Fermilab

TM-1287
2320.000

FERMILAB MAGNET DEVELOPMENT AND TEST FACILITY GPIB PACKAGE

Kevin McGuire

November 1984

CONTENTS

1.0 Abstract

This package has been developed for communicating with devices on the GPIB (General Purpose Interface Bus - IEEE Std. 488). Programs and subroutines which call the routines in this package communucate with devices on the GPIB via a GPIB controller residing in a CAMAC crate. CAMAC access is done in 16-bit transfers. The target hardware is a DEC VAX 730 which communicates with a byte-serial CAMAC loop through a Jorway Model 411 PDP-11 CAMAC Interface. CAMAC in turn, communicates with the GPIB through a Kinetic Systems Model 3388-G1A GPIB Interface. Although this is the hardware that is used at our installation, the software is designed to be used on other systems with a minimum of modifications.

2.0 GPIB Routine Overview

Primitives	Mnemonic/Function
GPATN	Enables AtTeNtion
GPDATN	Disables AtTeNtion
GPIFC	Performs an "InterFace Clear"
GPREN	Remote ENable
GPDREN	Disable Remote ENable
GPREAD	READs the Input Register of the controller as a binary integer
GPREADCH	READs the Input Register of the controller as a CHaracter
GPWRITE	WRITEs a binary integer to the Output Register of the controller
GPWRITECH	WRITEs a CHaracter to the Output Register of the controller
GPSTATUS	Reads the controller's Status Register
GPUAI	Enables the Universal and Addressed Command Inhibit - This command is used when the Kinetic Systems Model 3388 GPIB Interface is being used as the system controller. (I believe this command is unique to this particular controller). It affects the handshake.
GPDUAI	Disables the Universal and Addressed Command Inhibit - Use this when the 3388 is not the system controller.

GPIB Routine Overview (cont'd.):

Universal and Addressed Commands	Mnemonic/Function
GPLLO	Local Lock-Out
GPGET	Group Execute Trigger
GPSPE	Serial Poll Enable
GPSPD	Serial Poll Disable

General-Purpose Routines	Function
GPUNT	Untalk
GPUNL	Unlisten
GPUTALK	Sets a talker
GPULISTEN	Sets a listener
GPXMIT	Transmits an array
GPXMITSTR	Transmits a character string
GPRDAT	Reads an array
GPRDATSTR	Reads a character string
GPSPSTAT	Reads Serial Poll Status Words
GPPOLL	Polls devices
GPTRIG	Selectively triggers
GPSRQ	Waits for a Service Request

3.0 System Overview

The following section examines details which are unique to the system under which this package was developed. While it describes the way we have implemented these routines, it is not the only way that the routines in this package may be used. An attempt has been made to make this package flexible and general enough so that transporting it to other systems can be done with a minimum number (if any) of modifications.

The hardware that we use on our system consists of a VAX 730 computer which communucates with the CAMAC crates through a Jorway Model 411 PDP-11 CAMAC Interface. We use a special driver which handles the "Jorway"s as any other I/O device. The software interface between the FORTRAN routines in this package and CAMAC is provided by ANSI/IEEE Std758-1979 standard CAMAC routines. All of the GPIB routines transfer data to/from CAMAC using 16-bit transfers. This makes the package compatible with computers which have data busses less than 24 bits wide (most noteable PDP-11s) and it also provides increased speed and efficiency (over 24-bit transfers).

This package was designed for use at the Fermilab Development and Test Facility for use in programs which are used to test Tevatron I magnets. The overall scheme for testing these magnets requires that all hardware used in measurements will be entered into data bases. Information stored in these data bases will include the readable IDs of the particular piece of hardware, the FNAL IDs (an ID given to it by the laboratory for purposes of inventory), the calibration coefficients, and the Branch, Crate, slot Number, subAddress, and Function code for all CAMAC hardware. All records in these data bases are date and time-stamped so that a corelation can be made between data collected and which hardware was used to collect the data.

All measurements go through an initialization phase where the data bases are queried (using a Digital Equipment Corporation database query package known as DATATRIEVE) to determine which pieces of hardware should be used for the measurement and then the hardware itself is queried for its IDs. If the IDs do not match, the mis-match is flagged and the measurement will not proceed until either the proper piece of hardware is used (or the reason that it was read improperly is solved) or the data bases are repaired. (Sometimes entries in the data bases are made incorrectly.)

All of the measurement programs are written in FORTRAN. As the databases are queried through FORTRAN routines which interface with DATATRIEVE, COMMON blocks are filled which contain the information necessary to do things with the hardware.

The relevance which the preceeding discussion has with the GPIB package is that the CAMAC Branch, Crate, slot Number, subAddress, and Function codes to be used in controlling the GPIB bus are all entered into a data base known as CAMAC_OPERATION. The flexibility that this offers is that any GPIB controller may be used through CAMAC without the need to re-write all of the GPIB code. The GPIB controller we are using is the Kinetic Systems Model 3388-G1A GPIB Interface Module. If a different GPIB controller module were to be installed in its place, only the data base called CAMAC_OPERATION would need to be changed. No editing and re-compiling would need to be done to the source code.

For those who do not have DATATRIEVE or have it, and don't wish to use it for the same purposes which we use it, a block of DATA statements is provided which has all of the necessary information hard-coded. If your GPIB controller uses subaddresses (for different GPIB functions) that are different from the controller that we use, you must consult the manual for your controller and edit the file containing the DATA statements to make the necessary corrections. Those operations which run under VAX/VMS FORTRAN 77 may use the INCLUDE statement to easily insert these parameters into COMMON blocks at compile-time. Otherwise you will have to edit these into the module which declares the COMMON block.

4.0 CAMAC Interface

Since the GPIB package is so intimately tied to CAMAC, the Fermilab Magnet Development and Test Facility CAMAC Package is distributed along with it. This package adheres to IEEE Std. 758-1979 FORTRAN calling conventions.

For those installations without DATATRIEVE:

Since the purpose of DATATRIEVE at our installation is basically to fill COMMON blocks with values obtained in out databases, the use of DATATRIEVE is not central to the application of this GPIB package as long as the COMMON blocks are filled somehow. To this end, a subroutine (GPIBFILL) has been provided to fill these COMMON blocks. In order to fill these COMMON blocks, GPIBFILL has an INCLUDE statement which INCLUDEs a text module called GPIBPARAM. If you are using the same GPIB controller as we are (the Kinetic Systems Model 3388-G1A GPIB Interface Controller), you may use this text module (GPIBPARAM) without any changes. If you are using a different GPIB controller, you may need to edit GPIBPARAM to reflect different function codes and subaddresses required to perform different GPIB interface functions. Needless to say (but easy to overlook), your program must call GPIBFILL before trying to make any GPIB calls.

For those installations wishing to use DATATRIEVE:

If you wish to use DATATRIEVE, you have two choices. You may set up your own data base and write your own routines to fill the COMMON block HDW_GPIB, or you may duplicate out system exactly and use our FORTRAN DATATRIEVE access routines.

5.0 GPIB Routines

5.1 Primitive routines

The primitive routines make the interface between CAMAC and the GPIB controller. They exist for the sole purpose of doing CAMAC accesses. Therefore, the user can call GPIB routines without even thinking about CAMAC. All GPIB routines can be built from these primitives. In fact, all of the "non-primitive" routines (including the Universal and Addressed commands) are built from these primitives. There are also two "elaborated" primitive routines. These are GPREADCH and GPWRITECH. These do the same function as GPREAD and GPWRITE respectively, with the exception that the data is transferred as an ASCII character, rather than as an integer.

5.1.1 GPATN

Function:

Sets the GPIB bus in ATTENTION mode.

Calling Sequence:

```
CALL GPATN ( )
```

Input/Output:

Inputs (from the COMMON block HDW_GPIB) :

GPIB_ATN_ADDR	INTEGER*4	CAMAC address - includes Software Channel, Branch, Crate, and subAddress
GPIB_ATN_FUNC	INTEGER*4	CAMAC Function code to Enable Attention
G_TRY	INTEGER*4	The number of times to loop for a Q response

5.1.2 GPDATN

Function:

Disables ATteNtion on the GPIB bus.

Calling Sequence:

```
GPIBSTAT = GPDATN ( )
```

Input/Output:

Inputs (from the COMMON block HDW_GPIB) :

GPIB_DATN_ADDR	INTEGER*4	CAMAC address - includes Software Channel, Branch, Crate, and subAddress
GPIB_DATN_FUNC	INTEGER*4	CAMAC Function code to Disable Attention
Q_TRY	INTEGER*4	The number of times to loop for a Q response

5.1.3 GPDREN

Function:

Disables the Remote ENable on the GPIB bus.

Calling Sequence:

```
GPIBSTAT = GPDREN ( )
```

Input/Output :

Inputs (from the COMMON block HDW_GPIB) :

GPIB_DREN_ADDR	INTEGER*4	CAMAC address - includes Software Channel, Branch, Crate, and subAddress
GPIB_DREN_FUNC	INTEGER*4	CAMAC Function code to Disable Remote ENable
Q_TRY	INTEGER*4	The number of times to loop for a Q response

5.1.4 GPDUAI

Function:

Disables the Universal and Addressed Command Inhibit.

Calling Sequence:

```
GPIBSTAT = GPDUAI ( )
```

Input/Output :

Inputs (from the COMMON block HDW_GPIB) :

GPIB_DUAI_ADDR	INTEGER*4	CAMAC address - includes Software Channel, Branch, Crate, and subAddress
GPIB_DUAI_FUNC	INTEGER*4	CAMAC Function code to Disable Universal and Addressed Command Inhibit
Q_TRY	INTEGER*4	The number of times to loop for a Q response

5.1.5 GPIFC

Function:

Generates an InterFace Clear on the GPIB bus.

Calling Sequence:

```
GPIBSTAT = GPIFC ( )
```

Input/Output :

Inputs (from the COMMON block HDW_GPIB) :

GPIB_IFC_ADDR	INTEGER*4	CAMAC address - includes Software Channel, Branch, Crate, and subAddress
GPIB_IFC_FUNC	INTEGER*4	CAMAC Function code to execute an InterFace Clear (IFC)
Q_TRY	INTEGER*4	The number of times to loop for a Q response

5.1.6 GPREAD

Function:

READs one data word (zero-extended byte) from the GPIB bus.

Calling Sequence:

GPIBSTAT = GPREAD (READ_WORD)

Input/Output:

Inputs (from the COMMON block HDW_GPIB) :

GPIB_READ_IN_ADDR	INTEGER*4	CAMAC address - includes Software Channel, Branch, Crate, and subAddress
GPIB_READ_IN_FUNC	INTEGER*4	CAMAC Function code to Read the Input Register of the controller
Q_TRY	INTEGER*4	The number of times to loop for a Q response

Outputs:

READ_WORD	INTEGER*2	A zero-extended byte of data from GPIB

5.1.7 GPREADCH

Function:

READs one ASCII character from the GPIB bus.

Calling Sequence:

GPIBSTAT = GPREADCH (ASCII_CHARACTER)

Input/Output:

Inputs (from the COMMON block HDW_GPIB):

GPIB_READ_IN_ADDR	INTEGER*4	CAMAC address - includes Software Channel, Branch, Crate, and subAddress
GPIB_READ_IN_FUNC	INTEGER*4	CAMAC Function code to Read the Input Register of the controller
Q_TRY	INTEGER*4	The number of times to loop for a Q response

Outputs:

ASCII_CHARACTER	CHARACTER*1	An ASCII character from GPIB

5.1.8 GPREN

Function:

Sets GPIB devices in REMOTE, where operating parameters may be set by sofware, rather than manually.

Calling Sequence:

GPIBSTAT = GPREN ( )

Input/Output:

Inputs (from the COMMON block HDW_GPIB):

GPIB_REN_ADDR	INTEGER*4	CAMAC address - includes Software Channel, Branch, Crate, and subAddress
GPIB_REN_FUNC	INTEGER*4	CAMAC Function code to Enable Remote
Q_TRY	INTEGER*4	The number of times to loop for a Q response

5.1.9 GPSTATUS

Function:

Reads the status register in the GPIB Interface Module.

Calling Sequence:

GPIBSTAT = GPSTATUS (STATUS_WORD)

Input/Output:

Inputs (from the COMMON block HDW_GPIB) :

GPIB_READ_STAT_ADDR	INTEGER*4	CAMAC address - includes Software Channel, Branch, Crate, and subAddress
GPIB_READ_STAT_FUNC	INTEGER*4	CAMAC Function code to Read the Status Register of the controller
Q_TRY	INTEGER*4	The number of times to loop for a Q response

Outputs:

STATUS_WORD	INTEGER*2	The status byte from the controller

5.1.10 GPUAI

Function:

Enables the Universal and Addressed Command Inhibit.

Calling Sequence:

```
GPIBSTAT = GPUAI ( )
```

Input/Output:

Inputs (from the COMMON block HDW_GPIB):

GPIB_UAI_ADDR	INTEGER*4	CAMAC address - includes Software Channel, Branch, Crate, and subAddress
GPIB_UAI_FUNC	INTEGER*4	CAMAC Function code to Enable the Universal and Addressed command Inhibit
Q_TRY	INTEGER*4	The number of times to loop for a Q response

5.1.11 GPWRITE

Function:

WRITEs a byte to the GPIB bus.

Calling Sequence:

GPIBSTAT = GPWRITE (WRITE_WORD)

Input/Output:

Inputs (from the COMMON block HDW_GPIB):

GPIB_WRITE_OUT_ADDR	INTEGER*4	CAMAC address - includes Software Channel, Branch, Crate, and subAddress
GPIB_WRITE_OUT_FUNC	INTEGER*4	CAMAC Function code to write to the Output Register of the controller
Q_TRY	INTEGER*4	The number of times to loop for a Q response

Input (from argument list):

WRITE_WORD	INTEGER*2	A word of data to be written to GPIB

5.1.12 GPWRITECH

Function:

WRITEs a character to the GPIB bus.

Calling Sequence:

GPIBSTAT = GPWRITECH (ASCII_CHARACTER)

Input/Output :

Inputs (from the COMMON block HDW_GPIB):

GPIB_WRITE_OUT_ADDR	INTEGER*4	CAMAC address - includes Software Channel, Branch, Crate, and subAddress
GPIB_WRITE_OUT_FUNC	INTEGER*4	CAMAC Function code to write to Output Register
Q_TRY	INTEGER*4	The number of times to loop for a Q response

Inputs (from argument list):

ASCII_CHARACTER	CHARACTER*1	An ASCII character to write to GPIB

5.2 Universal and Addressed Commands

These routines perform the GPIB Universal and Addressed Commands. Not all Universal and Addressed Commands have been implemented in this package, since there was never a need for them at our installation.

5.2.1 GPGET

Function:

This routine performs a Group Execute Trigger on the GPIB bus.

Calling Sequence:

GPIBSTAT = GPGET ( )

Input/Output:

Inputs (from the COMMON block HDW_GPIB):

GPIB_TALK_ADDR	INTEGER*2	Talk address of the controller

Outputs:

none

5.2.2 GPLLO

Function:

This routine performs the GPIB Universal Command 'Local LockOut (LLO) on the GPIB (IEEE Std 488) bus.

Calling Sequence:

GPIBSTAT = GPLLO ( )

Input/Output:

none

5.2.3 GPSPD

Function:

This routine performs a Serial Poll Disable on the GPIB bus.

Calling Sequence:

GPIBSTAT = GPSPD ( )

Input/Output:

none

5.2.4 GPSPE

Function:

This routine performs a Serial Poll Enable on the GPIB bus.

Calling Sequence:

GPIBSTAT = GPSPE ( )

Input/Output:

none

5.3 Miscellaneous Routines

5.3.1 GPSPSTAT

Function:

This routine makes a GPIB device a TALKER and reads its Serial Poll Status Word (actually a byte).

Calling Sequence:

```
GPIB_STATUS = GPSPSTAT ( GPIB_DEVICE_TALK_ADDR, POLL_STATUS )
```

Input/Output:

Inputs (from the argument list):

GPIB_DEVICE_TALK_ADDR	INTEGER*2	TALK address of the device

Outputs (to argument list):

POLL_STATUS	INTEGER*2	The serial poll status word

Programmer's Comments :

You must enable Serial Polling before calling this routine.

5.3.2 GPSRQ

Function:

This routine reads the status register in the GPIB controller and returns to its caller when the SRQ bit is set.

Calling Sequence:

```
STATUS = GPSRQ ( )
```

Input/Output:

none

5.3.3 GPPOLL

Function:

This routine performs a serial poll on the specified devices and returns an array of their serial poll status words.

Calling Sequence:

```
GPIB_STATUS = GPPOLL ( POLL_TALK_ADDR, POLL_COUNT, POLL_STATUS )
```

Input/Output:

Inputs (from the COMMON block HDW_GPIB):

GPIB_LISTEN_ADDR	INTEGER*2	LISTEN address of interface module
GPIB_READ_STAT_MLA	INTEGER*2	The bit in the interface module which is set when the module is a listener
Q_TRY	INTEGER*4	The number of times for CAMAC to attempt an operation before giving up.

Inputs (from the argument list):

POLL_TALK_ADDR(POLL_COUNT)	INTEGER*2	An array containing the TALK addresses of devices to be polled
POLL_COUNT	INTEGER*4	The number of devices to poll

Outputs (to argument list):

POLL_STATUS(POLL_COUNT)	INTEGER*2	An array of serial poll status words

5.3.4 GPRDAT

Function:

Reads a word array containing ASCII data from the GPIB bus.

Calling Sequence:

```
GPIB_STATUS = GPRDAT ( GPIB_DEVICE_ADDR, DEVICE_COUNT,
2                      DATA_ARRAY, BYTES_TRANSMITTED )
```

Input/Output:

Input:

GPIB_DEVICE_ADDR	INTEGER*2(DEVICE_COUNT)	Array of GPIB addresses
DEVICE_COUNT	INTEGER*4	The sum of the number of TALKERS and LISTENERS on the GPIB bus.

Output:

DATA_ARRAY	INTEGER*2(BYTES_TRANSMITTED+2)	Data in ASCII format
BYTES_TRANSMITTED	INTEGER*4	The number of bytes actually transmitted. (not counting the CR and LF)

5.3.5 GPRDATSTR

Function:

Reads a character string containing data from the GPIB bus.

Calling Sequence:

```
GPIB_STATUS = GPRDATSTR ( GPIB_DEVICE_ADDR, DEVICE_COUNT,
2                         DATA_STRING, BYTES_TRANSMITTED )
```

Input/Output:

Input:

GPIB_DEVICE_ADDR	INTEGER*2(DEVICE_COUNT)	Array of GPIB addresses
DEVICE_COUNT	INTEGER*4	The sum of the number of TALKERS and LISTENERS on the GPIB bus.

Output:

DATA_STRING	CHARACTER*(*)	data from GPIB
BYTES_TRANSMITTED	INTEGER*4	The number of bytes actually transmitted. (not counting the CR and LF)

5.3.6 GPTRIG

Function:

This routine performs a Group Execute Trigger to selected devices on the GPIB bus.

Calling Sequence:

```
GPIBSTAT = GPTRIG ( DEVICES_TO_TRIGGER, NUMBER_OF_DEVICES )
```

Input/Output:

Inputs (from the COMMON block HDW_GPIB):

GPIB_TALK_ADDR	INTEGER*2	Talk address of the controller

Inputs (from the argument list):

DEVICES_TO_TRIGGER(NUMBER_OF_DEVICES)	INTEGER*2	Listen addresses of the devices to trigger
NUMBER_OF_DEVICES	INTEGER*2	The number of devices to trigger

Outputs:

none

5.3.7 GPULISTEN

Function:

Tells a device to become a LISTENER.

Calling Sequence:

```
GPIB_STATUS = GPULISTEN ( GPIB_DEVICE_LISTEN_ADDR )
```

Input/Output:

Input (from the COMMON block HDW_GPIB):

GPIB_READ_STAT_ATN	INTEGER*2	Bit in Status Register

Input (from the argument list):

GPIB_DEVICE_LISTEN_ADDR	INTEGER*2	GPIB listen address

5.3.8 GPUNL

Function:

This routine UNLISTENs all devices on the GPIB bus.

Calling Sequence:

GPIBSTAT = GPUNL ( )

Input/Output:

Input (from the COMMON block HDW_GPIB):

GPIB_READ_STAT_ATN	INTEGER*2	Bit in Status Register

5.3.9 GPUNT

Function:

This routine UNTALKs the current talker on the GPIB bus.

Calling Sequence:

GPIBSTAT = GPUNT ( )

Input/Output:

Input (from the COMMON block HDW_GPIB):

GPIB_READ_STAT_ATN	INTEGER*2	Bit in Status Register

5.3.10 GPUTALK

Function:

Tells a device to become a TALKER.

Calling Sequence:

GPIB_STATUS = GPUTALK (GPIB_DEVICE_TALK_ADDR)

Input/Output:

Input (from the COMMON block HDW_GPIB):

GPIB_READ_STAT_ATN	INTEGER*2	Bit in Status Register

Input:

GPIB_DEVICE_TALK_ADDR	INTEGER*2	GPIB talk address

5.3.11 GPXMIT

Function:

Transmits a word array containing ASCII data to the GPIB bus.

Calling Sequence:

```
GPIB_STATUS = GPXMIT ( DATA_ARRAY, BYTE_COUNT,
1                          GPIB_DEVICE_ADDR, DEVICE_COUNT,
2                          BYTES_TRANSMITTED )
```

Input/Output:

Inputs (from the COMMON block HDW_GPIB):

GPIB_WRITE_OUT_ADDR	INTEGER*4	CAMAC address - includes Software Channel, Branch, Crate, and subAddress
GPIB_WRITE_OUT_FUNC	INTEGER*4	CAMAC Function code to write to the Output Register

Inputs (from the argument list):

DATA_ARRAY	INTEGER*2(BYTE_COUNT)	data in ASCII format
BYTE_COUNT	INTEGER*4	Number of bytes to send
GPIB_DEVICE_ADDR	INTEGER*2(DEVICE_COUNT)	Array of GPIB addresses
DEVICE_COUNT	INTEGER*4	The sum of the number of TALKERS and LISTENERS on the GPIB bus.

Outputs (to the argument list):

BYTES_TRANSMITTED	INTEGER*4	The number of bytes actually transmitted

5.3.12 GPXMITSTR

Function:

Transmits an ASCII string to the GPIB bus.

Calling Sequence:

```
GPIB_STATUS = GPXMITSTR ( DATA_STRING, BYTE_COUNT,
1                         GPIB_DEVICE_ADDR, DEVICE_COUNT,
2                         BYTES_TRANSMITTED )
```

Input/Output:

Inputs (from the COMMON block HDW_GPIB):

GPIB_WRITE_OUT_ADDR	INTEGER*4	CAMAC address - includes Software Channel, Branch Crate, and subAddress
GPIB_WRITE_OUT_FUNC	INTEGER*4	CAMAC Function code to write to the Output Register

Inputs (from the argument list):

DATA_STRING	CHARACTER*(*)	ASCII data
BYTE_COUNT	INTEGER*4	Number of bytes to send
GPIB_DEVICE_ADDR	INTEGER*2(DEVICE_COUNT)	Array of GPIB addresses
DEVICE_COUNT	INTEGER*4	The sum of the number of TALKERS and LISTENERS on the GPIB bus.

Outputs (to the argument list):

BYTES_TRANSMITTED	INTEGER*4	The number of bytes actually transmitted