HIGH SPEED DIGITAL DATA INPUTS FOR THERMAL ARRAY CHART RECORDERS

David M. Gaskill
Vice president, Research & Development
Astro-Med Inc.
West Warwick, RI USA

KEYWORDS

Digital data, High speed network, Thermal array recorder

ABSTRACT

Many telemetry stations would like to convert from using digital-to analog converters (DAC's) to using direct digital inputs to their chart recorders but can't find a suitable recorder interface. These stations often have hundreds or even thousands of channels of information being bussed around at very high speeds on propriatary real-time computer systems. The lack of standardization has naturally presented recorder manufacturers with problems in selecting the appropriate interface hardware. Standard parallel interfaces, such as SCSI and GPIB, are usually too slow and not really suited for real-time transfer, although they can be used in some circumstances which will be described. The best choice seems to be a general purpose parallel port of at least 16 data bits which can support a large number of addresses. Such an interface can be used with a high speed network like SCRAMNet as well as with a general purpose computer or workstation. This paper will describe several available parallel ports using both TTL and RS-485 (long-line) hardware and some practical implementations of thermal array recorder use with SCRAMNet , GPIB, and general purpose parallel busses.

INTRODUCTION

Ever since thermal array recorders became generally available five or six years ago, it seemed obvious that users would soon take advantage of the fact that DAC's were no longer required and that digital data could be input to the recorders directly. In practice, however, this did not happen except in a very small number of installations. There were, and still are, several reasons for the persistence of analog data transfer. First, many stations were using thermal array recorders simply as replacements for pen recorders and already had DAC's in place. Many of these users felt that the benefits of direct digital transfer were not worth the expense of converting their installations, unless, perhaps, as part of a general station upgrade - not a frequent event. Second, even users with the desire and the budget to convert to digital transfer were having trouble with the actual hardware interface. We have found that there is no universal solution but that most computer systems can be accommodated with just a few parallel port options.

DISCUSSION

The first thing to consider is the suitability of an interface to system requirements. Speed and adressability are usually the primary factors with ease of implementation secondary. The system capability should be considered as a package taking into account the capabilities and limitations of the recorder as well as the nominal performance characteristics of the specific interface. For example, the data requirements of the recorder format may not fit the most efficient operating mode of the interface so that extra commands are required or that extra data bytes must be transmitted. Such practical considerations may reduce the achievable bandwidth to a fraction of the theoretical maximum. As far as adressability goes, the issue is straightforward; each interface has fixed, easily identified capabilities.

GPIB (General Purpose Instrument Bus) is a very popular and well known interface. It has been around for more than twenty years and is generally considered to be very stable and easy to use. Unfortunately, it is not always the best choice for telemetry applications because the real-time data rates are relatively low and the adressability is limited to 15 recorders per controller. However, because GPIB is inexpensive and easy to implement, it still has a lot of appeal and can be successful in certain situations. Most of GPIB's limitations are a result of its original design purpose. It was meant to control a series of relatively simple instruments like volt meters, power supplies, and frequency generators. It was meant to send set-up commands and occasionally read data but not necessarily in real time. When used with a modern thermal array chart recorder, a GPIB system should be capable of handling a sixteen channel real time rate of about 750 Hz per channel. While this may seem low to experienced GPIB users, each data point requires two bytes so the total data rate is over 24 KHz per recorder. This is on the borderline of "high speed" but the per channel rate is adequate for most telemetry purposes and when used with a distributed controller system, the aggregate data rate can be quite high.

SCSI (Small Computer System Interface) is also well known because of its popularity in workstations and high speed disk drives. Like GPIB, SCSI promises a standardized interface approach and has been very successful in personal computers of all types.

For real-time recording, however, there are problems. First, real-time data rates are limited. Like GPIB, SCSI is an eight bit interface so that each data point requires two bytes. In addition, each string of data points requires a six byte command descriptor block (CDB). In its original application, streaming large blocks of data between computers and storage devices, this is not burdensome but in a chart recorder this CDB overhead may represent one fifth of the total transmission. Even so, it's a little faster than GPIB and should be capable of over 1200 Hz per channel in sixteen channel mode. The second limitation is that only seven SCSI devices can be supported by each controller. The third, and perhaps the most important, potential pitfall is the lack of software drivers for real-time recorders. This is really a user perception problem rather than a design problem but nevertheless it has stopped many implementations. The only standard "plug and play" devices for SCSI are disk drive, tapes, and related peripherals like printers. Chart recorders fall into the "processor" class of SCSI instrunents which is not as tightly controlled as other classes because of the extreme variability possible in this class. Consequently, the user must write whatever drivers are needed, often with great difficulty compared to similar GPIB drivers. We have found that successful SCSI implementations must be done by experienced systems integrators; it is too difficult for most end-users.

True high speed recording requires a parallel port wide enough to hold all the data for one channel and its address in a single word. In practice, this means a propriatory parallel port although, if designed carefully, such a port can be very simple and easy to use. This sort of general purpose parallel port has three advantages over eight bit standard ports. First, each channel needs only one word. Second, no additional address or command bytes are required. Third, the overall bandwidth is much higher, up to 500 KHz. Fewer words at a higher rate mean big differences at the individual channel level. A 24 bit parallel port can support a 60KHz per channel data rate in an eight channel system. Individual channels can be updated at 200 Khz. The disadvantage is that the port does not follow an industry recognized standard and so requires custom hardware. On the other hand, it is similar in concept to other 24 to 32 bit wide ports already in use in the telemetry community and can be readily adapted to existing systems with a relatively simple interface board. Also, this port can support either TTL or RS-485 interfaces. When using RS-485, also known as "long-line", the recorder can be operated at considerable distances from the host computer system; at least 400 feet and as much as 1000 feet at lower data rates.

The eight tag bits support up to 256 channel addresses in each recorder. The channels are assigned either locally through the front panel or remotely through the host control port. Each recorder can have up to 32 different channel addresses although most telemetrists prefer to limit the chart to eight channels for readability. Therefore, 32 eight channel recorders can be supported on one bus.

A diagram of the parallel port data/command structure:

DATA	TAGS	Data Ready Device Ready
24 23 22 21 20 19 18 17 16 15 14 13 12 11	10 9 8 7 6 5	4 3 2 1
		ATTN Reset

Figure 1.

A comparison of an eight channel data stream using the three interfaces described above:

GPIB	SCSI	Parallel
Command	Operation Code	Ch. 1 data
Address	Logical Unit No.	Ch. 2 data
Ch. 1 MSB	Logical Address	Ch. 3 data
Ch. 1 LSB	Logical Add. LSB	Ch. 4 data
Ch. 2 MSB	Length	Ch. 5 data
Ch. 2 LSB	Control	Ch. 6 data
Ch. 3 MSB	Ch. l MSB	Ch. 7 data
Ch. 3 LSB	Ch. 1 LSB	Ch. 8 data
Ch. 4 MSB	Ch. 2 MSB	
Ch. 4 LSB	Ch. 2 LSB	
Ch. 5 MSB	Ch. 3 MSB	
Ch. 5 LSB	Ch. 3 LSB	
Ch. 6 MSB	Ch. 4 MSB	
Ch. 6 LSB	Ch. 4 LSB	
Ch. 7 MSB	Ch. 5 MSB	
Ch. 7 LSB	Ch. 5 LSB	
Ch. 8 MSB	Ch. 6 MSB	
Ch. 8 LSB	Ch. 6 LSB	
	Ch. 7 MSB	
	Ch. 7 LSB	
	Ch. 8 MSB	
	Ch. 8 LSB	
	Status	

Just a quick look at the relative lengths of the lists gives a good idea of the potential speed advantage of the parallel port.

IMPLEMENTATION

The first implementation is a good example of a medium speed GPIB system. The hardware is capable of faster operation but the user only required a data rate of about 7200 samples per second total. It is designed to be a real-time controller and monitor for an integrated computer system that will support multiple space program needs. The end user wanted a system with the flexibility to support continued development so that the overall installation could remain current for 10 to 30 years. Each of the electronics system consists of a VME bus based UNIX host computer and six strip chart recorders. Each of the recorders supports up to 12 channels of waveforms at 100Hz per channel and also provides alphanumeric printing for channel identification and report generation. In addition, the recorders can provide single channel plotting at 1000Hz for high resolution analysis. The size and location of each channel can be selected independently to give the end user the ability to tailor the chart presentation to special requirements if necessary.

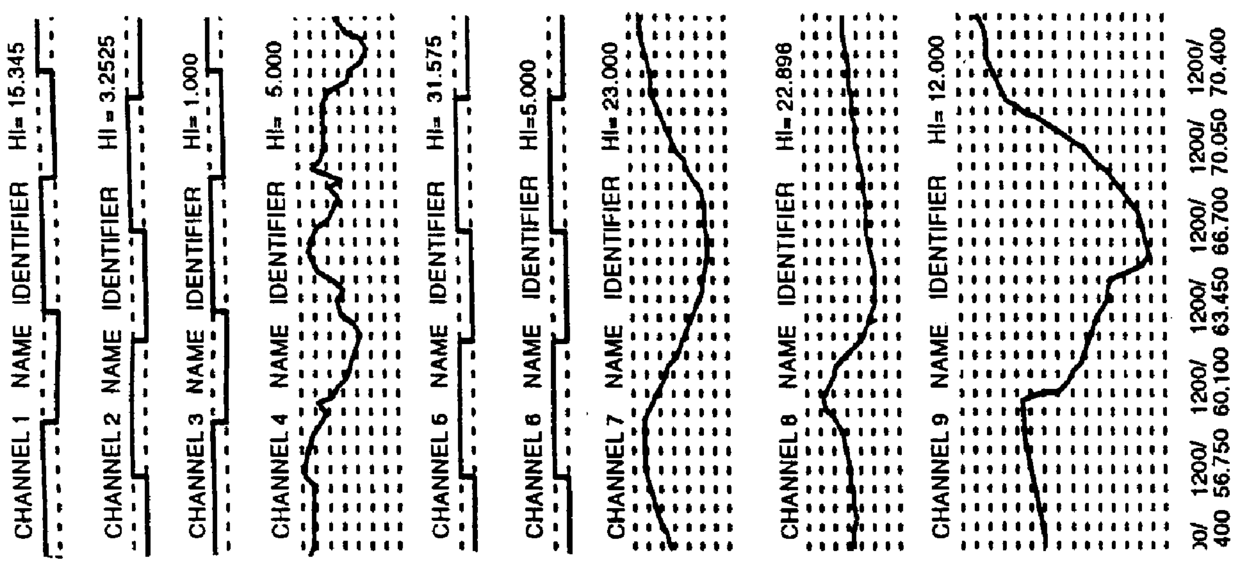


Figure 2. Sample Chart Format

The host communicates with each recorder using two GPIB ports, one for command and one for data. While this is not necessary at current data rates, it does simplify system control software and allows future speed increases. Also, it allows the use of standard plug-in data interface cards, an important consideration because the end user wanted a Commercial off-the-shelf (COTS) product to avoid the development pains of a custom design. The full installation is planned to include 20 six channel systems phased in over a four year period giving a total of 1440 channels and an aggregate bandwidth of 144KHz.

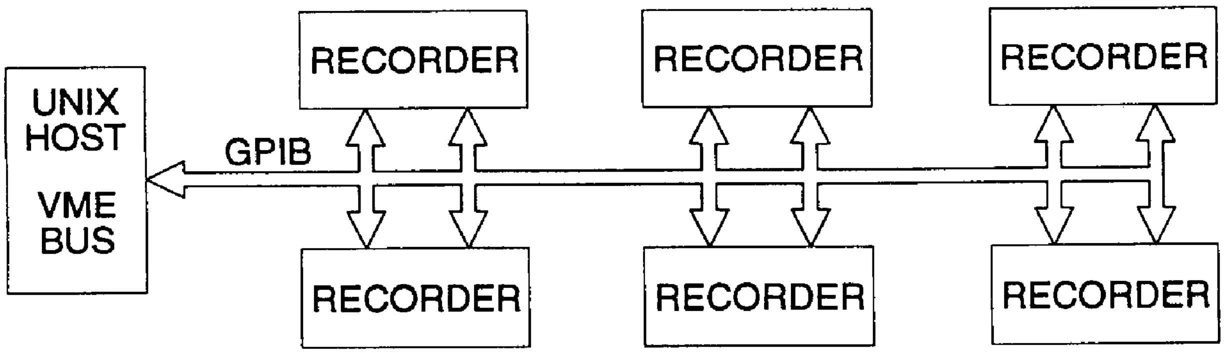


Figure 3. Chart recording system

The second implementation is a good example of how to convert a traditional DAC based system to a completely digital system without changing the basic architecture. The user is a commercial aircraft manufacturer and the application is in-flight testing. In this installation, data is acquired by a data acquisition sub system where it is digitized and recorded in a PCM format. The serial PCM stream is converted to parallel and scaled to appropriate engineering units. Data that is to be displayed on a chart recorder is further converted from floating point to integer format. The integer format data is then broadcast for recording. In the original system, this was routed to a Processor section which routed selected tags to a series of DAC's which fed analog signals to the chart recorders. The intent of the conversion was to minimize overall system impact while eliminating the DAC section. Also, it was considered desirable to maintain the capability of supporting standard analog input modules if required for special tests. In fact, this implementation alows digital and analog signals to be mixed freely and gives the local operator full control of all the normal chart recorders features.

The recorders will take data from an existing SCRAMNet bus through a fiber optics interface board. A set of address range switches are used to select a block of tags that the recorder will recognize. The comparator generates a strobe when the data is valid and the address is in the selected range. The fiber optics interface board then puts out a 12 bit word to the parallel port in the chart recorder. The input section of the parallel interface contains a dual port RAM which holds the data until the recorder is ready to print. This allows the recorder to operate completely asynchronously from the main data bus.

Finally , here is a detailed look at how to interface to our DWP-1 parallel interface board using inexpensive and easy to obtain components. This interface is usable in its own right with a data bandwidth of up to 20 Ksamples per second but for most

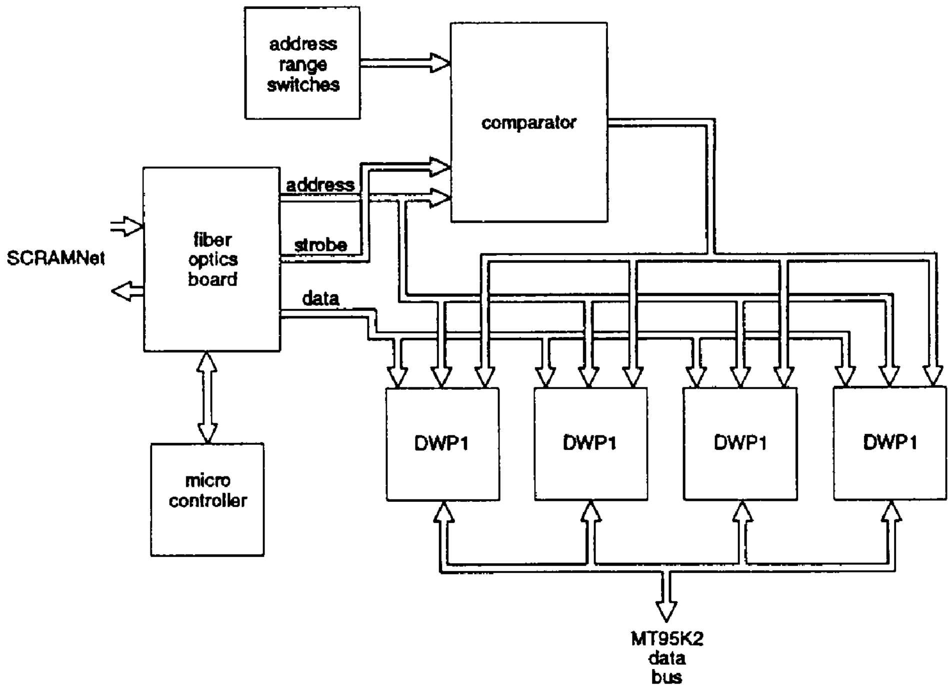


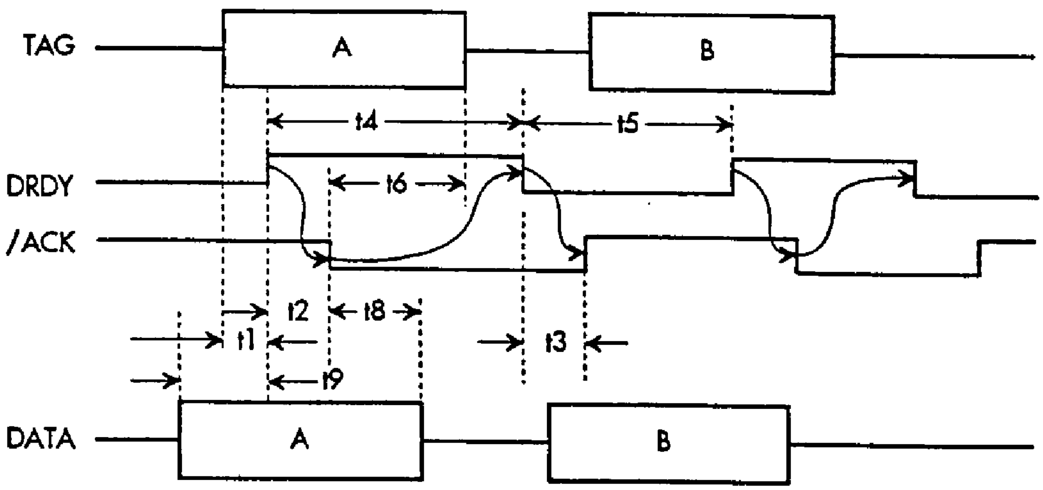
Figure 4. SCRAMNet to DWP-l Parallel interface

telemetry applications its value is more as an example of how to get started with a system integration plan without committing significant resources. The components required are a standard PC type computer running DOS and a National Instruments AT-DIO-32F parallel interface board. The DWP-1 is designed for a direct pin to pin connection with the AT-DIO-32F if the proper connectors are used. We recommend AMP "CHAMP" (#554085-1) on the DWP-1 and AMP Special Industries #2-746483-2 on the AT-DIO-32F. These connectors compensate for the apparent disparity between the two boards.

The input sequence looks like this:

1. Host sends data and tag.
2. Host raises DRDY to indicate that data and tag are stable.
3. DWP-l responds by lowering /ACK which indicates that host can send new data and tag.
4. Host lowers DRDY and sends new data and tag.
5. DWP-l raises/ACK.
6. Repeat steps 2 through 5.

The tag assignment is done locally through the front panel display and keyboard. Each channel must have a unique address although there is no other limit as to sequence. The default settings are channel 1 to n (8-32) in numerical order. A long-line version of this board is also available for applications requiring connections over about 15 feet.



t1	TAG valid to DRDY hi	100	--
t2	DRDY hi to /ACK lo	--	540
t3	DRDY lo to /ACK hi	80	(See Note 3)
t4	DRDY hi (See Note 1)	520	--
t5	DRDY lo (See Note 1)	40	--
t6	TAG hold after /ACK lo	0	--
t8	DATA hold after /ACK lo	0	--
t9	DATA valid to DRDY hi	40	--

All timing values are in nS.

Note 1: t4 + t5 > 2 µS
Note 2: If the National AT-DIO-32F board is used as host, the handshaking is LEVEL MODE.
Note 3: Cable length, loading, and pull-up resistor value affects t3 max. On board, pull-up is 100. Additional pull-ups added in parallel must not reduce total resistance to less than 10 .

CONCLUSION

While there is still little standardization in using digital data interfaces with thermal array chart recorders, this not necessarily mean that it must be difficult. Often 8 bit interfaces like GPIB and SCSI can be used effectively if data rates are not too high and if multiple addresses can be presorted. When higher rates are needed, there are straight forward parallel ports available even though some custom interfacing hardware is usually required. Often the implementation cost of minor customizing can be recaptured on only one or two recorders since one digital board can replace several expensive analog input boards. Therefore even stations that already have their DACs installed and paid for should be able to consider digital interfaces for any additional recorders.

Figure 5.

DWP1 Signals				National AT-DIO-32F Signals			
DESCRIPTION	*NAME*		*DWP1 PIN #*	*DESCRIPTION*	*NAME*		*AT32 PIN #*
Chassis GND	Shield	*	1	Port D	DIOD1	*	1
	unused	@	2	Port D	DIOD3		3
	reserved	%	3	Port D	DIOD6		5
	unused	@	4	Port D	DIOD2		7
Tag Data	TAG5		5	Port C	DIOC5		9
Tag Data	TAG3		6	Port C	DIOC3		11
Tag Data	TAG2		7	Port C	DIOC2		13
Tag Data	TAG6		8	Port C	DIOC6		15
	SIG COMMON		9		SIG COMMON		17
	SIG COMMON		10		SIG COMMON		19
	SIG COMMON		11		SIG COMMON		21
	SIG COMMON		12		SIG COMMON		23
	SIG COMMON		13		SIG COMMON		25
Data/Tag Ready *	DRDY		14	ack 1 (output)	ACK1		27
Device Ready	DEVRDY (OC)	$	15	extra input 1	IN1		29
+5 or NC=input	IO		16	extra output 1	OUT1		31
Data Acknowledge *	/ACK (OC)	$	17	request 1 (input)	REQ1		33
Waveform Data	DATA4		18	Port A	DIOA4		35
Waveform Data	DATA0		19	Port A	DIOA0		37
Waveform Data	DATA1		20	Port A	DIOA1		39
Waveform Data	DATA7		21	Port A	DIOA7		41
	unused	@	22	Port B	DIOB5		43
	unused	@	23	Port B	DIOB7		45
Waveform Data	DATA8		24	Port B	DIOB0		47
	unused	@	25	Port B	DIOB4		49
	reserved	%	26	Port D	DIOD4		2
	reserved	%	27	Port D	DIOD0		4
	reserved	%	28	Port D	DIOD7		6
	reserved	%	29	Port D	DIOD5		8
Tag Data	TAG7		30	Port C	DIOC7		10
Tag Data	TAG1		31	Port C	DIOC1		12
Tag Data	TAG0		32	Port C	DIOC0		14
Tag Data	TAG4		33	Port C	DIOC4		16
Tag Ready *	TAGRDY		34	ack 2 (output)	ACK2		18
Output Enable	OE		35	extra input 2	IN2		20
	unused	@	36	extra output 2	OUT2		22
Tag Acknowledge *	/TACK (OC)	$	37	request 2 (input)	REQ2		24
	SIG COMMON		38		SIG COMMON		26
	SIG COMMON		39		SIG COMMON		28
	SIG COMMON		40		SIG COMMON		30
	SIG COMMON		41		SIG COMMON		32
	SIG COMMON		42		SIG COMMON		34
Waveform Data	DATA6		43	Port A	DIOA6		36
Waveform Data	DATA2		44	Port A	DIOA2		38
Waveform Data	DATA3		45	Port A	DIOA3		40
Waveform Data	DATA5		46	Port A	DIOA5		42
Waveform Data	DATA10		47	Port B	DIOB1		44
	unused	@	48	Port B	DIOB6		46
Waveform Data	DATA11		49	Port B	DIOB3		48
Waveform Data	DATA9		50	Port B	DIOB1		50
Waveform Data	DATA7		21	Port A	DIOA7		41

SCRAMNet is a registered trademark of Systran Corporation