HP-IL Update

The Hewlett-Packard Interface Loop (HP-IL) was designed to fill the need for a low-power medium speed interface for battery powered devices.

The only existing standard that came close, was RS-232C. This interface was not adequate for a portable computer operation in several areas. It took too much power at the upper limits of the specification, more than the portable computer would use itself, in many cases. It was not fast enough for reasonable portable mass memory operations. Only two devices could normally be connected at a time, and were limited to 20 feet apart by specification and 100 feet in accepted practice. The connector and cable was bulky, and there were electrical *and* physical sexes, as any of you who have interconnected RS-232C devices know!

I am not familiar with the earliest decisions at Hewlett-Packard about HP-IL, but after some study, they decided on a low-power two wire balanced line ring or loop network based on HP-IB, HP's high speed, high power parallel interface (also known as GP-IB or IEEE-488). It was decided, however, not to submit the new interface to the IEEE or EIA for standardization, as a result of the problems Hewlett-Packard encountered having HP-IB standardized. Hewlett-Packard hopes, instead, that it will become a *defacto* standard. This will probably work, but might take more time. Who knows, once a standards committee got it, it could have taken years, and not even resemble what was submitted to them when they finish. As a side note, HP-IL was renamed 'PIL' for Personal Interface Loop about six months before public announcement. The name 'PIL' lasted only a short time, however, as the 'non-personal' divisions of Hewlett-Packard refused to support a *personal* interface. Also, at least one Corvallis Division woman working on HP-IL products was getting tired of 'on the Pil' jokes.

HP-IL, as it emerged, meets almost all of the needs of portable computer interfacing. It is implemented with low-power CMOS devices. The interconnection is with small, lightweight wires; although two cables are required at minimum. 961 devices can be supported at one time with up to 100 meters of cable between each device. Each data byte sent out on the loop is return to the sender by the receiving device(s) for verification. Data transfer rates of up to 20,000 bytes per second (200,000 baud in RS-232 comparison) are possible; although in normal use this is not practical, as each device 'echoes' the data, slowing it's trip around the loop. A full compliment of device control functions are included.

As we all know, the first HP-IL controller was the HP-41C, and I use the term 'controller' loosely in the case of the HP-41. This, along with the cassette drive and strip printer, while quite functional, was a very poor first showing for HP-IL. The HP-41C controller implementation was buggy and very limited in scope. Neither the printer nor the cassette responded to the ASCII ID command, making them harder to find on later, more advanced loops. None of the devices supported extended addressing, nor did the HP-41. To a large extent, this was a classic 'Chicken and Egg' situation. Do you build a full function controller first, when there are no devices to take advantage of the features, or would you build a full function device when there are no controllers to use or even fully test it. With the HP-41 as the only (portable) controller, the first devices were uninspired. The only exception to this was the 82166A Interface Module. This Module implemented almost the complete set

of capabilities of HP-IL. This certainly caused it's short product life, as the HP-41 was unable to deal with it on any but the simplest levels. Not until the late arrival of the EIO module could even a small part of it's capabilities be explored, and by then, it was too late. The 82166A even had (un-documented) the ability to be Loop and System controller. These features were not discussed, as they further complicated things. The lead marked VC1 was actually the system controller lead. A 'zero' level on first power-on made the 82166A the system controller. As the HP-41 is unable to give-up loop control, this would have only caused trouble.

Another interesting early device is the HP-3468A Multi-Meter. This is the only loop device or controller I know of that uses only two transformers to interface to the loop, instead of the normal three. Also, the meter division, which was the driving force behind the loop power up/down capability, then went on to produce a battery powered meter that will not loop power down.

Fortunately, most of the more recent HP-IL devices are much improved, although we now seem to have lost Loop Power Down almost completely. The HP-71 is by far the best controller for the loop Hewlett-Packard has produced to date. Anyone wanting to evaluate HP-IL must use the HP-71 before judging the loop. It is capable of transfer rates of over 6000 bytes per second (60,000 baud, in RS-232C comparison). It gracefully gives up loop control, and can exist in an active loop while powered down, a unique feature to the HP-71. The Disk Drive and ThinkJet both are better 'citizens' than their predecessors, with the exception of no Loop Power Down. Unfortunately, now that we have a controller (HP-71) that handles the loop power commands correctly; Hewlett-Packard seems to have abandoned that feature. The only 'throw-back' in terms of HP-IL is the HP-110. While it uses the loop well for it's own operating system purposes, it is probably weaker than the HP-41 in general and instrument control operations. The HP-110 also will not give up loop control under it's normal operation. One interesting recent HP-IL product is the HP82973 interface card for the IBM-PC™. While sold primarily as a link between the HP-110 and the IBM-PC; when installed in the IBM-PC, it gives the PC HP-IL at the same level as the HP-110. While limited in scope at present, this card opens many doors to the PC. At our location in Richmond, we routinely use HP-IL to transfer files between two of our IBM-PC ATs™ on different floors at 6000-7000 bytes per second. While not true networking, one machine can access all the drives on the other, when it runs special device software. While currently using the same driver code on the IBM-PC as is used in Nomad; improved software could give this interface any level of HP-IL implementation desired. A note of caution to any of you owning HP-110s and IBM-PCs: Do *not* use this interface to access the drives on any IBM-PC running MS or PC DOS™ 3.0 or higher with your HP-110. The 2.x DOS in your HP-110 does not understand the new FAT table used under DOS 3.0 for hard disks, and might corrupt your entire hard disk!. Both of our 'ATs run 3.0 and so understand each other.

The future of HP-IL is looking bright. Hewlett-Packard is actively pursuing second sources for the key parts. Already, the transformer assembly is available from Pulse Engineering in San Diego. TRW is considering making the connectors, and National Semiconductor is in the final testing stage of their HP-IL interface I.C.

When ready, these second sources will allow competitive companies to use HP-IL in their products without the fear of a single source solution. This, I believe, will bring about the *defacto* standard Hewlett-Packard wanted.

For your reference and amusement, I have included my first column on HP-IL, *The Data Frame*, as it may no longer be available from the original publisher.

THE DATA FRAME

As an overview, an HP-IL system typically consists of two or more devices connected together in a two wire loop, with one device as controller. The loop arrangement allows for error-checking, as the originator of a message generally gets it back from the other end of the loop, and can check it to see that it matches what it sent. Except in the simplest of two device systems, one device must be in control, that is, it is the device that allows orderly transfer of data between devices according to the need of those devices, by designating one device to send data, and one or more devices to receive it at a time. By putting the burden of control only on designated devices, other devices need be only as intelligent or simple as needed for their specific use. As an example, the 82162A printer requires less internal intelligence to be a printer, than does the 82161A cassette drive, but neither device can, or needs to, control the loop. The controller device communicates with the individual devices by first giving all devices an address. This is done, basically, by the controller sending a message around the loop that contains an address number. The first device to receive that message takes the address for its own, adds one to the address number in the message, and then sends it on around the loop to the next device, which repeats the operation until the message returns to the controller. The controller then knows how many devices are on its loop and can then talk to them one at a time by their address, with all other devices ignoring and just echoing any message not for their address. As you are beginning to see, the loop operation is not as simple as it first appears to be when using the HP-41 HP-IL interface. This is because almost all of the complex loop communication is done by the HP-41 as controller with specific devices on the loop; while you, the user, only request loop functions with simple commands. This works for now, but the beauty of the loop is in its flexibility in working with different devices, and the 'friendly' functions of the HP-41 only work with specific devices. Any new or user-devised devices will require more knowledge of the operation of the loop, and will require, in most cases, loop level commands such as the few in the HP-IL module now, and those in the forthcoming Extended I/O Rom.

The HP-IL interface, as you know by now, is a bit serial interface. That is, information is sent send bit by bit through the loop. The basic unit of information is a packet of eleven bits called a 'Frame'. All communication on the Loop is sent in the form of one or more eleven bit frames. The first bit in a frame has a unique pattern to identify the start of a frame. Frames move asynchronously around the loop, and even bits in a frame can be asynchronous.

There are four basic types of frames, Command frames, Data frames, Ready frames, and Identify frames. The frame class can be determined by the first three bits of the frame as shown below:

000 ---------- Data
001 ---------- Data with service request bit
010 ---------- Data end
011 ---------- Data end with service request
100 ---------- Command
101 ---------- Ready

110 ---------- Identify
111 ---------- identify with service request

The remaining eight bits in the frames are for data, in the case of the identify and data classes; and for specific commands, in the remaining classes.

As you can see, there are several sub-types in two classes, Data and Identify. In both classes, there is a bit for service request (more later), and in the Data class, there is a bit to flag the end frame.

Let the frames set for awhile, let's look at devices on the loop. There are four possible states an active device can have; System controller, Active controller, Active Talker, and Active Listener. A loop can have only one System controller. The System controller is the active controller on loop power-up, and can then pass active control to another device capable of running the loop, but the System controller, and only the System controller, can demand loop control be returned to it via a privileged command only it can issue. The Active controller is the device in active control of the loop. It is the device that assigns talkers, and listeners; and allows and controls data flow around the loop. The Active controller may also pass control to another device capable of being a controller. An Active Talker is a device that has been assigned as the device to source data frames in a data transfer. There can be only one active talker at a time on the loop. Active Listeners are the devices assigned to receive data frames. There can be any number of active listeners on the loop at a time, with the exception that the active talker may not be an active listener at the same time. The Active Controller may also be the Active Talker or an Active Listener

The Active Controller is the only device that can source (send) the Command class of frame. This class is used to assign talkers and listeners; and to do general device and loop control. Command frames are received and immediately retransmitted to speed up response to commands as commands usually require action by more than one device. Ready frames can also usually be sent only by the active controller but are of a type that require processing by a device before re-transmission. These are used for device addressing and starting and stopping data frame transmission around the loop. There is also one ready frame used to follow a command frame to let the active controller know when all devices have finished processing a command, as it will receive its command frame back before all devices have completed processing it. Identify frames, in normal operation, can also only be sourced by the active controller. This frame is used to help identify a device requesting service. As seen above, this frame, and the data frame, have a bit reserved for service request. Any device on the loop may set this bit on any data or identify frame to indicate to the controller that a device needs attention. The identify frame will locate the device in systems where a form of interrupt identification Known as 'Parallel poll' has been set up by the controller to allow specific devices to be located quickly if they need service. Where parallel poll is not implemented (in the case of the 41C) the identify frame still allows a device to signal for help when there is no data transfer in progress, but the controller must 'ask' each device in turn if it is the one that requested service. Data frames can only be sourced by the Active Talker. There is no difference, in the loop operation, between Data frames, and Data end frames; but specific Active listeners may respond differently to the two types. A talker signals

to the controller that it has completed its data transmission using a special pair of ready frames to signal completion with or without errors.

This completes the nickel tour of the HP-IL. I will go into much more detail in all of the above areas in future columns. This is where your input will greatly help in pointing out areas of particular confusion in this very complex system.

I will end this month with an explanation of the different number of devices you keep hearing the loop will support.

There are two modes of device addressing on the loop, simple addressing, and extended addressing. The addressing field in the command frames allows for addresses from 0 to 30. In simple addressing, there can, therefore, be 31 devices with addresses on the loop. As the controller can assign itself without an address, 32 devices total can exist on a loop with simple addressing. In the HP-41 HP-IL implementation, however, the address of 0 is reserved (for the 41C?) and unavailable, hence the 30 devices with the HP-41C as the 31st. In extended addressing, each device has a primary address from 0 to 30 and a secondary address from 0 to 30, allowing for 961 addressed devices, plus the controller; which need not be addressed; for a total of 962 devices on a loop. None of the currently available devices implement extended addressing, however, so the current limit is 31 devices.

Jim De Arras
Hand Held Products, Inc.
Research and Development Center
3928 MARGATE DRIVE
RICHMOND, VIRGINIA 23235
(804) 272-9285
TWX (910) 380-9256 HANDHELD PROD