Example 1

USE THE INTERACTIVE GPIB PROGRAM “IBIC”

This example assumes the use of an IBM PC or compatible equipped with a National Instruments GPIB interface card. The GPIB driver is left in default state so that the device name “dev4” corresponds to the GPIB address 4, the oscilloscope address. All `text` is entered by the user.

```
IBIC<cr>
program announces itself
: ibfind<CR>
enter board/device name:  dev4<CR>
dev4:  ibwrt<CR>
enter string:  “tdiv?”<CR>
[0100] ( cmpl )
count: 5
dev4:  ibrd<CR>
enter byte count:  10<CR>
[0100] ( cmpl )
count: 10
54 44 49 56 20 35 30 45           T D I V   5 0 E
2D 39                             – 9
dev4:  ibwrt<CR>
enter string:  “c1:cpl?”<CR>
[0100] ( cmpl )
count: 7
dev4:  ibrd<CR>
enter byte count:  20<CR>
[2100] ( end cmpl )
count: 11
43 31 3A 43 50 4C 20 44           C 1 : C P L D
35 30 0A                          5 0 z
dev4:  q<CR> to quit the program.
```

Example 2

USE THE GPIB PROGRAM FOR IBM PC (HIGH-LEVEL FUNCTION CALLS)

The following BASICA program allows full interactive control of the oscilloscope using an IBM PC as GPIB controller. As in Example 1, it is assumed that the controller is equipped with a National Instruments GPIB interface card. All commands can be used following this example simply by entering the text string of the command. For example, `“C1:VDIV 50 MV”`, without the quotation marks. The program automatically displays the information sent back by the oscilloscope in response to queries.

In addition, a few utilities have been provided for convenience. The commands ST and RC enable waveform data to be stored on, or retrieved from, a disk if the correct drive and file names are provided. The command LC returns the oscilloscope to local mode. Responses sent back by the oscilloscope are interpreted as character strings and are thus limited to a maximum of 255 characters.

```
1-99<DECL.BAS>

100CLS

110PRINT "Control of the 9300 via GPIB and IBM PC"

115PRINT ""

120PRINT "Options :EX to exitLC local mode"

125PRINT "ST store dataRC recall data"

130PRINT ""

140LINE INPUT "GPIB-address of oscilloscope (1...16)? :",ADDR$

145DEV$ = "DEV" + ADDR$

150CALL IBFIND(DEV$,SCOPE%)

155IF SCOPE% < 0 THEN GOTO 830

160TMO% = 10  ’timeout = 300 msec (rather than default 10 sec)

165CALL IBTMO(SCOPE%,TMO%)

170’

200LOOP% = 1

205WHILE LOOP%

210LINE INPUT "Enter command (EX --> Exit) : ",CMD$

220IF CMD$ = "ex" OR CMD$ = "EX" THEN LOOP% = 0 : GOTO 310

230IF CMD$ = "st" OR CMD$ = "ST" THEN GOSUB 600 : GOTO 300

240IF CMD$ = "rc" OR CMD$ = "RC" THEN GOSUB 700 : GOTO 300

250IF CMD$ = "lc" OR CMD$ = "LC" THEN GOSUB 400 : GOTO 300
```

```
260 IF CMD$ = "" THEN GOTO 300
270 CALL IBWRT(SCOPE%,CMD$)
275 IF IBSTA% < 0 THEN GOTO 840
280 GOSUB 500
300 WEND
310 GOSUB 400
320 END
400 ’
405 ’SUBROUTINE LOCAL_MODE
410 ’
420 CALL IBLOC(SCOPE%)
425 PRINT ""
430 RETURN
500 ’
505 ’SUBROUTINE GET_DATA
510 ’If there are no data to read, simply wait until timeout occurs
515 ’
520 CALL IBRD(SCOPE%,RD$)
525 I = IBCNT% ’IBCNT% is the number of characters read
530 FOR J = 1 TO I
535 PRINT MID$(RD$,J,1);
540 NEXT J
545 PRINT ""
550 RETURN
600 ’
605 ’SUBROUTINE STORE_DATA
610 ’
615 RD1$=SPACE$(3)
620 LINE INPUT "Specify trace (TA...TD,M1...M4,C1...C4): ",TRACE$
625 LINE INPUT "Enter filename : ",FILE$
630 CMD$="WFSU NP,0,SP,0,FP,0,SN,0; CHDR SHORT"
```

```
640CALL IBWRT(SCOPE%,CMD$)
645CMD$=TRACE$+":WF?"
650CALL IBWRT(SCOPE%,CMD$)
660CALL IBRD(SCOPE%,RD1$)      ’Discard first 3 chars of response
665CALL IBRDF(SCOPE%,FILE$)
670IF IBSTA% < 0 THEN GOTO 840
675PRINT ""
680RETURN
700’
705’SUBROUTINE RECALL_DATA
710’
715LINE INPUT "Specify target memory (M1...M4):",MEM$
720LINE INPUT "Enter filename : ",FILE$
730CMD$=MEM$+":"
735CALL IBWRT(SCOPE%,CMD$)
740CALL IBWRTF(SCOPE%,FILE$)
745IF IBSTA% < 0 THEN GOTO 840
750PRINT ""
755RETURN
800’
810’ERROR HANDLER
820’
830PRINT "IBFIND ERROR"
835END
840PRINT "GPIB ERROR -- IBERR: ";IBERR%;"IBSTA: ";HEX$(IBSTA%)
END
```

NOTE:

- ***It is assumed that the National Instruments GPIB driver GPIB.COM is in its default state. This means that the interface board can be referred to by its symbolic name 'GPIB0' and that devices on the GPIB with addresses 1 to 16 can be called by the symbolic name 'DEV1' to 'DEV16'.***
- ***Lines 1–99 are a copy of the file DECL.BAS supplied by National Instruments. The first six lines are required for the initialization of the GPIB handler. DECL.BAS requires access to the file BIB.M during the GPIB initialization. BIB.M is one of the files supplied by National Instruments, and must exist in the directory currently in use.***
- ***The first two lines of DECL.BAS each contain a string "XXXXX" which must be replaced by the number of bytes which determine the maximum workspace for BASICA (computed by subtracting the size of BIB.M from the currently available space in BASICA). For example, if the size of BIB.M is 1200 bytes and when BASICA is loaded it reports "60200 bytes free", "XXXXX" would be replaced by the value 59000 or less.***
- ***The default timeout of 10 seconds is modified to 300 ms during the execution of this program. However, the default value of the GPIB handler remains unchanged. Whenever a remote command is entered by the user, the program sends it to the instrument with the function call IBWRT. Afterwards, it always executes an IBRD call, regardless of whether or not a response is expected. If a response is received it is immediately displayed. If there is no response, the program waits until time-out and then asks for the next command.***

Example 3

USE GPIB PROGRAM FOR IBM PC (LOW-LEVEL FUNCTION CALLS)

This example has the same function as Example 2, but is written with low-level function calls. The program assumes that the controller (board) and oscilloscope (device) are at addresses 0 and 4, respectively, and the decimal addresses are:

	Listener Address	Talker Address
CONTROLLER	32(ASCII <space>)	64 (ASCII @)
DEVICE	32+4=36 (ASCII $)	64+4=68 (ASCII D)

```
1–99<DECL.BAS>
100CLS
110PRINT "Control of the 9300 (address 4) via GPIB and IBM PC"
115PRINT "": PRINT "Options :  EX to exit        LC local mode"
120PRINT "           ST store data     RC recall data": PRINT""
125LOOP=1
130CMD1$ = "?_@$" 'Unlisten, Untalk, Board talker, Device listener
135CMD2$ = "?_ D" 'Unlisten, Untalk, Board listener, Device talker
140BDNAME$= "GPIB0": CALL IBFIND(BDNAME$,BRD0%)
145IF BRD0% < 0 THEN GOTO 420
150CALL IBSIC(BRD0%): IF IBSTA% < 0 THEN GOTO 425
155WHILE LOOP
160LINE INPUT "Enter command (EX --> Exit) : ",CMD$
165V% = 1: CALL IBSRE(BRD0%,V%)
170IF CMD$ = "ex" OR CMD$ = "EX" THEN LOOP = FALSE: GOTO 205
175IF CMD$ = "st" OR CMD$ = "ST" THEN GOSUB 285: GOTO 200
180IF CMD$ = "rc" OR CMD$ = "RC" THEN GOSUB 365: GOTO 200
185IF CMD$ = "lc" OR CMD$ = "LC" THEN  GOSUB 240: GOTO 200
190IF CMD$ = "" THEN GOTO 200
195CALL IBCMD(BRD0%,CMD1$): CALL IBWRT(BRD0%,CMD$): GOSUB 270
200WEND
```

```
205CALL IBSIC(BRD0%): V%=0: CALL IBSRE(BRD0%,V%)
210CALL IBSIC(BRD0%)
215END
220’
230’LOCAL MODE
235’
240V% = 0: CALL IBSRE(BRD0%,V%): PRINT ""
245RETURN
250’
260’SUBROUTINE GET_DATA
265’
270CALL IBCMD(BRD0%,CMD2$): CALL IBRD(BRD0%,RD$): I=IBCNT%
275FOR J=1 TO I: PRINT MID$(RD$,J,1);: NEXT J: PRINT ""
280RETURN
285’
290’SUBROUTINE STORE_DATA
295’
300RD1$=SPACE$(3)
305LINE INPUT "Specify trace (TA...TD,M1...M4,C1...C4):  ",TRACE$
310LINE INPUT "Enter filename : ",FILE$
315CALL IBCMD(BRD0%,CMD1$)
320CMD$="WFSU NP,0,SP,0,FP,0,SN,0;CHDR SHORT"
321CALL IBWRT(BRD0%,CMD$)
325CMD$=TRACE$+":WF?": CALL IBWRT(BRD0%,CMD$)
330CALL IBCMD(BRD0%,CMD2$): CALL IBRD(BRD0%,RD1$)
335CALL IBRDF(BRD0%,FILE$)
340IF IBSTA% < 0 THEN GOTO 430
345PRINT ""
350RETURN
355’
360’SUBROUTINE RECALL_DATA
```

```
365’
370LINE INPUT "Specify target memory (M1...M4): ",MEM$
375LINE INPUT "Enter filename : ",FILE$
380CALL IBCMD(BRD0%,CMD1$)
385CMD$=MEM$+":": CALL IBWRT(BRD0%,CMD$)
390CALL IBWRTF(BRD0%,FILE$)
395IF IBSTA% < 0 THEN GOTO 430
400PRINT ""
405RETURN
410’
415’ERROR HANDLER
420’
425PRINT "IBFIND ERROR": STOP
430PRINT "GPIB ERROR -- IBERR : ";IBERR%;"IBSTA : ";HEX$(IBSTA%)
435STOP
440END
```

NOTE: The Template also describes an array named DUAL This is simply a way to allow you to use the `INSPECT?` query to examine the two data arrays together.

.