DL/CSE/TM34

technical memorandum Daresbury Laboratory

DL/CSE/TM34

THE ADDITION OF A REMOTE CONTROL FACILITY TO THE NSF DATA ACQUISITION SYSTEM, FOR USERS INSTRUMENTATION, USING THE IEEE-488 GENERAL PURPOSE INTERFACE BUS

by

R. BERRY, Daresbury Laboratory.

NOVEMBER, 1984

Science & Engineering Research Council

Daresbury Laboratory

Daresbury, Warrington WA4 4AD

IMPORTANT

The SERC does not accept any responsibility for loss or damage arising from the use of information contained in any of its reports or in any communication about its tests or investigations.

THE ADDITION OF A REMOTE CONTROL FACILITY TO THE NSF DATA ACQUISITION SYSTEM, FOR USERS INSTRUMENTATION, USING THE IEEE-488 GENERAL PURPOSE INTERFACE BUS

by

ROBERT BERRY

SCIENCE AND ENGINEERING RESEARCH COUNCIL
DARESBURY LABORATORY

INTRODUCTION

There is an increasing need for users of the Daresbury Nuclear Structure Facility to be able to remotely control their own instrumentation in conjunction with the specific requirements of certain experiments. The recent installation of a Mobile Graphics System in Area 1 has made remote control of this kind feasible in the experimental areas.

This document describes the manner in which a user could set up such a system, using part of the existing Data Acquisition System to carry the remote messages. The only special requirement is that the instrument to be controlled is fitted with the IEEE-488 bus option.

SYSTEM OVERVIEW

The system is based on the General Purpose Interface Bus, (GPIB), which is designed to IEEE-488 standard[(1)]. A Camac to GPIB interface[(3)] is installed in any spare station of an existing Camac crate that is in communication, via the serial highway, with one of the Data Acquisition Computers. Figure 1 shows the system layout. The most appropriate computer is the C machine, as this is the device that the Mobile Graphics Station is connected to in Area 1, however, any processor could be used depending upon the individual users needs. In the near future all experimental areas will be equipped with a serial highway outlet, such that the Mobile Graphics Station can be moved to any area and thus provide this remote control facility. Once the position of the instrumentation control crate is decided, any terminal, including the control crate terminal, connected to the same processor may be chosen to send and/or receive remote messages.

GENERAL PURPOSE INTERFACE BUS

It is not the intention to give the reader precise details of the operation of the GPIB here, however, an attempt at giving a general appreciation of the bus operating procedure is made. It will be necessary to consult the references specified at the end of this document, if a precise understanding is to be gained. This is particularly the case when studying the section entitled "Software Requirements', detailing a specific instrument's needs.

The GPIB was initially designed to be an electronics instrument interface bus. It is referred to by many different titles, two of which are detailed below:

1. Hewlett Packard Interface Bus (HPIB). Hewlett Packard (the originators of the bus), make extensive use of it in their computers and instrumentation. The bus interface handshaking protocol is patented by Hewlett Packard, and a licence is required to manufacture devices using it.
2. IEEE-488 Bus. The 'Standard'[1] to which the bus is designed.

The GPIB transfers data and command information between the components of an instrumentation system over sixteen signal lines. Eight of the lines are reserved for data and other message transfers, which are coordinated by three handshake lines. The remaining five lines are used to control bus activity. Figure 2 shows the GPIB concept[2]. The devices connected to the bus may be Talkers (sources), Listeners (acceptors), or controllers. The controller specifies the role of each of the other devices. It does this by setting various control lines and sending the appropriate Talk or Listen addresses to the devices. There can be more than one listening device active simultaneously, but only one talking device at a time. Information is transferred in a bit-parallel, byte-serial manner using an asynchronous, sequential handshake procedure. No step in the sequence can be initiated until the previous step is complete. Information transfer can proceed as fast as devices can respond, but no faster than the slowest device presently addressed as active. This permits several devices to receive the same message byte concurrently.

SETTING UP THE SYSTEM

The setting up procedure is best explained by practical example.

The set up to be described is one used in an experiment by the Oxford Low Temperature Nuclear Orientation Group. Only one instrument is used in this experiment, however up to fifteen instruments may be included on one contiguous bus.

The basic requirement of the experiment was to perform a frequency search, synchronised with data accumulation. At the start of a search, an automatic set up of some initial signal generator parameters was also required. Because of problems associated with signal attenuation, it was essential that the signal generator be located as close as possible to the rest of the experimental equipment, whilst retaining the ability to control the device from the NSF Main Control Room (100 metres away).

System Components:

Daresbury Mobile Graphics Station	Located in Area 1, and in communication with the GEC 'C' processor.
Kinetic Systems GPIB Interface Type 3388[3]	Inserted in a spare station in the Mobile Graphics Station Camac Crate
Hewlett Packard Signal Generator Type 8656A[4]	Connected to GPIB interface via standard bus cable

Software Requirements:

The programme, written in FORTRAN 3, to send command strings to the signal generator is shown in listing 1. This programme uses the Camac control subroutines written by D. Brightly.

When the programme is executed, the command string sent to the signa generator is stored as the JCL variable HENERGY which is then written intl the energy field in DSM (Data Storage Manager), when spectra are saved. The essential programme steps are as follows:

1. The module containing the Camac to GPIB interface is booked.
2. Clear GPIB interface.
3. The GPIB ATN (attention), and REN (remote enable) functions are enabled.
4. The GPIB Talk address, and the signal generators Listen address are sent to the interface.
5. ATN (attention) is disabled.
6. The command string is sent to the interface character by character. A test is made before each character is sent to ensure the last one has been accepted and cleared by the interface, before the next one arrives.
7. The command string is terminated by a 'linefeed' character upon which the signal generator executes the command string.

The macro used to control the signal generator and data acquisition is shown in listing 2. It contains various comments to help make its functions transparent.

CONCLUSION

The practical application of remotely controlling the signal generator as described earlier, is the first implementation of this facility on the NSF. Its success has shown the usefulness of such an addition to the existing Data Acquisition System.

The procedure will allow the remote control of any instrumentation that has the GPIB option fitted. The monitoring of additional experimental parameters during live runs, will enable 'fine tuning' to be effected in the convenience of the NSF Main Control Room.

ACKNOWLEDGEMENTS

My thanks to Vincent Green, of the Oxford Low Temperature Nuclear Orientation Group, for describing the experiment in which the remote control facility was used, and for providing the two software listings used in the experiment.

REFERENCES

1. IEEE-488 - 1975 Digital Interface for Programmable Instrumentation

2. Tektronix 7854 Oscilloscope Operators Manual

3. Kinetic Systems GPIB Interface type 3388-G1A Instruction Manual

4. Hewlett Packard 8656A Signal Generator Handbook

LISTING 1

```
      PROGRAM HP
C
C      HP8656A SIGNAL GENERATOR CONTROL PROGRAM.
C      USES GPIB INTERFACE IN CAMAC CRATE 1, STATION 18
C      OF BRANCH 3.
C      IF THIS IS ALTERED THE PROGRAM MUST BE RECOMPILED
C      WITH THE NEW POSITION IDENTIFIERS.
C      USE FORTRAN 3
C      LINK LIBRARIES SYS.DCAM3LIB
C                     SYS.SUPV3LIB
C                     SYS.FORT3LIB
C
C      V.R. GREEN 19-JULY-1984
C
C
C
C      THE PROGRAM USES A PROFORMA HP WHICH IS SET UP ON SIGNING ON
C      AS USER NOG ON THE C MACHINE.
C
C      PROFORMA HP COMM STRING() OUT STREAM 2(*/NEW)
C
C      TO RUN PROGRAM
C      .HP(P=HP) COMMANDSTRING
C
C
C      THE COMMANDNAME IS SAVED IN THE ENERGY FIELD
C      IN DSM.
C
      CHARACTER*20 COMM
      INTEGER COMMLN,ERRCOD,BRANCH,CRATE,STAT,QCODE
      INTEGER STATUS
C
C      GET COMMAND STRING
C
      CALL FSARG('COMM',4,COMM,COMMLN,ERRCOD)
      IF(ERRCOD.LT.0)WRITE(2,1)
 1    FORMAT(1X,'ERROR IN CALL TO FSARG')
C
C      SET JCL VARIABLE HENERGY TO BE COMM.
C
      CALL FSETSV('HENERGY',7,COMM(1:COMMLN),COMMLN,ERRCOD)
      IF(ERRCOD.LT.0)WRITE(2,80)
 80   FORMAT(1X,'ERROR IN SETTING JCL VARIABLE HENERGY')
C
C      DEFINE BRANCH,CRATE AND STATION NUMBERS FOR GPIB INTERFACE.
C
      BRANCH=3
      CRATE=1
      STAT=18
C
C      BOOK MODULE
C
      CALL CAMBK(BRANCH,CRATE,STAT,ERRCOD)
      IF(ERRCOD.LT.0)WRITE(2,2)
 2    FORMAT(1X,'ERROR IN CALL TO CAMBK')
C
C      CLEAR INTERFACE
C
      CALL CAMWT(BRANCH,CRATE,STAT,0,25,1,ERRCOD,QCODE)
      IF(ERRCOD.LT.0)WRITE(2,30)
 30   FORMAT(1X,'ERROR IN IFC')
C
C      ENABLE UNIVERSAL AND ADDRESS COMMAND INHIBIT.
C
      CALL CAMWT(BRANCH,CRATE,STAT,4,26,1,ERRCOD,QCODE)
      IF(ERRCOD.LT.0)WRITE(2,40)
 40   FORMAT(1X,'ERROR IN UNIVERSAL AND ADDRESS COMMAND INHIBIT')
C
C      ENABLE ATTENTION.
C
      CALL CAMWT(BRANCH,CRATE,STAT,0,26,1,ERRCOD,QCODE)
      IF(ERRCOD.LT.0)WRITE(2,3)
 3    FORMAT(1X,'ERROR IN ENABLING OF ATTENTION')
C
C      ENABLE REMOTE ENABLE
C
      CALL CAMWT(BRANCH,CRATE,STAT,1,26,1,ERRCOD,QCODE)
      IF(ERRCOD.LT.0)WRITE(2,4)
 4    FORMAT(1X,'ERROR IN ENABLING OF REMOTE ENABLE')
C
C      SEND HP LISTEN ADDRESS (DECIMAL 39)
C
      CALL CAMWT(BRANCH,CRATE,STAT,0,16,39,ERRCOD,QCODE)
      IF(ERRCOD.LT.0)WRITE(2,5)
 5    FORMAT(1X,'ERROR IN SENDING OF HP LISTEN ADDRESS')
C
C      SEND GPIB TALK ADDRESS (DECIMAL 64)
C
      CALL CAMWT(BRANCH,CRATE,STAT,0,16,64,ERRCOD,QCODE)
      IF(ERRCOD.LT.0)WRITE(2,6)
 6    FORMAT(1X,'ERROR IN SENDING GPIB TALK ADDRESS')
C
C      DISABLE ATTENTION.
C
 200  CONTINUE
      CALL CAMWT(BRANCH,CRATE,STAT,0,24,1,ERRCOD,QCODE)
      IF(ERRCOD.LT.0)WRITE(2,7)
 7    FORMAT(1X,'ERROR IN DISABLING ATTENTION')
C
C      SEND DATA MESSAGE.
C
      DO 100 I=1,COMMLN
      ICOMM=ICHAR(COMM(I:I))
      CALL CAMWT(BRANCH,CRATE,STAT,0,16,ICOMM,ERRCOD,QCODE)
      IF(ERRCOD.LT.0)WRITE(2,8)
 8    FORMAT(1X,'ERROR IN SENDING COMMAND')
C
C      TEST IF DATA SENT
C
 400  CONTINUE
      CALL CAMRD(BRANCH,CRATE,STAT,12,1,STATUS,ERRCOD,QCODE)
      IF(MOD(STATUS,16).LT.8)GOTO 400
 100  CONTINUE
C
C      SEND LINE FEED.
C
      CALL CAMWT(BRANCH,CRATE,STAT,0,16,10,ERRCOD,QCODE)
      IF(ERRCOD.LT.0)WRITE(2,9)
 9    FORMAT(1X,'ERROR IN SENDING LINE FEED')
      STOP
```

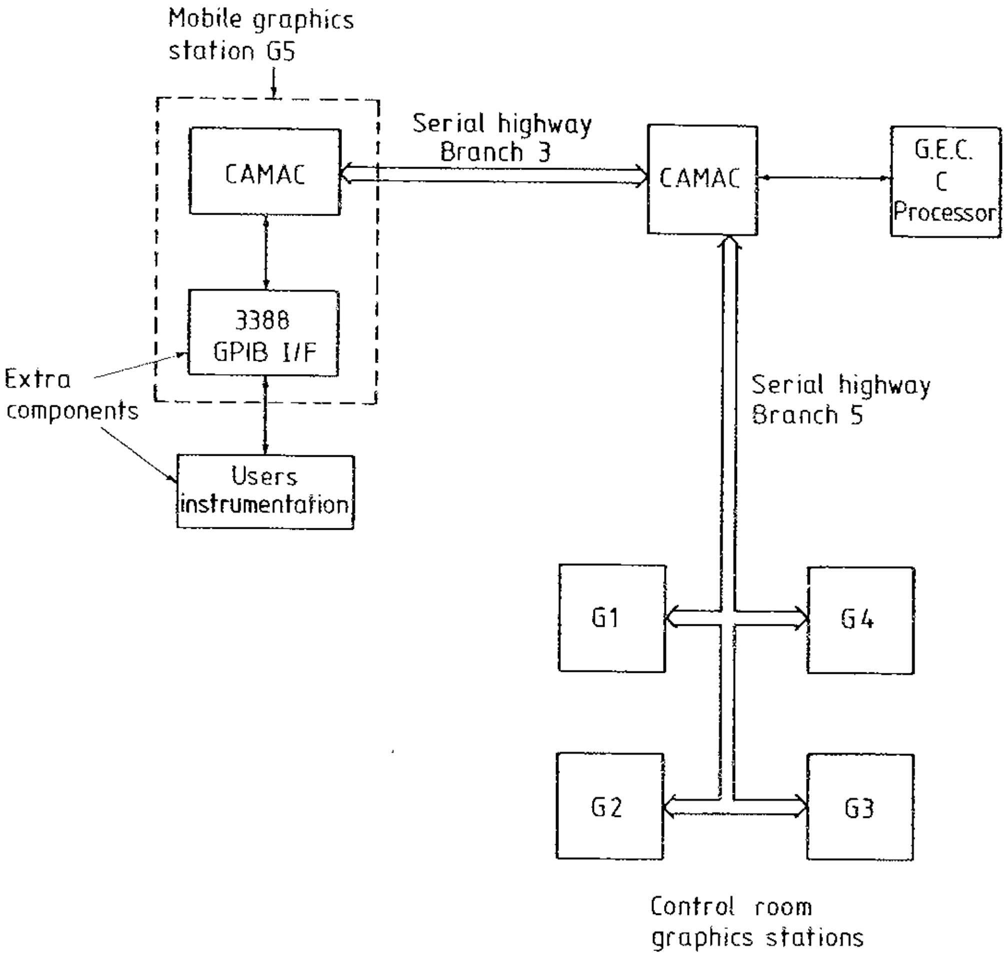
LISTING 2

```
ARGUMENT OPT STRING ()

IF '<OPT>' EQ 'AGAIN' THEN GOTO LABAGAIN
IF '<OPT>' EQ 'HALT' THEN GOTO LABHALT
IF '<OPT>' EQ 'GO' THEN GOTO LABGO
SAY *** OPTIONS ARE GO, AGAIN, HALT !!!
RETURN

// THIS BIT STARTS THE LOOP

LABGO:

//SET UP SOME INITIAL GENERATOR OPTIONS

SET AP=[*/OSCILLATOR AMPLITUDE IN MILLI-VOLTS ]
.HP(P=HP) AP[AP]MV
SET FM=[*/FREQUENCY MODULATION IN KHZ ]
.HP(P=HP) S3FM[FM]KZ
SET FMONOFF='[*/MODULTAION ON AND OFF AT EACH FREQUENCY (Y/N) ]'

//SET UP INITIAL FREQUENCY AND INCREMENT FREQUENCY.

SET FS=[*/FREQUENCY STEP IN MHZ ]
SET FI=[*/INITIAL FREQUENCY IN MHZ ]
.HP(P=HP) FR[FI]MZ

//SET UP TIMING ETC.

SET TIME=[*/ACQUISITION TIME FOR CYCLE IN S ]
SET TI=[*/WAIT TIME BETWEEN FREQUENCY RESET AND ACQUIRE IN S ]
SET BEAM='[*/FRIDGE STATUS ]'
SET RUN='[*/MASS NUMBER ]'

//WAIT TI SECONDS.

SAY WAITING [TI] SECONDS BEFORE STARTING ACQUISITION.
UNSET CLEARPROTECT
HALT
CLEAR SINGLES ALL
GO
EVENT 1 '.RF AGAIN'  [TIME]
COPY ZI+ [EXPTFILE] OUT SINK
STARTING .RF AT [DATETIME] WITH PERIOD [TIME] SECONDS.
+
RETURN

//THIS BIT INCREMENTS FREQUENCY BY PRESET VALUE
//AND PUTS FM OFF/ON.

LABAGAIN:
HALT
SAY SAVING SPECTRA TO DSM
RUN DISCSAVE SINGLES ALL RUN '[RUN]' BEAM '[BEAM]'-
 ENERGY '[HENERGY]' LIST [EXPTFILE]
SAY SPECTRA SAVED
CLEAR SINGLES ALL
IF '[FMONOFF]' EQ 'N' THEN GOTO LABA

//PUT FM OFF

SAY PUTTING FM OFF
.HP(P=HP)  FMS4

//WAIT TI

SAY WAITING [TI] SECONDS BEFORE RESTARTING ACQUISITION.
WAIT [TI]
GO
SAY ACQUIRING DATA FOR [TIME] SECONDS.
WAIT [TIME]
HALT
SAY SAVING SPECTRA TO DSM
RUN DISCSAVE SINGLES ALL RUN '[RUN]' BEAM '[BEAM]'-
 ENERGY '[HENERGY]' LIST [EXPTFILE]
SAY SPECTRA SAVED
CLEAR SINGLES ALL

//PUT FM ON AGAIN

SAY PUTTING FM ON.
.HP(P=HP) S3FM[FM]KZ
LABA:

//INCREMENT FREQUENCY.

SAY INCREMENTING FREQUENCY BY [FS]MHZ.
SET FI=[FI]+[FS]
SAY NEW FREQUENCY IS [FI]MHZ.
.HP(P=HP) FR[FI]MZ
SAY WAITING [TI] SECONDS BEFORE RESTARTING ACQUISITION.
WAIT [TI]
GO
EVENT 1 '.RF AGAIN'  [TIME]
RETURN

LABHALT:
SAY HALTING .RF
EVENT 1
HALT
COPY ZI+ [EXPTFILE] OUT SINK
STOPINGg .RF AT [DATETIME]
+
RETURN
```

FIGURE CAPTIONS

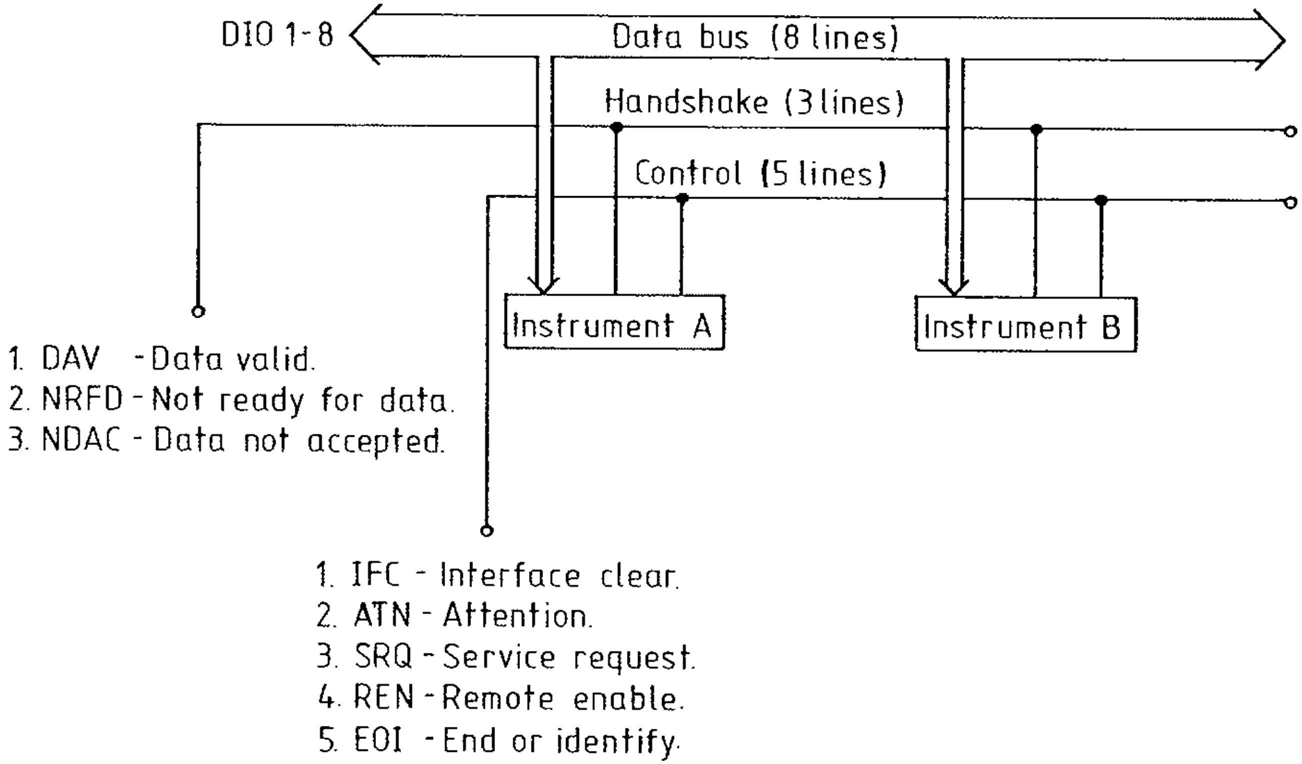
Fig.1 System Layout

Fig.2 GPIB Concept



SYSTEM LAYOUT

Fig.1



GPIB CONCEPT

Fig. 2