

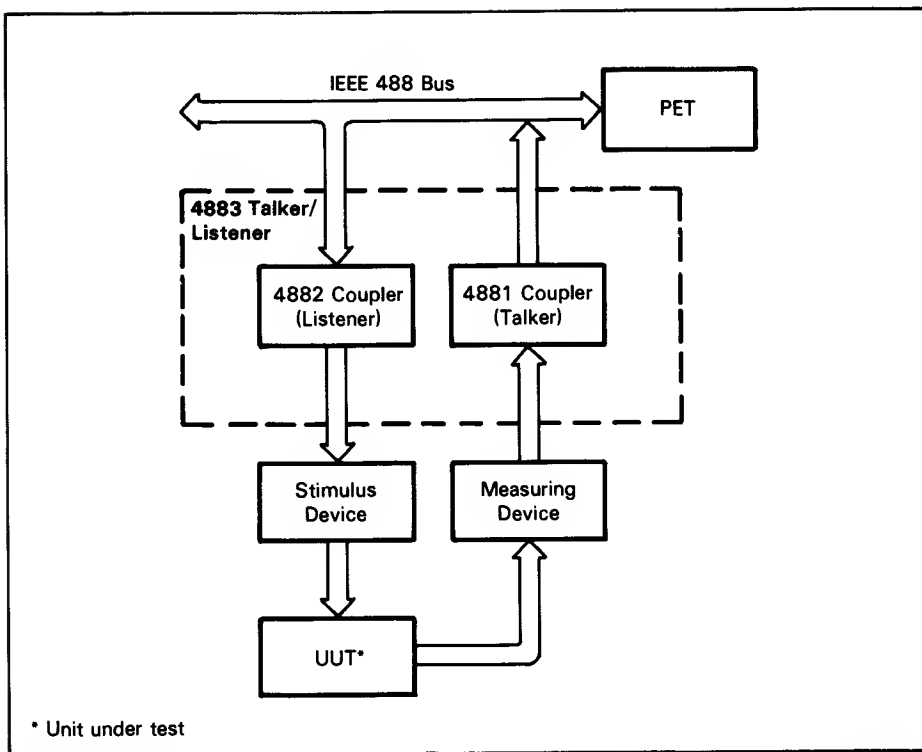
488 BUS INTERFACE COUPLER, ICS MODEL 4880

Quite often digital electronic instruments, although TTL compatible, are designed for a dedicated purpose and do not communicate with other devices on an IEEE 488 Bus. Also, many of the older digital instruments are not IEEE 488 Bus compatible; these instruments typically have multiple digits of input/output where each digit requires four bits (lines). For example, a minimum of 32 lines or wires is required for an 8-bit device. For some time a means has been needed that would interconnect these devices through a buffer and provide IEEE 488 Bus compatibility. Such an instrument is now marketed. It is the 488 Bus Interface Coupler, Model 4880, manufactured by ICS Electronics Corporation. This device, illustrated in Figure 7-25, has 40 lines for either input or output and comes in three basic configurations: talker (Model 4881), listener (Model 4882), and talker/listener (Model 4883).



Photo courtesy of ICS Electronics Corporation

Figure 7-25. The Model 4880 Instrument Coupler



Courtesy of ICS Electronics Corporation

Figure 7-26. The PET and the 4880 Coupler in a typical application

The Model 4880 couplers provide all the necessary interface functions to connect any digital programmable device or an instrument's data output to the GPIB. The devices, which will be connected by the coupler to the GPIB, can be talkers, listeners, or talker/listeners. Figure 7-26 shows two, one as a stimulus device (a listener) and the other as a measuring device (the talker).

When the system is assembled, addresses are set into each coupler device (Models 4881, 4882, and 4883) to provide a unique address or code for the instrument to which it is coupled. Thus, the 4880 modules respond (for the coupled devices) to a poll and they accept input or output data depending on the coupled devices' function within the system. Each 4880 module, according to its role in the system, performs the appropriate IEEE functions such as handshake, single or extended address talk or listen, service request, parallel poll, device clear, and device trigger.

With the permission of the ICS Electronics Corporation, the information below is taken directly from the data sheet describing the 488 Bus Interface Coupler Model 4880.

TALKER VERSION

As a talker, the 4880 Coupler can be used to transfer data from any instrument to the IEEE 488 Bus. In this configuration the 4880 can recognize the BUS Device Trigger command and initiate an instrument read cycle or wait for a free running instrument.

At the end of the instrument read cycle, the coupler will generate a service request (SRQ) or output the data if the 4880 has been addressed to talk. The 4880 will respond to a serial poll with a hex 41 character if it has data ready to be sent to the controller, with a hex 0 if no data is available. The 4880's data transmission format can be any mix of data and ASCII symbol characters, up to 16 characters long. The standard 4880 data transmission format consists of 12 characters and is:

DDDDDDDDDDCR LF

Where:

D is either a numeric digit 0-9 or coded as a . + - E or space

CR is carriage return

LF is line feed

The user can order a non-standard transmission format (option 06) or reprogram the UV Prom format table following the directions in the manual. The 4880 talker converts BCD and HEX instrument data into ASCII numbers and characters according to the Talk Character Table. Again, the user can order a non-standard table or alter the character table by following the instructions in the manual.

The 4880 talker versions respond to the universal device clear and selected device clear commands by resetting their internal logic and all device command lines. The response to the universal command, DT (device trigger), is to output a pulse which initiates a reading. The 4880 will asynchronously generate a service request for the controller when the data is ready. The user can inhibit generation of service requests by depressing the front panel SRQ INHIBIT switch. The user can put the 4880 into the 'talk only' mode by setting the ONLY and TALK switches on the rear panel (part of the address switch).

LISTENER VERSION

As a listener, the 4880 can be used to control programmable devices from the bus or to output parallel data. In this configuration, the 4880 can be told to listen only when addressed by the bus controller or at all times. The user can obtain the commanded device status from the 4880 by conducting a parallel poll. The BUS Parallel Poll Enable (PPE) command causes the 4880 to put the current READY Status bit in the poll response word. The 4880 outputs the poll response word during the parallel poll IDY 'on' time.

When addressed as a listener, the 4880 accepts the complete ASCII number set and transfers the least four bits on to the output register. The ASCII numbers 0-9 result in the BCD digits 0-9 on the output connector while the ASCII symbols : ; < = > and ? result in the HEX characters A through F on the output connector. The command string may be up to 10 characters long followed by a delineator (CR, LF or comma) or ended by the EOI signal. The 4880 holds all command characters in latches which are only changed when new data is transferred from memory after receipt of the complete command message.

The 4880 Listener version responds to the universal device clear and selected device clear commands by resetting its internal logic and all command outputs to their initial value. The user can place the 4880 listener into the 'listen-only' mode by setting the ONLY and LISTEN switches on the rear panel.

TALKER/LISTENER VERSION

A 4880 Coupler configured as a talker/listener can be used to control or take data from a programmable measuring device like a programmable DVM. The coupler would accept program commands from the bus and pass them onto the DVM's control inputs, initiate readings and output data when addressed. The talker/listener version combines all the capability and operational features of the talker and listener in one unit. The talker/listener can also be used to control one instrument and input data from a second instrument by properly cabling it to the two instruments. The talker/listener can be reduced to either a talker only or a listener only by properly setting the mode switches on the rear panel.

PET SERVICE REQUEST (SRQ) PROGRAM

Those who are familiar with the PET's internal connections at the IEEE 488 user port J1 know that the PET's Service Request (SRQ) line recognizes an input-only signal. There is no output control of this line by the PET, nor is SRQ implemented in PET BASIC. Thus, a problem exists when a remote device wants to tell the PET, for example, to take the device's readings.

However, connected through the 4880 coupler, a non-bus device (for example a talker) can be made to tell the PET when to take its readings. This is accomplished via the GPIB by a program that tells the PET to monitor the SRQ line. (Location 59427, bit 7, which senses the input of a service request, is tested.) This program allows the remote device beyond the coupler to "probe" the coupler, which in turn makes a service request to the PET. With this set-up, the PET is able

to recognize that the coupler is making a service request, a capability that the PET normally does not have.

To illustrate how such a service request can be made, we connected the 4880 coupler and the PET to the GPIB with the standard bus cabling. The program, which is listed in Figure 7-27, was written to test the coupler's input and output functions.

```
5 PRINT "J":REM CLEAR SCREEN
10 OPEN 5,6
20 INPUT# 5,B#
30 A = PEEK(59426)
40 IF PEEK(59427) AND 128 THEN GOSUB 100
50 PRINT "TESTING"
60 GOTO 40
100 PRINT "J":REM CLEAR SCREEN
105 INPUT# 5,I$
110 PRINT# 5,I$
120 A = PEEK(59426)
130 RETURN
```

Figure 7-27. Service Request (SRQ) program

In this program, the SRQ line in its non-asserted state allows the PET controller to continue performing some off-line operation such as a calculation or data manipulation. When asserted by the 4880 coupler, the SRQ signal is detected by the PET. It then accepts a reading from an external device.

Upon detecting an SRQ, the program clears the PET CRT screen, reads output data from the device coupler, and then immediately prints the data back out to the bus and to the input of the device coupler. The program remains in a wait loop, printing a "TESTING" message continuously to the screen, until an SRQ is received.

In this program, line 5 clears the screen. Line 10 opens a file number 5 for device number 6 residing on the bus. The INPUT statement on line 20 clears any pending service request from the device coupler. Also, any previous transitions that may have occurred on the bus and been stored in the PET are cleared (line 30). Both of these would have interfered with coming operations. The loop at lines 40-60 monitors the SRQ signal line to see if there has been a high-to-low transition. At line 40, location 59427 is tested for bit 7, which senses the input of an SRQ. If no transition has occurred, the word "TESTING" is printed repeatedly to the CRT; the screen scrolls continuously.

If there has been a transition of the SRQ signal, the subroutine at line 100 is called. After clearing the screen (line 100), the subroutine reads input from the device coupler (line 105). Then the program echoes the same data (line 110) to be picked up by the device coupler. Line 120 clears the acknowledged SRQ, and line 130 returns to the wait loop.

OTHER IMPLEMENTATION NOTES

The following paragraphs are adapted from Application Bulletin 48-7, put out by the ICS Electronics Corporation and used with its permission. The material was contributed by Robert Huenemann of United Air Line Maintenance Base, San Francisco, California. The operation described in detail below is based on having several devices take data simultaneously as a result of a Group Execute Trigger (GET) command. The program suggests other possibilities that could provide IEEE functions not directly implemented by PET BASIC statements. Examples are DEVICE CLEAR, PARALLEL POLL and SERIAL POLL.

Group Execute Trigger Program

Not to be confused with the GET statement calling for PET to "get" and examine one character at a time, this GET command is an acronym for Group Execute Trigger. Such a command is often necessary to command all devices on the bus simultaneously and cause them to trigger and start taking readings (for example) at the same time. However, this command is not implemented directly by the PET.

The GET can be accomplished by using standard BASIC statements shown in the program in Figure 7-28.

```
1000 Z=63:GOSUB 7000:REM UNLISTEN
1010 Z=36:GOSUB 7000:REM LISTEN ADDRESS
1020 Z=8:GOSUB 7000:REM GET
1030 INPUT# 4,Z$,Y$
1040 IF Z$ = " " THEN Z$ = ", "+Y$
1050 REM 1030 AND 1040 CIRCUMVENT THE FACT THAT THE PET INTERPRETS ", " AS A
1060 REM DELIMITER, EVEN IN A CHARACTERSTRING. THE FIRST CHARACTER FROM THE
1070 REM 4880 CAN BE ANY OF 16 CHARACTERS IN THIS APPLICATION, INCLUDING ", ".
7000 GOSUB 8000:REM PET TALK HANDSHAKE
7010 IF (PEEK(59456) AND 64) <> 64 GOTO 7010
7015 REM WAIT FOR NRFD FALSE
7020 POKE 59426, 255-Z
7025 REM INVERT DATA AND OUTPUT
7030 POKE 59456, 251:REM SET ATN TRUE
7040 POKE 59427, 52:REM SET DAV TRUE
7050 IF (PEEK(59456) AND 1) <> 1 GOTO 7050
7055 REM WAIT FOR NDAC TRUE
7060 GOSUB 8000:RETURN
8000 REM HOUSEKEEPING
8005 POKE 59427, 60:REM SET DAV FALSE
8010 POKE 59456, 255:REM SET ATN FALSE
8020 POKE 59426, 255:REM SET DATA HIGH
8030 POKE 59425,60:REM SET NDAC HIGH
8040 RETURN
```

Figure 7-28. PET Group Execute Trigger program

In this program, line 1000 executes the universal unlisten command on the GPIB, which tells all listeners to "get off the bus."

Line 1010 first issues the address to the listener that is going to be triggered. This is followed by the GET command on line 1020, which causes the instrument to send a trigger pulse. This GET command causes the 4880 coupler to generate an output strobe, which an external device can use to start its own conversion or triggering function.

The remark at line 1050 states that the small input routine shown on lines 1030 and 1040 will allow characters (such as commas) from the 4880 coupler to be input without the PET interpreting these inputs as delimiters.

At line 7000, a subroutine performs the IEEE handshake procedure at the PET BASIC rate, which is much slower than normal GPIB operation. However, because of the handshake nature of the IEEE Bus, i.e., the sequential, one-signal-dependent-on-another timing, this subroutine yields satisfactory results.

The subroutine at line 8000 returns several of the control lines to their inactive states.

Output Limitations of the PRINT Statement

If you have remote instruments coupled to the GPIB through the Model 4880, you must become aware of the format limitations in the PRINT statement concerning the characters that are sent to the bus. In particular, note that when the PET sends numbers through the coupler to a remote device, a different series of characters is generated at the receiving end from those initiated by the PET. This series of characters depends on the size of the number and whether the number is an integer, floating point, or string.

You should also be thoroughly familiar with the unit before proceeding to write programs or otherwise try to operate it in the GPIB system.* This is because there are unique features in the 4880 instrument. For example, the coupler looks at the most significant digit first and proceeds to the least significant digit, using only the last four bits of each individual character that is sent down the bus. This takes place before the coupler can read in a character and load its output device. Therefore, if the PET places the binary bit pattern 0100 0001 on the bus, the coupler would provide only the last four bits (the 0001) at its output.

*Refer to the 4880 Coupler Instruction Manual.