```
URI1:   ;USER ROUTINE INITIALISE ROUTINE.
        ;THIS ROUTINE IS CALLED BY THE BADEDAS COMMAND
        ;INTERPRETER WHENEVER AN INITIALISE COMMAND IS
        ;RECEIVED FOR URI ( USER ROUTINE 0 ).
        ;ON ENTRY AC2 POINTS TO THE COMMAND BLOCK.
        ;
        SAV                     ;SAVE ACCUMULATORS
        LDA     AC3,1,AC2       ;AC3 = NO. OF BYTES IN COMMAND BLOCK
        LDA     AC0,FOUR        ;AC0 = 4
        SGE     AC3,AC0         ;HAS AT LEAST 1 PARM BEEN PROVIDED?
        JMP     UIRET           ;NO - SO RETURN
        LDA     AC0,4,AC2       ;YES - AC0 = PARM 0
        STA     AC0,A           ;A = PARM 0
        LDA     AC0,SIX         ;AC0 = 6
        SGE     AC3,AC0         ;HAVE AT LEAST 2 PARMS BEEN PROVIDED?
        JMP     UIRET           ;NO - SO RETURN
        LDA     AC0,5,AC2       ;YES - AC0 = PARM 1
        STA     AC0,B           ;B = PARM 1
        LDA     AC0,EIGHT       ;AC0 = 8
        SGE     AC3,AC0         ;HAVE AT LEAST 3 PARMS BEEN PROVIDED?
        JMP     UIRET           ;NO - SO RETURN
        LDA     AC0,6,AC2       ;YES - AC0 = PARM 2
        STA     AC0,C           ;C = PARM 2
        LDA     AC0,TEN         ;AC0 = 10
        SGE     AC3,AC0         ;HAVE AT LEAST 4 PARMS BEEN PROVIDED?
        JMP     UIRET           ;NO - SO RETURN
        LDA     AC0,7,AC2       ;YES - AC0 = PARM 3
        STA     AC0,D           ;D = PARM 3
UIRET:  RET                     ;RETURN
```

APPENDIX D

BADEDAS FORTRAN EXAMPLE

The following Fortran 5 program makes use of BADEDAS subroutine calls to do the following:

* to initialise the A/D converter to operate in data channel mode with an external clock, to sample from channel 1 and to have a input word count of 1024.

* to set up a BADEDAS pipe between the A/D converter and the host computer, which includes the user routine 1, and sets the block count to 1.

* to read the block input by the A/D converter from the Micronova.

The program finally writes the block into an AOS disk file.

```
INTEGER IARRAY(1024),ARGARRAY(2)
OPEN 20,"FILE.DAT",ATT="RO",LEN=2048
ARGARRAY(1)=1024
ARGARRAY(2)=170000K
CALL BADI("AD0",ARGARRAY,2)
CALL BADP("AD0","UR1",HD0,1)
CALL BADR(IARRAY,1024,ISIZE)
WRITE BINARY (20) IARRAY
CLOSE 20
END
```

DEPARTMENT OF ELECTRICAL ENGINEERING

BADEDAS - USER MANUAL

University of the
Witwatersrand
Johannesburg

TABLE OF CONTENTS

TABLE OF CONTENTS

1.0 SCOPE

This manual provides the information necessary to use BADEDAS (Basic Analog and Digital Experimental Data Acquisition System). The aim of this manual is to explain the functions provided by BADEDAS and the user interaction needed to initiate these functions.

2.0 INTRODUCTION

BADEDAS is a simple but powerful data acquisition system, which runs on a system consisting of an AOS based Eclipse minicomputer front-ended by a Micronova microcomputer. All user interaction is conducted from the Eclipse, while the Micronova is responsible for the real-time acquisition (input) of digital samples of analog signals as well as real-time dissemination (output) of digital samples in the form of analog signals. The BADEDAS user interface software, which runs on the Eclipse, automatically down-line loads the BADEDAS control software for the Micronova front-end.

BADEDAS provides two types of interface to the user. The first is an AOS CLI-type interface, which allows the user to control BADEDAS using simple commands entered from an Eclipse user console in the same way as one would enter AOS CLI commands. The second gives the user a means of controlling BADEDAS using subroutine calls issued from a Fortran 5 program running on the Eclipse.

BADEDAS handles data in units of variable length data blocks. All input, output and data transfers in BADEDAS are done in terms of data blocks. Input and output devices in BADEDAS are called source and destination devices. BADEDAS source devices input data blocks and BADEDAS destination devices output data blocks. Data block sizes are specified by the user when initialising the source device. A data block may be any size from 1 to 24 000 words. A fixed overhead is associated with the handling of each block, so it is inefficient to specify small data block sizes (ie.less than 100 words). Typically, blocksizes should be between 256 and 4096 words.

All BADEDAS data acquisition and data dissemination is controlled from the Micronova front-end. The acquisition of digital samples of analog signals is achieved using an analog-to-digital (A/D) converter, which can input up to 25 000 digital samples per second. The dissemination of digital samples as analog signals is achieved using a digital-to-analog

(D/A) converter, which can convert in excess of 25 000 samples per second.

The primary function provided by BADEDAS is the ability to set up a virtual connection between a source and a destination device. This connection, called a BADEDAS pipe, allows blocks of data input by a source device to pass to the designated destination device for subsequent output.

The standard BADEDAS source devices include the A/D converter and the host computer (ie. the Eclipse) and standard destination devices include the D/A converter and the host computer. Each BADEDAS device is allocated a three letter mnemonic, which is used to refer to that device in BADEDAS commands and subroutine calls. Internally the BADEDAS system uses a one byte code, called the BADEDAS internal code, for each device. (see table 1)

NOTE

In table 1 the host computer has three different source device mnemonics. This enables the host computer to be the source device in up to three different pipes simultaneously. In order to take advantage of this feature, additional destination devices need to be installed. For example, in a BADEDAS system with two D/A converters, a Fortran 5 user program could set up a pipe between the host and each of the D/A converters, using a different mnemonic for the host in each of the pipes. Each time the user program wishes to send a block of data through one of the pipes, its BADEDAS call would then specify the host computer source mnemonic associated with that particilar pipe.

BADEDAS also provides a facility by which a user-written routine may be linked into the BADEDAS Micronova software. Thereafter, the user routine may be included in BADEDAS pipes, between the source and destination devices. This facility allows for limited processing to be performed on blocks of digital samples en route from the source to the destination device. A typical application of a user routine would be to implement a digital filter. Up to three user-written routines may be linked into BADEDAS at any time. As with BADEDAS devices, each user routine has a three byte mnemonic and a one byte internal code (see table 1).

TABLE 1: BADEDAS MNEMONIC AND INTERNAL CODE TABLE

MNEMONIC	INT CODE	DESCRIPTION
AD0	"A"	A/D converter
AD1	"B"	A/D converter 1 (not installed)
-	"C"	Spare for extra source device
-	"D"	Spare for extra source device
HS0	"E"	Host source 0
HS1	"F"	Host source 1
HS2	"G"	Host source 2
UR0	"H"	User routine 0
UR1	"I"	User routine 1
UR2	"J"	User routine 2
DA0	"K"	D/A converter 0
DA1	"L"	D/A converter 1 (not installed)
-	"M"	Spare for extra dest. device
-	"N"	Spare for extra dest. device
HD0	"O"	Host destination

3.0 PREPARATIONS

The following preparations and checks should be carried out before running BADEDAS:

1. The Eclipse console port, to which the Micronova is connected, must be dropped from the operating system's list of user consoles. This is done using the AOS CLI macro command DROP.

   eg. DROP @CON2

2. Ensure that there is a correctly wired cable joining the relevant Eclipse console port to the first teletype port of the Micronova.

3. Ensure that the baud rate and serial transmission format of the Micronova TT0 port and the Eclipse console port are the same. The baud rate should not exceed 9600. The format must allow for 8 bit data without a parity bit and the number of stop bits must be the same for both ports.

4. If BADEDAS message logging is required, ensure that a terminal is connected to the second teletype port, TT1.

5. Ensure that the Micronova is powered-up and in console debug mode. This is achieved by turning the power key on the Micronova front panel to the "RESET" and then the "RUN" position.

4.0 BADEDAS COMMAND INTERFACE

The quickest and most simple method of using BADEDAS is by way of the BADEDAS command interface. This interface provides the user with four commands which may be entered as command lines from the user console:

1. the initialise command

2. the pipe command

3. the terminate command

4. the reset command

NOTE

When entering BADEDAS commands, numeric parameters must be entered as positive decimal or octal integers. An octal integer is a numeric string consisting of the digits 0 to 7 and followed by the letter "O".

eg. 1234560

Decimal integers must be in the range 0 to 65535 and octal integers must be in the range 0 to 177777 octal.

4.1 INITIALISE COMMAND

DESCRIPTION:

This command initialises a device or user routine as required by the the user. It is used to set up devices, specify blocksizes, provide values of coefficients for user routines, etc.

FORMAT:

X BAD/I mnemonic,argument 1,...,argument N

ARGUMENTS:

* mnemonic - this argument must be the BADEDAS mnemonic of the device or user routine to be initialised.

* arguments 1 to N - these arguments are the values used to initialise the device or user routine. Their order and values depend on the device or user routine being initialised.

NOTES:

If the device is the A/D converter, argument 1 will be the word count and argument 2 will be the A/D mode word (see 6.1). If the device is the D/A converter, argument 1 will be the D/A mode word. If the arguments are omitted, their default values are used.

EXAMPLES:

X BAD/I AD0,2048,170000

This command initialises the A/D converter, setting the word count to 2048 and using the a mode word of 170000 octal.

X BAD/I DA0,150000

This command initialises the D/A converter, setting the mode word to 150000 octal.

4.2 PIPE COMMAND

DESCRIPTION:

This command sets up and starts a pipe operation between a BADEDAS source device or AOS disk file and a BADEDAS destination device or AOS disk file.

FORMAT:

X BAD/P source{,user-routine},destination{,block-count}

ARGUMENTS:

* source - this argument must be either the BADEDAS mnemonic of a source device or the name of an existing AOS disk file. If the pipe source is an AOS disk file, this argument should include the blocksize as a switch, unless the default blocksize (1024 Words) is to be used.

* user-routine - this argument is optional. If included it must be one of UR0, UR1 or UR2, depending on which user routine is to be incorporated in the pipe.

* destination - this argument must be either the BADEDAS mnemonic of a destination device or the name of an AOS file, which will be created.

* block-count - this argument is optional. If included it must be a positive decimal or octal integer.

NOTES:

1. It is invalid to specify both the pipe source and destination as AOS disk files.

2. Before issuing a pipe command, the pipe source must be initialised, unless the pipe source is to be an AOS disk file, in which case the initialisation is done automatically.

EXAMPLES:

X BAD/P AD0,UR1,DA0,2

This command sets up a BADEDAS pipe with the A/D converter as its source, user routine UR1, the D/A converter as its destination and a block count of 2.

```
X BAD/P DATA.FILE/2048,DA0
```

This command sets up a BADEDAS pipe with the AOS disk file DATA.FILE as its source and the D/A converter as its destination. Since no block count is specified, the transfer of data will continue until the end-of-file is encountered. The switch on the file name argument sets the length of blocks transferred from the file to the D/A to 2048 words in length.

4.3 TERMINATE COMMAND

DESCRIPTION:

This command is used to terminate a pipe operation before it has run to completion.

FORMAT:

X BAD/T source

ARGUMENTS:

The only argument for the terminate command is the pipe source device's BADEDAS mnemonic.

EXAMPLES:

X BAD/T AD0

NOTES:

Only a source device may be terminated. This command is intended for terminating pipe commands involving a number of blocks and is only effective if more than one block remains to be input by the pipe source device, otherwise it is ignored.

4.4 RESET COMMAND

DESCRIPTION:

This command resets the BADEDAS micronova control software by causing it to execute a halt instruction. The next time communication takes place between the BADEDAS command interface and the Micronova front-end, the command interface finds the Micronova in this state and proceeds to down-line load the BADEDAS control software for it.

FORMAT:

X BAD/R

ARGUMENTS:

None

EXAMPLE:

X BAD/R

5.0 THE BADEDAS FORTRAN 5 INTERFACE

The BADEDAS Fortran 5 interface provides the user with a more powerful and more flexible means of using BADEDAS. This interface provides six BADEDAS subroutine calls which may be included in the user's Fortran 5 program :

1. The Initialise Call - functionally identical to the BADEDAS initialise command.

2. The Pipe Call - similar to the BADEDAS pipe command except when the host computer is either the pipe's source or destination.

3. The Read Call - when the host computer is the destination device in a pipe command, the program running on the host computer must issue a read call for each block of data when it is ready to input that block.

4. The Write Call - when the host is the source device in a pipe command, the program running on the host must issue a write call for each block of data it wants to send to the front-end.

5. The Terminate Call - functionally identical to the BADEDAS terminate command.

6. The Reset Call - functionally identical to the BADEDAS reset command.

An example of a Fortran 5 program, which incorporates BADEDAS calls is given in Appendix A.

5.1 BADI - BADEDAS Initialise Call

FORMAT:

```
CALL BADI (unit-mnemonic,arg-array,arg-count,
           source-mnemonic)
```

ARGUMENTS:

* unit-mnemonic - the 3 letter BADEDAS mnemonic of a BADEDAS I/O device or user routine.

* arg-array - an integer array holding the initialisation parameters (each parameter occupies 1 16-bit word).

* arg-count - the number of initialisation parameters in the array.

* sender-mnemonic - the mnemonic of the source device making the call. ie. HS0, HS1 or HS2.

EXAMPLES:

```
CALL BADI ("AD0",ARGARRAY,2,"HS0")
```

This call causes the A/D converter to be initialised. Two parameters are provided and they are stored in array ARGARRAY. The first parameter will be the blocksize and the second the mode word.

```
CALL BADI ("UR1",UARRAY,3,"HS0")
```

This call causes the user routine UR1 to be initialised. Three parameters ar provided and they are stored in array UARRAY.

NOTES:

This call causes an initialise command to be sent to the BADEDAS Micronova control software. All pipe source devices should be initialised before issuing a pipe command. If a BADI call is made with no initialisation parameters, the unit will be initialised using its default parameters. Once a user routine or a destination device has been initialised, it will maintain those characteristics given to it by the initialise command, until it is initialised by another initialise command.

5.1 BADI - BADEDAS Initialise Call

FORMAT:

```
CALL BADI (unit-mnemonic,arg-array,arg-count,
           source-mnemonic)
```

ARGUMENTS:

* unit-mnemonic - the 3 letter BADEDAS mnemonic of a BADEDAS I/O device or user routine.

* arg-array - an integer array holding the initialisation parameters (each parameter occupies 1 16-bit word).

* arg-count - the number of initialisation parameters in the array.

* sender-mnemonic - the mnemonic of the source device making the call. ie. HS0, HS1 or HS2.

EXAMPLES:

```
CALL BADI ("AD0",ARGARRAY,2,"HS0")
```

This call causes the A/D converter to be initialised. Two parameters are provided and they are stored in array ARGARRAY. The first parameter will be the blocksize and the second the mode word.

```
CALL BADI ("UR1",UARRAY,3,"HS0")
```

This call causes the user routine UR1 to be initialised. Three parameters are provided and they are stored in array UARRAY.

NOTES:

This call causes an initialise command to be sent to the BADEDAS Micronova control software. All pipe source devices should be initialised before issuing a pipe command. If a BADI call is made with no initialisation parameters, the unit will be initialised using its default parameters. Once a user routine or a destination device has been initialised, it will maintain those characteristics given to it by the initialise command, until it is initialised by another initialise command.

5.1 BADI - BADEDAS Initialise Call

FORMAT:

```
CALL BADI (unit-mnemonic,arg-array,arg-count,
           source-mnemonic)
```

ARGUMENTS:

* unit-mnemonic - the 3 letter BADEDAS mnemonic of a BADEDAS I/O device or user routine.

* arg-array - an integer array holding the initialisation parameters (each parameter occupies 1 16-bit word).

* arg-count - the number of initialisation parameters in the array.

* sender-mnemonic - the mnemonic of the source device making the call. ie. HS0, HS1 or HS2.

EXAMPLES:

```
CALL BADI ("AD0",ARGARRAY,2,"HS0")
```

This call causes the A/D converter to be initialised. Two parameters are provided and they are stored in array ARGARRAY. The first parameter will be the blocksize and the second the mode word.

```
CALL BADI ("UR1",UARRAY,3,"HS0")
```

This call causes the user routine UR1 to be initialised. Three parameters are provided and they are stored in array UARRAY.

NOTES:

This call causes an initialise command to be sent to the BADEDAS Micronova control software. All pipe source devices should be initialised before issuing a pipe command. If a BADI call is made with no initialisation parameters, the unit will be initialised using its default parameters. Once a user routine or a destination device has been initialised, it will maintain those characteristics given to it by the initialise command, until it is initialised by another initialise command.

To inform BADEDAS of the size of blocks originating from a user program, the host source interface (HS0, HS1 or HS2) must be initialised with the blocksize (in 16 bit words).

5.2 BADP - BADEDAS Pipe Call

FORMAT:

```
CALL BADP (source-mnemonic,user-routine-mnemonic,
           destination-mnemonic,block-count,
           sender-mnemonic)
```

ARGUMENTS:

* source-mnemonic - the mnemonic of the pipe source device

* user-routine-mnemonic - the mnemonic of the user routine

* destination-mnemonic - the mnemonic of the destination device

* block-count - the number of blocks to pass along the pipe

* sender-mnemonic - the mnemonic of the sender of the pipe command (eg. HS0,HS1 or HS2).

EXAMPLES:

```
CALL BADP ("HS0","UR1","DA0",1,"HS0")
```

This call sets up a BADEDAS pipe with the host as source, user routine UR1, the D/A converter as destination and a block count of 1.

```
CALL BADP ("AD0","","DA0",10,"HS1")
```

This call sets up a BADEDAS pipe with the A/D converter as host, no user routine, the D/A converter as destination and a block count of 10.

NOTES:

This call causes a pipe command to be sent to the BADEDAS Micronova control software. The pipe source device and, if necessary, the pipe destination device and the user routine

5.2 BADP - BADEDAS Pipe Call

FORMAT:

```
CALL BADP (source-mnemonic,user-routine-mnemonic,
           destination-mnemonic,block-count,
           sender-mnemonic)
```

ARGUMENTS:

* source-mnemonic - the mnemonic of the pipe source device

* user-routine-mnemonic - the mnemonic of the user routine

* destination-mnemonic - the mnemonic of the destination device

* block-count - the number of blocks to pass along the pipe

* sender-mnemonic - the mnemonic of the sender of the pipe command (eg. HS0,HS1 or HS2).

EXAMPLES:

```
CALL BADP ("HS0","UR1","DA0",1,"HS0")
```

This call sets up a BADEDAS pipe with the host as source, user routine UR1, the D/A converter as destination and a block count of 1.

```
CALL BADP ("AD0","","DA0",10,"HS1")
```

This call sets up a BADEDAS pipe with the A/D converter as host, no user routine, the D/A converter as destination and a block count of 10.

NOTES:

This call causes a pipe command to be sent to the BADEDAS Micronova control software. The pipe source device and, if necessary, the pipe destination device and the user routine

should be initialised, using BADI calls, prior to making a BADP call. If no user routine is to be included in the pipe, then this must be indicated by the use of a null string, "", in place of the user-routine-mnemonic.

5.3 BADR - BADEDAS Read Call

FORMAT:

```
CALL BADR (array,array-size,data-size,
           sender-mnemonic)
```

ARGUMENTS:

* array - the name of the array to accept the block of data.
* array-size - the size in 16-bit words of the array.
* data-size - returned by the call - the number of 16-bit words of data read.
* sender-code - returned by call - the BADEDAS internal code of the source device from which the data block originated.

EXAMPLE:

```
CALL BADR ( IARRAY,1024,IASIZE,RNEM)
```

NOTES:

After a pipe command has been sent to BADEDAS, specifying the host computer as the destination device, BADR calls may be made to read each block of data from the Micronova front-end into the specified array.

5.4 BADW - BADEDAS Data Block Write Subroutine Call

FORMAT:

CALL BADW (array,data-size,sender-mnemonic)

ARGUMENTS:

1. array - the name of the array holding the data to be sent

2. data-size - the number of 16 bit words of data in the array

3. sender-mnemonic - the mnemonic of the source of the data block (ie. HS0, HS1 or HS2).

EXAMPLE:

```
CALL BADW (IARRAY,1024,"HS1")
```

NOTES:

After a pipe command has been sent to BADEDAS, which specifies the host computer as a source device, the BADW call is used to transmit data from the specified array to the Micronova front-end.

5.5 BADT - BADEDAS Terminate Call

FORMAT:

CALL BADT (device-mnemonic,sender-mnemonic)

ARGUMENTS:

* device-mnemonic - the mnemonic of the device which is the source of the pipe to be terminated.

* sender-mnemonic - the mnemonic of the sender of the terminate command.

EXAMPLES:

```
CALL BADT ("AD0","HS0")
```

NOTES:

Only a source device may be terminated. This call is intended for terminating pipe operations involving a number of blocks and is only effective if more than one block remains to be input by the pipe source device, otherwise it is ignored.

5.6 BADX - BADEDAS Reset Call

FORMAT:

CALL BADX (sender-mnemonic)

ARGUMENTS:

* sender-mnemonic - the mnemonic of the sender of the reset command.

EXAMPLE:

```
CALL BADX ("HS0")
```

NOTES:

This call causes the BADEDAS Micronova control software to be reset by making it execute a halt instruction. The next time communication needs to take place with the Micronova, the Micronova is found to be in this state, so the BADEDAS control software is automatically down-line loaded.

6.0 A/D AND D/A CONVERTERS

A detailed coverage of the Micronova A/D and D/A converters is provided in a Data General programmer's reference manual called "Micronova Sensor Interfaces - Models 4222, 4223 and 4224". A brief coverage of the aspects of these devices, which are of relevance to BADEDAS is given here.

6.1 Model 4223 Analog To Digital Converter

The model 4223 A/D converter interface allows the Micronova to input digitised data from external analog signals. It operates with jumper-selectable voltage ranges of 0 to 5v, 0 to 10v, -5 to +5v and -10 to +10v and provides data with a resolution of 12 bits. The data input by the A/D converter can be selected by jumper on the controller board to be in either offset binary or two's complement format. The format of each word of input data is as follows:

BITS	NAME	FUNCTION
0	sign	in two's complement: 0=pos, 1=neg in offset binary: 1=pos, 0=neg
1 to 11	magnitude	11 bit digital representation of analog signal that was converted
12 to 15		reserved

In order to use the A/D converter in BADEDAS, being a source device, it must always first be initialised. There are two parameters which may be provided by the user in this initialisation, the data blocksize (or word count in DG terminology) and the mode word. The parameters must be provided in this order. If the user provides does not provide a value for one or both of the parameters the default value or values are used. The default value for the blocksize is 1024 words and the default mode word is 150000 octal.

The blocksize is the number of words of data to be input by the A/D converter and must be provided as a positive octal or decimal integer. The mode word is a control word, which is loaded directly into the A/D converter's mode register. The mode word, which must be supplied as an octal integer, has the following format: