Asynchronous and Synchronous
Auto Dialing Equipment on the HP 3000
Why, When, and How

Benedict Bruno
STR Software Company
PO Box 12506
Arlington, Virginia 22209

Most HP 3000 computer systems have asynchronous modems connected to CPU ports to support remote terminal users. If connectivity to other HP 3000 or IBM mainframe (or compatible) computers is desired, then synchronous modems may also be connected to the Intelligent Network Processor (INP) accessed by subsystems such as Remote Job Entry (RJE), Multi-leaving Remote Job Entry (MRJE), and Distributed Systems (DS).

Very few people truly understand the fundamentals of data communications. Even fewer understand how the task of dialing the requested telephone number is performed. Believe me, it is not peformed by "magic", but rather by several different well defined methods.

In this presentation, I will describe how automatic dialing equipment performs in both asynchronous and synchronous environments. This will include the RS-366 interface as defined by the Bell 801C automatic call unit specification guide, the Hayes modem (and compatible) automatic call facility, and other dialing systems. In particular, proper MPE I/O configuration, troubleshooting suggestions, and sample programs to support automatic calling from an asynchronous terminal port will also be discussed.

Asynchronous Background Information

The HP3000 is not a full duplex machine with respect to asynchronous communications. This should not be a surprise to any of you now that you have been using your HP3000 for quite some time. Let us review how the HP3000 actually communicates with an HP (or compatible) terminal.

Since the HP3000 is not a full duplex machine, the designers created a flow control and character echo scheme that is quite unique in the computer industry. The scenario is the following: the terminal may only send when the HP3000 says it is ready. Furthermore, the HP3000 may only send data in a maximum of 80 character transmissions whenever the terminal acknowledges that he is ready. This strategy is known as the HP inquiry and acknowledge protocol or better known as ENQ-ACK. This strategy is found in Figure 1 below:

```
        HP3000                                Terminal

 FWRITE of data where
 data <= 80 characters ---------->
 inquiry (ENQ) ------------------>

                                  <----------- OK, send more (ACK)

     (Repeats until all data sent from CPU to terminal)


  FREAD informs the terminal
  that the HP is ready for data

  Read trigger (DC1) ------------>

                                              Terminal sends data
                                  <------------ terminated with CR
```

Figure 1: HP3000 ENQ/ACK protocol

Notice how this protocols works: The terminal connects to the CPU (either modem or hardwired) and issues a carriage return. The HP3000 senses the correct speed and parity and echoes the received carriage return and adds a line feed. The HP3000 is then ready to receive an MPE command (the :HELLO command) by issuing the read trigger, i.e., the DC1 character. This read trigger is actually sent to the terminal when the FREAD (or READ or READX) intrinsic is executed. This DC1 character inhibits or unlocks the keyboard in order for you the terminal user to actually begin keying your command. Until this DC1 character arrives, you cannot begin keying any data.

The HP3000 will echo each character that you type at the keyboard. Depending upon the asynchronous terminal driver that you are using, this echo is implemented differently. The Asynchronous Terminal Controller (ATC) on the Series II and Series III machines would pass every received character directly to the CPU in order for the character to be echoed. Thus each character keyed at the keyboard would interrupt the CPU. This same technique is employed on the Asynchronous Data Communications Controller (ADCC) on the Series 30,33,40,44,48. With the creation of the Asynchronous Terminal Processor (ATP) the CPU is no longer interrupted for each character. Notice the term "processor"! The ATP provides the character echo directly to the terminal user. When the data is completed with the carriage return, the ATP then interrupts the CPU by passing the data buffer directly. You can see the obvious advantages of this method.

Now you are all experts on the ENQ/ACK protocol. But how do we attach a modem to an asynchronous port on the HP3000? More specifically, to an ATC, ADCC, or ATP port? Well, this does not seem that hard but we need some more background information.

All devices in the world of computers fall into two categories: either a DTE or DCE. Loosely defined, a device is classified as data terminal equipment (DTE) if it provides computer processing support (a CPU) and user input/output (a CRT). A device is classified as data communications equipment (DCE) if it provides transmission capability (modems, PBX, and so on). So what is the HP3000? A DTE of course. Wrong! Why? The HP3000 actually is a DTE in function, but it is cabled as a DCE device. This causes alot of people to incorrectly construct their cables. Why did HP do it this way? Because an HP terminal is a DTE device and a modem is a DCE device. If the HP3000 is cabled as a DCE device, then the same cable could be used between the terminal and the HP3000 or between the terminal and the modem!

Figure 2 below shows the data flow relationship between a DTE and DCE. The 25-pin DB25 connector usually associated with RS-232-C devices specifies that pin 2 is used for transmit data by the DTE device. Furthermore, pin 3 is used for receive data by the DTE device. A DCE device is exactly the reverse. Notice that the direction of the arrow points to the DCE for pin 2 and to the DTE for pin 3. This is fundamental in constructing our cable between the HP3000 and the modem.

```
Pin #       DTE        DCE

 2       ------->    ------>

 3       <-------    <------
```

Figure 2: DTE vs DCE data flow

But how do I connect a real DCE device such as a Hayes modem to the HP3000, that is, how does one connect two similar devices? Using Figure 1 and the state fact that a DTE transmits on pin 2, two like devices (DTE to DTE or DCE to DCE) must cross the transmit and receive pins. This crossover is simply constructed with pin 2 on the CPU end wired to pin 3 on the modem end. Next pin 3 on the CPU end is wirted to pin 2 on the modem end. The remaining signals necessary for modem controls must also be crossed as well. This includes pins 20 (data terminal ready or DTR) and pin 6 (data set ready or DSR), pin 4 (request to send or RTS) and pin

8 (carrier detect or CD), while pin 7 (signal ground or GND) is wired straight through. This cable is simply known as the HP3000 dataset cable with part number HP30062B. It is diagrammed in Figure 3 below. Notice that the additional signal of ring indicator (RI) on pin 22 is crossed with clear to send (CTS) to provide uniformity with products overseas and in the United States.

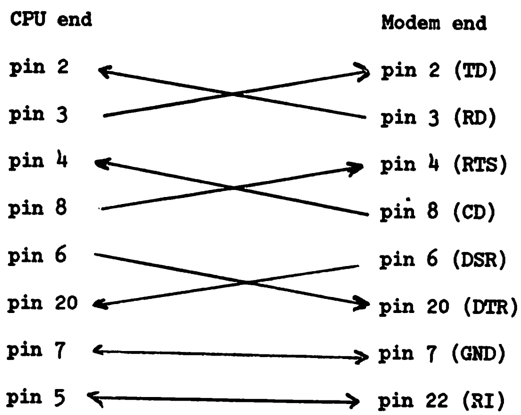


Figure 3: Connection of a modem to an HP3000 CPU port

MPE I/O Configuration

Now that the HP3000 is connected with the data set cable to the Hayes modem, the next step is to configure the modem within the MPE I/O configuration. The MPE :SYSDUMP program is used to modify the I/O configuration. The System Operation and Resource Management reference manual documents the :SYSDUMP program and its options. The items of importance here are the terminal type, subtype, and speed.

The terminal type of 10 specifies the standard supported HP terminal protocol of ENQ/ACK. The terminal type of 18 specifies any non-HP type of terminal and does not use ENQ/ACK. Specify the terminal type based upon the device that this modem will be calling. Since the incoming :HELLO command can specify the terminal type (;TERM=nn) and the outgoing program can specify the terminal type (FCONTROL 10), the terminal type should be for the device that will mostly use this port.

The subtype specifies whether the device is hardwired connection or a modem connection. Subtype of zero (0) specifies a hardwired connection and the HP3000 requires only three computer signals of transmit data (pin 2), receive data (pin 3) and signal ground (pin 7). Thus only three wires are necessary to connect an HP (or compatible) terminal to the HP3000 CPU using RS-232-C. This fact reduced the connector size for what HP classifies as "direct connect" ports on the ATP. Direct connect ports on the ATP use a special three pin connector rather than the standard DB25 connector associated with RS-232-C. Furthermore, with only three signals, the additional modem signals of data terminal ready, data set ready, carrier detect, etc. are not provided.

Could we attach a modem to either the three pin direct connect or the 25 pin "modem port" on the ATP (or ADCC or ATC ports) and configure the MPE I/O configuration for this port as hardwired? Certainly, but remember that the HP3000 CPU will not provide data terminal ready to the modem. Thus the Hayes modem would have to be strapped for forced data terminal ready. This means that the Hayes modem will always answer an incoming call regardless if the HP3000 CPU is really up. Not really a good idea.

The proper way to attach a modem to the HP3000 is with the subtype of 1 specifying modem support. All modem signals are supplied with subtype 1. Remember from our earlier discussion that all the signals are reversed at the CPU end! The CPU monitors data terminal ready raising the signal when the HP3000 is running. The signal is momentarily dropped when a disconnect occurs, when the remote user issues the :BYE command, or when the :ABORTJOB command is used on the session number of the remote user active on this port. This sounds more of what we want, control of our session in the case of disconnects and logoffs.

The last item, speed, is really of no consequence because MPE will "sense" the speed from that specified with the character string received at the CPU. A Hayes modem supports speeds of 300, 1200, and 2400 baud. Depending upon the speed of the remote terminal user, the local modem at the CPU will synchronize with the remote modem at the terminal. The HP3000 CPU will then sense the correct speed and echo the carriage return with a line feed and the colon prompt.

Asynchronous Auto Dialing

Our objective is to have a program open the logical device number with our Hayes modem attached. With our configuration parameters above, can we do it? Almost. The Hayes modems provide a simple command mode interface that can easily be written by any DTE device. We can specify any speed of 300, 1200, and 2400 baud with the FCONTROL intrinsic. But we cannot open the port and issue an FWRITE (with the Hayes "AT" commands) without the HP3000 knowing that the "unit (modem) is ready". By this we mean that the HP3000 must see carrier detect (CD) and data set ready (DSR) high from the modem in order for the FWRITE to complete.

The Hayes modem has dip switches specifying a number of options. One of the options specifies the signals of carrier detect and data set ready as either normal or forced on. We need this to be forced on. While we are at it, specify the dip switches as in Figure 4 below:

DTR from interface
Terse command mode responses
Enable command mode responses
Disable echo of command characters
Auto answer enabled
CD and DSR forced high
Enable command mode

CTS forced on
Dial-up operation
Blind dialing method
Asynchronous operation

Figure 4: Hayes modem strap settings

Notice that DTR is specified from the interface, i.e., from the HP3000. With subtype 1 the HP3000 provides this signal; otherwise we would have to force it high for subtype 0. Commands executed by the Hayes modem always return a response. This response can be a character string such as "OK", "CONNECT", etc. or can be a single digit also known as terse mode. In terse mode, the Hayes modem transmits a single ASCII character terminated by a carriage return. Just what we want, data terminated with a carriage return. Obviously we want these responses to our command and we do not want the command characters echoed back to the CPU. The remaining signals are self explanatory. One should notice that the only signal we MUST have is that of CD and DSR forced high. All of the remaining signals can be specified with an appropriate AT command that we may send programmatically.

<u>Program Pseudo Code</u>

Now that we have the MPE I/O configuration correct, the Hayes modem strapped, and the cable connecting the modem and the CPU, we are now ready to develop our program to use the autodialer and control the modem. will describe this as a series of steps that can be programmed in any language on the HP3000.

<u>Step 1</u>: Open the port

We must first open the logical device number assigned in our I/O configuration to the attached modem. The FOPEN intrinsic may be used with a file reference to specify which logical device number the modem is on as follows:

```
:FILE PORT;DEV=ldevnumber
filenum = FOPEN ("*PORT, %400, %4, -80)
```

Notice that the FOPEN intrinsic references the formal designator of "*PORT" requiring that a file equation for PORT be present. The file options (FOPTIONS) of octal 400 (%400) specify carriage control and a new file. More on carriage control later. The access options (AOPTIONS) value of %4 allow us to read and write to this device. The record size of 80 bytes is for a standard terminal. You may need to increase this to talk to your device.

Step 2: Disable echo

Since the modem will respond to our AT commands with a response, we do not want the HP3000 to echo back this response to the modem. Recall, the HP3000 always echoes incoming data back to the sending device. This will obviously confuse the modem if this data is echoed back! Use the FCONTROL intrinsic with the returned file number from step 1 as follows below:

```
FCONTROL (filenum, 13, ldummy)
```

Notice that parameter 13 disables the echo feature. No parameter value is required to disable echo. Hence the value of the "ldummy" variable is meaningless and unused.

Step 3: Specify modem speed

Recall that the modem supports a variety of speeds. We must specify the speed of the modem to match that of the device that we are trying to call. The FCONTROL intrinsic allows for the specification of the input and output speed. As it turns out, the HP documentation states that specification of both the input and output speed may cause problems. Therefore, we only need specify the output speed with option 11. The speed is actually specified in characters per second (not as the baud rate) in the ldummy parameter. Values of 30, 120, and 240 characters per second correspond to 300, 1200, and 2400 baud respectively. This can be done as follows:

```
ldummy = 30
FCONTROL (filenum, 11, ldummy)
```

Step 4: Disable MPE automatic CR/LF

For each record received by the HP3000, i.e., for each read completed with a carriage return from the device, the HP3000 will echo an automatic carriage return and line feed back to the device. Obviously, this will create havoc for the modem when we have just received the command response. This automatic carriage return and line feed can be disabled with the FSETMODE intrinsic as follows:

```
FSETMODE (filenum, %4)
```

Step 5: Change DC1 read trigger to CR

Recall from the background discussion earlier, that the HP3000 will transmit the DC1 character as the read trigger. The remote device may only transmit data to the HP3000 when this read trigger character arrives. The Hayes modem has no idea what to do with the DC1 character. The solution is to define some character that the HP3000 may send as the read trigger that will in turn complete the information that the modem wants. How about a carriage return? By using a carriage return as the read trigger, whatever data was previously transmitted by the HP3000 in the FWRITE will now be complete from the FREAD! Not only that, but we are guaranteed at the HP3000 of being ready for the modem response.

The DC1 read trigger is changed to a carriage return with the FDEVICECONTROL intrinsic. This intrinsic accesses the Work Station Configurator in order to perform the change. The Work Station Configurator is a nice product supplied by HP to change many of the attibutes in communicating with an asynchronous device. These "changes" are placed into terminal type or TT files in PUB.SYS and can be referenced programmatically and within the I/O configuration.

The FDEVICECONTROL specifies that the read trigger character be placed in an integer variable (16 bit word). The actual character is placed in the second half of the word, i.e., bits 8 thru 15. The first half of the word is ignored and should be set to binary zero. The parameter of 192 specifies the Work Station Configurator. The parameter value of 32 specifies the read trigger should be accessed. The parameter value of 3 specifies first change the read trigger to that supplied in I and then display the value in I. This validates the change. If an error occurs, the value is returned in the error variable which can be used in a call to the FERRMSG intrinsic to locate the message from the error catalog.

```
I=%000015
FDEVICECONTROL (filenum,I,1,192,32,3,error)
```

Step 6: Enable terse messages; disable command echo

As discussed earlier, using terse messages from the Hayes modem yields a single ASCII digit response to our AT commands. This digit response is completed with a carriage return. The modem must not echo the command responses because the HP3000 will think it is valid input from the remote device. The echo of command data should be disabled. Terse messages and disabling command echo can be requested with a single command to the modem. Notice that the data is not terminated with a carriage return (nor a line feed) as specified with the FWRITE parameter of octal 320. Why? Because we will transmit the carriage return as the read trigger when we execute the FREAD intrinsic.

```
FWRITE ("ATV0E0", -6, %320)
```

Step 7: Read digit response terminated with CR

Once the modem completes our command in step 6, a single ASCII digit is returned terminated with a carriage return. This can be requested by the HP3000 using a two character terminal read. The received data can be checked against the valid responses supplied by the modem manufacturer. The Hayes specification states that zero (0) indicates successful completion of the previous command.

```
FREAD (answer,-2)
```

Step 8: Dial the number

Now that the modem is ready, we can simply send the dialing sequence. This dialing sequence will include the digits of the telephone number, delay characters, wiqait for second dial tone, etc. Place the dialing sequence in a buffer and send it to the modem with the FWRITE intrinsic as in step 6 above. Note that the FWRITE specifies no carriage return.

```
FWRITE ("ATDT 1-703-689-2525", -length, %320)
```

Step 9: Read the modem response

As before, the previous command is completed by issuing the FREAD intrinsic. The FREAD transmits the read trigger character (carriage return) and awaits a two character response from the modem. The modem places the call and determines status. Locate the list of valid responses for your particular modem. You may also wish to enable a timed read for the modem response. Use the FCONTROL with a parameter of 4 and an integer value for the number of seconds that the HP3000 must wait for a response.

FREAD (answer,-2)

Step 10: Continue programming!

Assuming a connected response from the modem, you are now able to continue with your program. Don't forget now that you are connected, the carriage return may be inappropriate as the read trigger. This depends on the device to which you are communicating.

Should you wish to disconnect abnormally, the Hayes modems may be interrupted while connected and return to command mode. The default string of three plus characters (+++) when received by the modem invokes command mode. How might we do this? Simply issue a two character FWRITE of "++", change the read trigger with FDEVICECONTROL to "+", and then issue the FREAD. You will then be in command mode. Remember to then change the read trigger back to a carriage return in order to continue with the AT disconnect command.

Synchronous Auto Dialing

The Intelligent Network Processor (INP) provides all synchronous communications capabilities for the HP3000. One of the most important features of the INP is its ability to establish the telephone call of switched (dialup) telephone lines automatically, eliminating manual intervention. Autodialing with the INP is completely different than that described above with the Hayes autodialing method. As before, let's describe how to configure and cable the INP for autodialing. Then continue with a discussion of the auto call unit itself.

Configuring the INP

Depending upon the HP3000 series computer, different hardware and software capabilities are available on the INP. The INP for the Series III requires two boards which unfortunately do not support the automatic call feature. The INP for the Series 30, 33, 4X, 6X and 70 computers has been available as the original non-auto dial feature (part number 30020A or 'A' board) and as the current auto dial feature (part number 30020B or 'B' board). The INP board for the Series 37 also provides auto dial support but is completely different from the 'B' board and cannot be interchanged. The reader should note that HP offers the INP within its 'Link Services' product offerings making it somewhat easier to bundle the INP, cable, and download software within one product.

The :SYSDUMP command invokes the MPE I/O and system configurator program. Many parts of this dialogue are also supplied with any of the system startup procedures, i.e., WARMSTART, COOLSTART, UPDATE, COLDSTART, and RELOAD. I am sure that most of you have created or updated the system I/O configuration of your own HP3000. However, should you wish to add the automatic calling feature to your INP, you may notice that it is readily documented within any of the HP3000 manuals. This includes the INP Installation Reference Manual. So what is the trick? How does one do it?

The three prompts of 'DIAL FACILITY', 'ANSWER FACILITY', and 'AUTOMATIC ANSWER' are provided by the SYSDUMP dialogue in order to properly configure the dial and answer capabilities of the equipment connected to this INP. Obviously the 'DIAL FACILITY' prompt appears related to specifying that this INP is to utilize the automatic calling feature but the System Operation and Resource Management Reference Manual and the Data Communications Handbook specify that values of 'YES', 'NO', or [RETURN] are the only valid responses. The YES response specifies that a telephone handset is attached in order for the operator to manually dial the phone; the NO response is identical to the [RETURN] response in that no telephone handset is attached. The reader should note that MPE issues a system console request with the specific phone number that must be dialed. When the connection is made, the operator simply replies to this console request.

However, this still does not answer the question of enabling the automatic dialing feature. Since HP did not want to add another prompt to the dialogue, the developers decided to allow another value. By supplying the logical device number of the INP to the 'DIAL FACILITY' prompt (which is obviously the value supplied to the first prompt of 'LOGICAL DEVICE #'), we enable the INP to support automatic dialing.

Furthermore, how can you determine from a SYSDUMP (or for that matter SYSINFO) listing of the I/O configuration and CS devices which INPs have the automatic calling feature enabled and which do not? Consider the output from the SYSDUMP program in Figure 5. Two identical INPs are displayed utilizing 4800 bps modems. One INP provides automatic dial support and the other provides manual dial support. Which is which? You guessed it, there is no way to determine it! When a customer explained having difficulty in establishing the automatic dialing feature on the INP, the first thing I did was to configure the entry once again. This way I was certain that the I/O configuration was correct.

Figure 5: List I/O and CS Devices

DEV #	#	NIT	HANE	TYPE	TYPE	TERMINAL TYPE	SPEED	WIDTH	DEV	NAME	CLASSES
11	181	0	0	17	0			0	0	IOINPO	INP
12	181	0	0	17	0			0	0	IOINPO	INP

LDN	PM	PRT	LCL MOD	TC	RCV TMOUT	LCL TMOUT	CON TMOUT	MODE	TRANSMIT SPEED	TM	BUFFER SIZE	DC	DRIVER OPTIONS
11	0	X	X	X	20	60	900	OIA	600	1	1024	N	0
12	0	X	X	X	20	60	900	OIA	600	1	1024	N	0

Notice that most of the columns are self explanatory. I was happy to see that with the Update #1 (January 1985) release of the System Operator and Resource Management Reference Manual the values of the 'MODE' field are now documented. The reader is referred to page 7-4 for a description of each of the heading and entry values. The mode of 'OIA' specifies dial out, manual answer, and automatic answer. We do not know if the auto dial feature is enabled or disabled. Maybe an additional mode value of say 'U' (from aUtomatic dial) can be added to the list and displayed if the automatic dial feature is enabled.

The reader should also notice that both logical device numbers of 11 and 12 specify the same Device Reference Table (DRT) entry. Thus an INP may have multiple configurations in order to support several devices and capabilities. Therefore, you may enable auto dial support on ldev 11 and disable auto dial support on ldev 12. I strongly suggest that you add meaningful device class names of 'AUTODIAL' and 'MANDIAL' respectively in order to avoid further confusion!

The reader should also beware that if the auto dial feature is enabled on ldev 11, that MPE will expect to find an automatic calling unit attached. If not, then nothing will happen. No console request will be issued for the operator to dial the remote number should you attach a telephone handset to place the call manually. The same is true if the INP is configured without auto dial support and you connect an automatic calling unit to the INP.

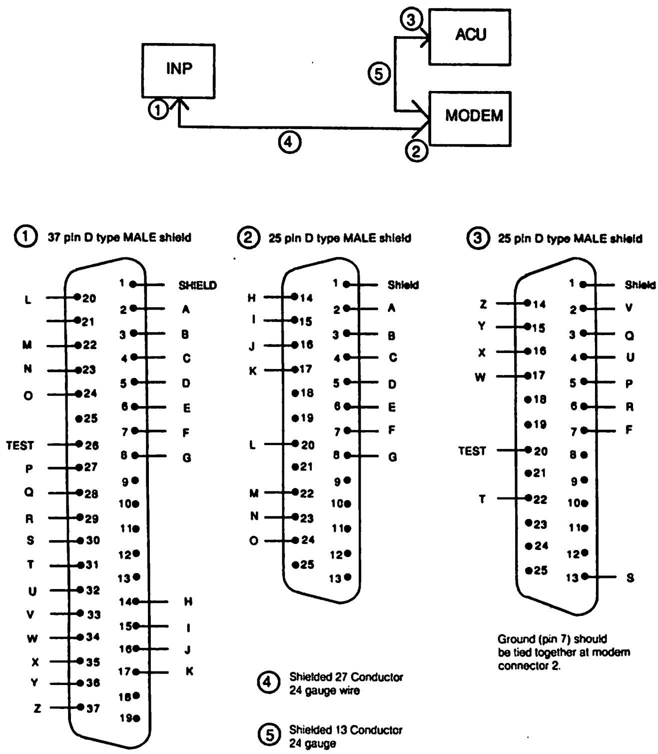
HP30221G Auto Dial Cable

The automatic dial facility requires a special cable between the INP junction panel connector and the necessary external equipment of the modem (2400bps, 4800bps, etc) and the automatic calling unit. This is the first mention of the automatic calling unit. Synchronous modems were developed requiring an external device place and control the dialing of the telephone number. This is far different than many of the asychronous modems of today that are 'Hayes compatible' in that the modem and the dialer are all in one nice package.

The defacto standard for the automatic calling unit is the Bell 801C auto call unit (ACU). Connection to the Bell 801C ACU requires the RS-366 interface. This interface is well documented in the Bell 801C-LI/2 Data Auxiliary Set Interface Specification, PUB 41203.

Although the RS-366 interface utilizes the same DB-25 connector as the RS-232-C interface, the two are vastly different. Attachment of the ACU and modem to the INP requires the HP30221G cable (or 'Y' cable as it is often called) as diagrammed in Figure 6. Notice that the connectors differ at each end of the cable. The connection to the INP requires a 37 pin connector in order for all 27 signals be passed to the INP. The connection to the ACU and the modem utilize the standard 25 pin male connector. However, notice that the pins used by the ACU appear similar to the modem signal usage. This appearance is totally misleading.

Figure 6: HP30221G cable schematic



The pin designation of the RS-232-C standard is diagrammed in Figure 7. Notice that the direction of the signal clarifies whether it is an inbound or outbound signal. For example, the HP3000 raises data terminal ready (DTR) on pin 20. The modem then reacts by raising data set ready (DSR) on pin 6 when the link is established. The remaining signals will not be discussed.

Figure 7: RS-232-C Pin Designations

Pin	Name	To INP (<---) To modem (->)	Description
1	FG		Frame ground
2	TD	------------->	Transmit data
3	RD	<-------------	Receive data
4	RTS	------------->	Request to send
5	CTS	<-------------	Clear to send
6	DSR	<-------------	Data set ready
7	SG		Signal ground
8	DCD	<-------------	Data carrier detect
9		<-------------	Positive DC test voltage
10		<-------------	Negative DC test voltage
11			Unassigned (not used)
12	SDCD	<-------------	Secondary data carrier detect
13	SCTS	<-------------	Secondary clear to send
14	STD	------------->	Secondary transmit data
15	TC	<-------------	Transmitter clock
16	SRD	<-------------	Secondary receive data
17	RC	<-------------	Receiver clock
18		------------->	Receiver dibit clock
19	SRTS	------------->	Secondary request to send
20	DTR	------------->	Data terminal ready
21	SQ	<-------------	Signal quality detect
22	RI	<-------------	Ring indicator
23		------------->	Data rate select
24	TC	------------->	External transmitter clock
25		------------->	Busy

The pin designation of the RS-366 standard is diagrammed in Figure 8. Notice that the signals do not resemble any of those specified in the RS-232-C standard in Figure 7.

Figure 8: RS-366 Pin Designations

Pin	Name	To INP (<---) To ACU (--->)	Description
1	FG		Frame ground
2	DPR	------------>	Digit present
3	ACR	<------------	Abandon call, retry
4	CRQ	------------>	Call request
5	PND	<------------	Present next digit
6	PWI	<------------	Power indicator
7	SG		Signal ground
8			Unassigned, not used
9			Positive DC test voltage
10			Negative DC test voltage
11			Unassigned (not used)
12			Unassigned (not used)
13	COS	<------------	Call origination status
14	NB1	------------>	Value 1 of digit
15	NB2	------------>	Value 2 of digit
16	NB4	------------>	Value 4 of digit
17	NB8	------------>	Value 8 of digit
18			Unassigned (not used)
19			Unassigned (not used)
20			Unassigned (not used)
21			Unassigned (not used)
22	DLO	<------------	Data line occupied
23			Unassigned (not used)
24			Unassigned (not used)
25			Unassigned (not used)

The pin assignments in Figure 8 require some explanation. The digit present (DPR) signal on pin 2 is set high by the INP whenever the digit is sent by the INP on pins 14, 15, 16, and 17; otherwise it is set low.

The abandon call retry (ACR) signal on pin 3 is set high by the ACU to indicate the probability of an unsuccessful completion of the call attempt. The INP alerts the user with a CS error 59 suggesting another call attempt.

The call request (CRQ) signal on pin 4 is set high by the INP to request the ACU to originate a call. This signal remains high througout the entire data communications transfer and is set low when the INP wishes to disconnect the telephone.

The present next digit (PND) signal on pin 5 is set high by the ACU to control the presentation of digits on the digit signal circuits. When set on, the ACU is ready to accept the next digit indicated on pins 14, 15, 16, and 17 as set by the INP. The digit is read when the INP sets DPR high. When set low, the INP may set low DPR and again present the next digit on pins 14, 15, 16, and 17.

The power indication (PWI) signal on pin 6 is set high when the ACU detects available power.

The call origination status (COS) signal on pin 13 is set high when the ACU has completed the call request function. Previously, this signal was named data set status (DSS) and later renamed to the call origination status. Once set high by the ACU, the INP begins its data transfer depending upon the specific protocol being emulated, i.e., 3780 on RJE, HASP on MRJE, etc.

The digit signal circuits on pins 14 (NB1), 15 (NB2), 16 (NB4), and 17 (NB8) present the binary coded value of the digits 0 through 9. Obviously, this is the reason why NONE of the HP data communications subsystems (RJE, MRJE, DS, IMF, and MTS) allow for any characters in the phone number other than the digits separated by the dash character. Thus, the Hayes modem phone number values such as 'k' or ',' for pause cannot be issued. This misconception was always a common question asked by customers when I was at HP.

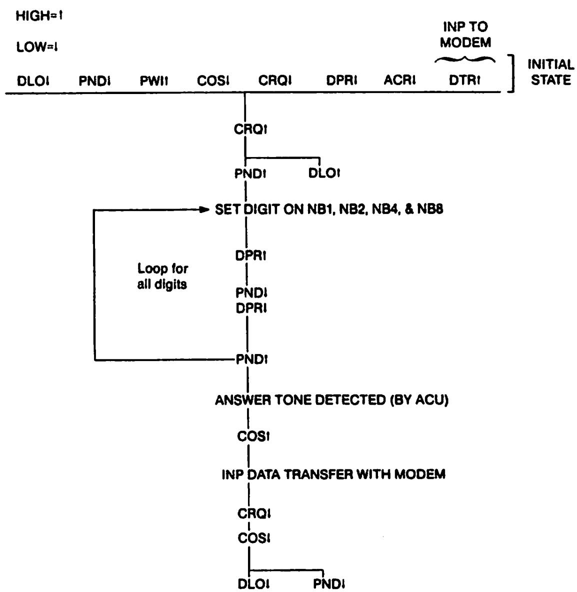
The last signal of data line occupied (DLO) on pin 22 is set high indicating that the data communications channel is in use for automatic calling, data transfer, voice, or testing. When set low and the power indication signal is set high, i.e., the unit is powered on, then the INP may initiate a call request.

Now that the signals are discussed, let me simply explain the automatic calling process between the INP and the ACU. The process is diagrammed in Figure 9. The initial signals of the ACU are set low when the power is off. When the ACU detects available power when the unit is turned on, then the PWI signal on pin 6 is set high by the ACU. This will be the true 'idle state' of the ACU when not in use by any of the HP3000 data communications subsystem software. The INP initiates the call request by setting the call request signal on pin 4 high. The ACU responds by setting both the present next digit signal on pin 5 and data line occupied signal on pin 22 high. The data line occupied signal remains high throughout the entire call until the INP terminates the line. The ACU requests the value of the telephone digit when the present next digit signal is high. At this point the INP places the digit in the digit signal circuits on pins 14, 15, 16, and 17. Once complete, the INP incates the digits are present by setting pin 2 high. Once detected by the ACU, the present next digit and digit present signals are set low by the ACU and INP respectively. This process repeats for each of the digits of the telephone number to be dialed, i.e., ACU sets high PND, INP sets digit signal circuits and high DPR, ACU sets low PND, and INP sets low DPR. Once the phone number is complete, the ACU attempts the call.

The ACU must indicate successful completion or failure to the INP. The call origination status (COS) signal on pin 6 is set high if successful or low for a failure. In addition, the abandon call retry signal on pin 3 is set high for a failure.

If successful, then the ACU is released from the data transmission circuit in that the INP now sends and receives signals from the modem. When the INP is ready to terminate the data communications activity, the DTR signal is set low which releases the modem and the CRQ signal is set low which releases the ACU. The ACU responds by setting low the call origination status, data line occupied, and present next digit. This leaves all equipment in the 'idle state' ready for the next call request.

Figure 9: INP-ACU Interface



Supported External Modems and ACUs

Hewlett-Packard tested and certified specific modems and auto call units when the auto dial support was added to the INP in late 1981. Vendor equipment from both Bell and Racal-Vadic were used. Specifically, the Bell 801C and Racal-Vadic VA811 auto call units were certified in connection to the INP. Lab engineers connected the Bell 201C and Bell 208B synchronous 2400 bps and 4800 bps modems to the INP.

HP concluded that any Bell 801C compatible ACU and Bell 201C or Bell 208B compatible modem are supported by the INP. However, customers were warned that other vendor equipment must be tested and certified. This process could be performed with HP assistance on a time and materials basis.

Case History Example

Over the years, I have been involved with several customers attaching equipment other than Bell and Racal-Vadic to the INP for automatic dialing of the HP data communications products. Each of the HP sales offices used either the Bell 801C or the Racal-Vadic VA811 with no installation or operational difficulties. However, I have experienced some problems with testing and certifying other vendor equipment to the INP. Most of these were resolved, but by far the most interesting and troublesome case deserves mention here.

MCI Telecommunications, Inc. had already been using nearly every data communications product to implement the MCI Mail Electronic Mail Network (see "Integrating HP Data Comm for Electronic Mail", INTERACT, October 1986). However, MCI wanted to automatically dial remote computers using DSN/RJE with the 3780 protocol.

Rich Oxford, MCI Technical Support Manager, contacted HP for assistance when the Penril 8208 modem (Bell 208B compatible at 4800 bps) already installed and working on the HP3000 Series 64 could not function with the Penril 8801C ACU (Bell 801C compatible) with the INP configured for auto dial. Tom Benedict, HP Network Consultant and I (HP Senior Applications Engineer at that time) responded.

We first determined that the I/O configuration was indeed correct. When RJE attempted the dial sequence from the #RJIN command, the ACU and modem performed the correct signals as in Figure 9. The INP appeared hung, experiencing a brief period of idle activity terminating in CS error 59. This included the call request by the INP, the present next digit by the ACU, the digit sequence, and so on. The ACU just never released control of the telephone line to the modem in order for the INP to continue.

We suspected either the ACU or the MCI constructed cable. The ACU and modem were sent to Penril for repair while we checked the cable on the HP sales office machine. The cable tested fine with the HP3000 Series 68 in the Rockville Sales Office. The ACU and modem were verified from the Penril engineers. So what was the problem?

After no success on site, we connected the MCI equipment on the HP3000 Series 68 at the Rockville Sales Office. Again with no success. At this point, Tom and I reached the time committment of the support contract and suggested that all continuing work would be on a time and materials basis. (I offered to assist Rich at night on my own time to develop a solution. I was intrigued by this one!)

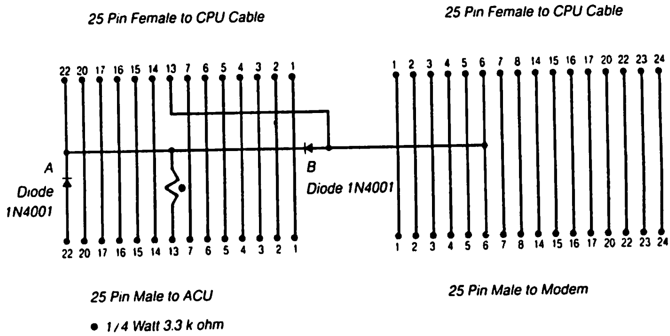
Returning to MCI, Rich and I studied the 801C specification directly from the Bell technical publication given to us from Tom. Rich decided to place breakout boxes and data scopes between both the ACU and the modem. This permitted us to not only monitor but bridge any signals from the equipment.

We solved the problem with some clever observations by Rich. The call origination status (COS, pin 13) is always low on the Penril ACU. Data line occupied (DLO, pin 22) is set high by the ACU when the INP initiates the call request (CRQ, pin 4). In the previous discussion of Figure 9, the 801C specification clearly states that DLO must remain high by the ACU during the entire call. Using the breakout box between the ACU and the INP, Rich noticed that when the ACU detected answer tone thereby releasing the phone line to the modem, the DLO signal was set low! Which is totally wrong based upon the Bell specification, i.e., not truly Bell 801C compatible! How could this ever work? Both vendors (HP and Penril) specify their equipment worked satisfactorily and yet we noticed that they did not worked together!

Therefore the Penril failed on two Bell 801C requirements. DLO must remain high during the entire call and secondly COS follows the DSR signal. The trick is to make the DLO signal appear high to the INP when the Penril equipment mistakenly sets it low. This must happen once the ACU detects answer tone thereby releasing the line to the modem. The modems synchronize which sets the DSR signal high. At this point, the COS and DLO signals were jumpered with DSR (pin 6) of the modem cable and IT WORKED! The COS, DLO, and DSR signals all remained high, the INP detected connection, and RJE continued with the user commands. The line terminated correctly when the RJEND command was encountered.

The final conclusion involved the construction of an adaptor cable between both the ACU and modem to the INP cable as diagrammed in Figure 10. Notice that two 1N4001 diodes and a 3.3k ohm resistor are required in this cable. Using layman terms, let me explain how the cable works.

Figure 10: Penril adapter cable



The resistor and pair of diodes are connected as a hardware logical OR gate. A diode permits positive voltage to flow through the direction of the arrow. When DLO (pin 22) goes high by the ACU, diode A permits this voltage to flow through to the pin 22 at the CPU end. Furthermore, this positive voltage cannot pass through to pin 6 of the modem nor pin 13 because of diode B (notice the direction of the arrow). When the call is answered, DLO goes low by the ACU, but the DSR signal is set high by the modem. This positive voltage passes to COS (pin 13) and through diode B in effect making DLO appear high to the INP. In fact, the INP never even notices that the signal drops at all!

The cable corrects the misinterpreted implementation of the Bell specification by the Penril ACU. The resistor is necessary in order to bias the diodes. The call origination status (COS) signal was previously named distanct station connected (DSC) which typically is set the same as data set ready (DSR). This explains why the COS lead on the CPU end is tied to the DSR lead in order to make the COS signal follow that of DSR. Incidently, total cost of the components from Radio Shack was only $2.20.

The Bell 801C ACU is implemented by a number of vendors. As you can see, its actual function is quite well defined.

Additional Features

Synchronous modems have been further enhanced during the past several years. Modems have become faster, cheaper, smaller, and more efficient at utilizing switched telephone lines. One of the most notable enhancements to the synchronous modems is the addition of a built in auto dialer which elminates purchasing a separate 801C type ACU. In fact, a 4800 bps (Bell 208B compatible) synchronous modem with built in 801C ACU is packaged even smaller than the older 801C ACU from the Bell System! (Of course, even less expensive.)

In addition to combining the auto call unit function within the modem itself, manufacturers are now offering other types of dialing interfaces. It is possible to utilize an asynchronous port to control the auto dialing function of a rack of synchronous modems (usually 16 modems). The synchronous data comm products on the HP3000 require the use of an INP and does not allow for this asynchronous type of auto dial control.

Another type of dialing control mechanism is implemented directly by the modem only. In this particular method, the host CPU initiates a synchronous dialogue to control the auto dial feature of the modem. This feature was pioneered by Racal-Vadic as the Synchronous Auto Dial Link or better known as SADL. The SADL method provides IBM bisynchronous and HDLC protocol support. Using SADL with bisynch, the host computer would control the modem as if it were another computer in that it would send to and receive from the modem auto dial information directly. Once the modem makes the remote connection, the auto dial function has been completed and control is no longer needed.

SADL is a very effective and interesting method of controlling the auto dial function for synchronous lines. However, in the current implementation of the HP3000 RJE product by HP, it cannot be used as a viable dialing option from the INP. We must resort to the standard 801C auto call unit.

Summary

In this paper I have discussed the types of auto dialing equipment in use in both asynchronous and synchronous environments on the HP3000. The pseudo code programming example provides the essentials for accessing an asynchronous auto dial modem from an HP3000 terminal port. The case history example disproves the fact that "oh, any compatible equipment will work". Good luck!

References

1. Bell 801C-LI/2 Data Auxiliary Set Interface Specification, November 1976, Bell System Data Communications Technical Reference.

2. System Operation and Resource Management Reference Manual, Update #1, January 1985, Hewlett-Packard Company.

3. Data Communications Handbook, Third Edition, April 1981, Hewlett-Packard Company.

4. Penril 8801C Automatic Calling Unit Operations Manual, Publication 8801C-OM-0381, Penril.

5. Penril Data Modem 8208A/B Instruction Manual, 60A047A01-01, Rev A, Penril.