, array element name, or constant.

i Specifies the integer number of analog halfword data to be output. This argument is either an INTEGER*4 variable name, array name, or constant.

k Designates an INTEGER*2 array from which the analog output values are taken. The output order corresponds to the order of k. This argument is a variable name, an array name, or an array element name.

m In hexadecimal, indicates the INTEGER*2 status of the request as follows:

0000	Normal Completion
8600	Incorrect Address Alignment

This argument is either a variable name or an array element name.

Digital Input Handshaking Mode

This subroutine inputs halfword data from any number of digital points. Prior to performing the next input, the driver waits for the internal strobe line to be pulsed. The input values are sequentially stored, one halfword per input, in an INTEGER*2 array k. The form of this call is:

CALL DIWH (l,i,k,m)

where:

l Specifies the interger LU assigned to the digital input controller. This argument is either an INTEGER *4 variable name, array element name, or constant.

i Specifies the integer number of external digital halfwords to be read. This argument is either an INTEGER*4 array element name, variable name, or constant.

k Designates an INTEGER*2 array to which the requested values are assigned. This argument is either an array name, a variable name, or an array element name.

m In hexadecimal, indicates the INTEGER*2 status of the request as follows:

0000	Normal Completion
A000	Device Unavailable (DU)
8400	Unrecoverable Error (DU during Transfer)
8200	Device Timed-out
8600	Incorrect Address Alignment

This argument is either a variable name or an array element name.

Digital Input Non-handshaking Mode

This subroutine inputs halfword data from any number of digital points. The input occurs continuously, one halfword per input. The values are stored sequentially in an INTEGER*2 array k. The form of this call is:

CALL DIW (l,i,k,m)

where l,i,k, and m are defined as for Digital Input Handshaking mode.

Digital Output Handshaking Mode

This subroutine outputs digital signals organized in halfword form. The driver waits for the internal strobe lines to be pulsed prior to performing the next output. The form of this call is:

CALL DOLWH (l,i,k_1,k_2,m)

where:

l Specifies the integer LU assigned to the digital output controller. This argument is either an INTEGER *4 variable name, array element name, or constant.

i Specifies the integer number of digital halfwords to be output. This argument is either an INTEGER*4 variable name, array element name, or constant.

k_1 Designates an INTEGER*2 array whose values are the image halfwords for output. This argument is either an array name, an array element name, or a variable name.

k_2 Designates an INTEGER*2 array of halfwords designating masks that define what corresponding bit in k_1 is to be changed. A bit set in the k_2 array indicates that the digital output is to be modfied to the state defined by the corresponding integer array element value in k_1. The order of the elements in k_2 corresponds to the order of k_1. This argument is either an array name, an array element name, or a variable name.

m In hexadecimal, indicates the INTEGER*2 status of the request:

0000	Normal Completion
A000	Device Unavailable (DU)
8400	Unrecoverable Error (DU during Transfer)
8200	Device Timed-out
8600	Incorrect Address Alignment

This argument is either a variable name or an array element name.

Digital Output Non-handshaking Mode

This subroutine is similar to the Digital Output Handshaking mode with the exception that transfers occur continuously, one halfword per output. The form of this call is:

CALL DOLW (l,i,k_1,k_2,m),

where l,i,k_1,k_2, and m are defined as for the Digital Output Handshaking mode.

CHAPTER 4
SYSTEM APPLICATIONS

INTRODUCTION

The Mini I/O System has several application areas depending upon user needs: manufacturing processor control and monitor, medical data acquisition and display, analysis of communication data, performance monitoring of power station generators, and others. Providing examples for each application area is beyond the scope of this manual. However, specific examples are given to demonstrate some of the digital and analog features of the Mini I/O System.

DIGITAL EXAMPLE

The following example demonstrates a ten-second countdown on a single digit numeric display. Seven digital input lines are attached to the visible display device as shown in Figure 4-1.

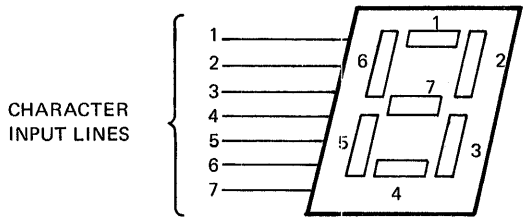

Figure 4-1. 7 Segment Single Digit Numeric Display

Each segment of the numeric display represents a digital input. The resultant displays are shown in Figure 4-2.

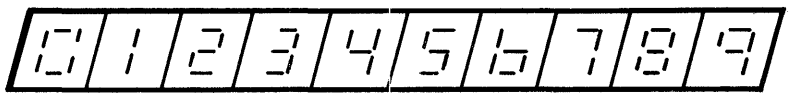

Figure 4-2. 7 Segment Display Coding

Halfwords containing the code necessary to evoke the desired numeric display (Figure 4-3) are prepared for use in the FORTRAN program data table.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
								7	6	5	4	3	2	1	←SEGMENT NO.	
															HEX CODE VALUE	RESULTANT DISPLAY
								1	1	0	0	1	1	1	X'67'	9
								1	1	1	1	1	1	1	X'7F'	8
								0	0	0	0	1	1	1	X'07'	7
								1	1	1	1	1	0	0	X'7C'	6
								1	1	0	1	1	0	1	X'6D'	5
								1	1	0	0	1	1	0	X'66'	4
								1	0	0	1	1	1	1	X'4F'	3
								1	0	1	1	0	1	1	X'5B'	2
								0	0	0	0	1	1	0	X'06'	1
								0	1	1	1	1	1	1	X'3F'	0

Figure 4-3. Halfword Codes for Data Table

Appendix 1 is a FORTRAN program which counts down from 9 to 0 in one second intervals. The Mini I/O RTL routine DOLW is used thus:

```
CALL DOLW (1,1,NUMTBL(II),K2,STATUS)
```

where the first argument passed specifies LU1 which is assigned to the digital output controller, the second argument specifies that one halfword of data is to be output, the third argument NUMTBL(II) is an element in the INTEGER*2 array NUMTBL which contains the hexadecimal code values necessary for displaying the numbers nine through zero, the fourth argument K2 is an INTEGER*2 constant (X'FFFF') which specifies that all bits in NUMTBL(II) are to be output and the last argument STATUS is the halfword status returned from this request.

ANALOG EXAMPLE

Another example demonstrates the use of the Analog Input subroutine AIRDW. The external device is assumed to be a voltage source. The purpose of this program is to detect any variance in the Analog Input voltage source which exceeds a defined value. The program then ouputs the exceeded value and the time of occurrence. The voltage source is assumed to be strapped to channel address of the Analog Input Controller. Appendix 2 demonstrates a FORTRAN program that does this. The Mini I/O RTL routine AIRDW is used thus:

```
CALL AIRDW (1,1,ADDR,INVAL,STATUS)
```

where the first argument sepcifies LU 1 which is assigned to the analog input controller, the second argument specifies that one halfword of data is to be input, the third argument ADDR is an INTEGER*2 constant of eight which is the channel to be sampled, the fourth argument INVAL specifies an INTEGER*2 variable where the input value is stored, and the last argument STATUS contains the halfword status of the request. It is assumed that a 12-bit resolution is used with a range of ± 5.12 volts. Therefore, the input values are shifted right arithmetically to move the least significant bit (bit 12) to bit 15. The expected voltage input is +1.0 volt which is 190 in hexadecimal after the shift. The output data appears in Table 4-1.

TABLE 4-1. OUTPUT DATA

TIME-OF-OCCURRENCE	INPUT VALUE	EXPECTED VALUE
13:28:52	18B	190
13:28:52	18B	190
13:28:53	18B	190
13:28:54	18B	190
13:28:56	18B	190
13:28:56	18B	190
13:28:57	18B	190
13:29:00	18B	190
13:29:02	18B	190
13:29:02	18B	190
13:29:03	18B	190
13:29:10	18B	190
13:29:12	18B	190
13:29:17	18B	190
13:29:21	18B	190
13:29:22	18B	190
13:29:33	18B	190
13:29:38	18A	190
13:29:39	18B	190
13:29:40	18B	190
13:29:44	18B	190
13:29:46	18B	190
13:29:47	18B	190
13:29:49	18B	190
13:29:53	18B	190
13:30:01	18B	190
13:30:19	18B	190
13:30:28	18B	190
13:30:32	18B	190

APPENDIX 1

DIGITAL OUTPUT PROGRAM LISTING

```
                                                                    PAGE 0001

C                                                                   AP100000
C   THIS PROGRAM EXAMPLE DEMONSTRATES THE USE OF DOLW AND IT        AP100010
C   ASSUMES THAT BEFORE EXECUTION, LOGICAL UNIT 1 IS                AP100020
C   ASSIGNED TO THE DIGITAL OUTPUT CONTROLLER.                      AP100030
C                                                                   AP100040
      INTEGER*2 NUMTBL(10),K2,STATUS,INIT                           AP100050
      DATA NUMTBL/X'67',X'7F',X'07',X'7C',X'6D',X'66',X'4F',        AP100060
     1     X'5B',X'06',X'3F'/                                       AP100070
      DATA K2/X'FFFF'/                                              AP100080
      INIT=0                                                        AP100090
      CALL DOLW(1,1,INIT,K2,STATUS)                                 AP100100
      DO 20 II=1,10                                                 AP100110
      CALL DOLW(1,1,NUMTBL(II),K2,STATUS)                           AP100120
      CALL WAIT(1,2,STATUS)                                         AP100130
20    CONTINUE                                                      AP100140
      STOP                                                          AP100150
      END                                                           AP100160
.U      EXT FUNC
NUMTBL  INT2 VAR
K2      INT2 VAR
STATUS  INT2 VAR
INIT    INT2 VAR
DOLW    EXT FUNC
20      LABEL
II      INT4 VAR
WAIT    EXT FUNC
.S      EXT FUNC
.V      EXT FUNC

0000 ERRORS:  FORTRAN V LEVEL 1 R02.02A
```

APPENDIX 2

ANALOG INPUT PROGRAM LISTING

```
                                                                    PAGE 0001

C                                                                          AP200000
C  THIS PROGRAM EXAMPLE DEMONSTRATE THE USE OF AIROW.  BEFORE             AP200010
C  EXECUTION, LU 1 MUST BE ASSIGNED TO THE ANALOG INPUT CONTROLLER,        AP200020
C  AND LU 6 MUST BE ASSIGNED TO AN OUTPUT DEVICE SUCH AS THE PRINTER       AP200030
C                                                                          AP200040
      INTEGER*2 INVAL,ADDR,VOLT,DIFF,STATUS                                AP200050
      DOUBLE PRECISION CLOCK                                               AP200060
      DATA VOLT/X'190'/                                                    AP200070
      ADDR=8                                                               AP288888
      WRITE(6,20)                                                          AP200090
10    CALL TIME(CLOCK,ISTAT)                                               AP200100
      CALL AIROW(1,1,ADDR,INVAL,STATUS)                                    AP200110
      INVAL=ISHFT2(INVAL,-3)                                               AP200120
      DIFF=IABS2(INVAL-VOLT)                                               AP200130
      IF (DIFF .LE. 4) GO TO 10                                            AP200140
      WRITE(6,30)CLOCK,INVAL,VOLT                                          AP200150
      GO TO 10                                                             AP200160
20    FORMAT (1H1,'TIME-OF-OCCURRENCE  INPUT VALUE   EXPECTED VALUE',/)    AP200170
30    FORMAT(1H ,5X,A8,10X,Z4,12X,Z4)                                      AP200180
      END                                                                  AP200190
.U      EXT FUNC
INVAL   INT2 VAR
ADDR    INT2 VAR
VOLT    INT2 VAR
DIFF    INT2 VAR
STATUS  INT2 VAR
CLOCK   DBLE VAR
20      LABEL
@H      EXT FUNC
10      LABEL
TIME    EXT FUNC
ISTAT   INT4 VAR
AIROW   EXT FUNC
ISHFT2  EXT FUNC
IABS2   EXT FUNC
30      LABEL
.V      EXT FUNC

0000 ERRORS:  FORTRAN V LEVEL 1 R02.02A
```

APPENDIX 3

MAGNETIC AND DISC MEDIA TABLE OF CONTENTS

MAGNETIC MEDIA:

DCB138 INITAIDV DCB139 INITAODV DCB140 INITDIOV	Mini I/O Drivers (Object) R00
— FM —	
AISQW AIRDW AOW DIW DIWII DOLW DOLWII	Mini I/O FORTRAN RTL (Object) R00
— FM —	
— FM —	
**DCB138 **INITAIDV **DCB139 **INITAODV **DCB140 **INITDIOV	Mini I/O Drivers (Source) R00
— FM —	
— FM —	

DISC MEDIA:

Volume OSMT

FILE NAME	TYPE	LENGTH	KEYS	NLR	DESCRIPTION
MIODVR.OBJ	IN	126	0	15	Mini I/O drivers object
MIODVR.CAL	IN	80	0	1FF	Mini I/O driver source
MIORTL.OBJ	IN	126	0	D	Mini I/O RTL routines object

PUBLICATION COMMENT FORM

Please use this postage-paid form to make any comments, suggestions, criticisms, etc. concerning this publication.

From _____ Date _____

Title _____ Publication Title _____

Company _____ Publication Number _____

Address _____

_____

_____

FOLD FOLD

Check the appropriate item.

☐ Error Page No. _____ Drawing No. _____

☐ Addition Page No. _____ Drawing No. _____

☐ Other Page No. _____ Drawing No. _____

Explanation:

FOLD FOLD

CUT ALONG LINE

Fold and Staple

No postage necessary if mailed in U.S.A.

6434

STAPLE

STAPLE

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 22 OCEANPORT, N.J.

POSTAGE WILL BE PAID BY ADDRESSEE

PERKIN-ELMER

Computer Systems Division
2 Crescent Place
Oceanport, NJ 07757

TECH PUBLICATIONS DEPT. MS 322A

FOLD

FOLD

STAPLE

STAPLE

6433