# THE IMPACT OF THE IEEE-488 CONCEPT ON MEASUREMENT PRODUCTIVITY

**George C. Stanley**
**Hewlett-Packard Company**
**1819 Page Mill Road**
**Palo Alto, CA 94304**

## ABSTRACT

In the mid 1970's three events came together that have had a major impact on measurement productivity. The most significant of these was the IEEE-488 Interface Bus. The second was the "friendly", interpretive-language desktop computer and the third was the intelligent or "smart" instrument. Since the combined impact of these three has been enormous, their-development and relationships will be briefly traced. Next covered will be the technical characteristics of the interface bus with emphasis on what sets the technical limits of speed, distance, and number of products connected. Knowing this leads to a discussion of how to go beyond these limits if required. Finally, this paper examines the application of these concepts using two aerospace examples and describes the resultant productivity improvements.

## INTRODUCTION

Since its introduction in 1975 IEEE-488 has been a tremendous success. In fact, it is the most popular standard IEEE has ever published with more than 30,000 copies having been distributed. This same concept also appears as IEC 625-1. IEC stands for International Electrotechnical Commission and is a world-wide standards group based in Geneva, Switzerland. In addition it has been published as ANSI MC 1.1 and under such names as HP-IB (Hewlett-Packard Interface Bus), GPIB (General Purpose Interface Bus) and Plus Bus (a name used by Solartron) . The only difference between all of these is that IEC 625-1 uses the modem 25 pin connector. All the others use a special 24 pin connector.

In order to understand why it has been so succesful it is helpful to go back and examine what one typically had to do to put a system together prior to IEEE-488.

## HISTORY AND EVOLUTION

The idea that eventually became IEEE-488 started in 1965 at the Hewlett-Packard Co. when they began to look at ways to standardize the interfacing of all future Hewlett-Packard products. This was done because at that time every interfacing project had to be considered on an individual basis. For example, a person putting together a measurement system had to find the instruments that would make the specific measurements and then hope they had some interfacing capability. Usually they did not. At this point in time, interfacing was not considered primarily important, and if it was considered at all, it was either left until the end of the project or possibly given to the junior engineer to do. There were two reasons for this attitude: One was that computers were generally perceived as complex devices that measurement engineers often didn’t understand. The other reason was that almost every engineer had a different ideas as to how an interfacing standard should be created. As an example of what one faced there was one product, a computing counter, which had 163 unique lines that had to be specially interfaced. Obviously, in situations like this the cost and complexity for each instrument become severe. In short, there was no common interfacing arrangement, i.e., no common connector, no standard signal or logic levels and no common cabling. As a result, it was not only difficult but quite expensive to put a system together.

## PROBLEMS IEEE-488 SOLVED

Expanding on the problems of interfacing prior to IEEE-488 it is easy to see that individual instruments’ complexity led to a great many unique solutions and substantial costs. At that time it was estimated for planning purposes that software costs would be *at least* 2x the hardware cost and this was a relatively conservative number. Because of unexpected problems, actual software costs were often 5x or more the hardware costs. Of course these estimates are very speculative but after IEEE-488, software costs were often below or equal to .5x the hardware costs. That’s a savings of at least 4:1. On top of that, system hardware costs have been coming down because all of the unique fixturing, cabling and interface processing have been eliminated. For example, prior to IEEE-488 a unique hardware interface and software driver had to be provided for *each* system instrument. In many cases two interfaces for each instrument were needed if two-way communication was desired.

Two other factors arrived in the mid 1970’s which tremendously helped IEEE-488 succeed. These were introduction of the “friendly” desktop computer/controller and the “intelligent” or “smart” instrument. The “friendly” computer played a particularly significant role because prior to its introduction the measurement engineer either had to be computer knowledgeable or work with a computer specialist. This almost always added time and cost to the system development cycle. The “friendly” desktop computer helped immensely as it used an interpretive language instead of a compiled language. An interpretive language process is one where each line of code is checked for errors at the time that line is stored. Then if an error was made, the computer has the ability to spot the place in the line where the error was made, describe the error and suggest a possible correction. Even though this process is slightly slower than the compiled computer process, where the program is all compiled at one time, it simplifies the writing of computer programs by non-computer specialists and test engineers. This is a highly-efficient process and usually much less costly. The other factor which helped was the “intelligent” or “smart” instrument which allowed the instrument to do by itself many things that previously had to be done step-by-step by the computer.

## IMPACT ON PRODUCTIVITY

The combination of the interface bus, the “friendly” desktop controller and the “smart” instrument have gone a long way in helping make better measurements in less time. This is the essence of productivity which is so important in today’s world. For example, the present electronics market is growing at 12.5%/year but engineering graduates are projected to stay relatively constant for the next five years. Closely coupled here is the prime rate which has been fluctuating between 10 and 20%. This fact makes spending capital equipment money much more closely watched in terms of return on investment (ROI) and productivity improvements.

## STATUS OF IEEE-488 TODAY

On a world-wide basis there are approximately 1200 products today (7/81) that conform to the IEEE-488 bus and it is projected that by the end of 1982 there will be over 2000 products available. The present makeup is 56% measurement instruments, 11% computer, 12% systems, 8% peripherals and 13% other. The IEEE-488 standard was revised slightly in 1978. Most of the changes were in the area of editorial classifications and minor technical changes. No major area was changed in concept.

There is continuing work on IEEE-488 in the area of Code and Format Conventions and this companion document is expected in late 1981 or early 1982. The reason for this extra document is that the present standard defines the mechanical characteristics (the connector), the electrical characteristics (the signal levels and logic conventions) and the functional characteristics (the relationship between the control lines and data lines). The operational characteristics (the way data is formatted) was purposely left undefined in the original IEEE-488 document. The reason was to not restrict the design engineer. Experience has shown that some definitions and recommended guidelines are needed in the area of machine-to-machine information interchanges in order to promote comparability between different manufacturers’ products.

## TECHNICAL CONSIDERATIONS

IEEE-488 defines a bus structure that has three basic parts. There is the data bus, which consists of 8 data lines (DIO 1-8), and the Control bus, which consists of 5 control lines (IFC, ATN, REN, SRQ, and EOI). Finally, there is the handshake bus consisting of 3 handshake lines (DAV, NRFD, and NDAC). Summarizing the interface technical specifications:

No. of devices - 15 max (incl. controller)

Signal lines - 8 data and 8 control

Data rate - Up to 1 megabyte/sec

Data transfer - Byte serial, bit parallel, bidirectional, interlocked handshake

Transmission path - 2 metres x the number of instruments up to a maximum of 20 metres

A few comments on these specifications are in order. The limit of 15 was selected using three considerations. The first was that discrete component drivers were desired with drivers and loads located in each instrument. This would allow complete flexibility as to cabling. At the time the standard was prepared, discrete low-cost drivers had a current-drive capability that would handle 15 instruments in parallel but not much more. The second consideration was whether or not a 15 instrument limit was realistic. Studying many existing systems indicated that 15 was a very practical upper limit as typically that was a three-bay system. The last consideration was how difficult it would be to go beyond the 15 limit if required. Since this only entails the addition of one more interface card and virtually no programming changes, it was felt that this was a very workable limit.

The data rate of up to 1 megabyte/sec is only achieved if the cabling is restricted to 1 metre/instrument and tri-state drivers are used. Since virtually no instruments operate at this speed, this interface limit on data transfer has rarely been a problem. Much more important to the user is the number of readings /sec that an instrument can take and/or send over the bus. For example a 4 digit voltmeter reading may require twelve or thirteen bytes of data to send one reading. The reason is that the reading is probably sent in floating point notation as shown:

N + 1.234 + 02 CR LF

The first byte may be used to indicate if the reading is normal (N) or overload (0) so the computer can branch appropriately. A typical DC voltmeter takes 4.5 readings/sec. Then the bus structure (interface plus computer) would only have to handle 50 bytes/sec which is a far cry from the 1 megabyte/sec limit. A high-speed DC voltmeter taking 3,000 readings/sec would need 40,000 bytes/sec which is still quite far from the 1 megabyte/sec limit.

Data structure is byte-serial, bit-parallel and is shown in Figure 1. The important points are that data is bidirectional, meaning it can flow either way on the bus, and that a handshake occurs after each data byte.

Transmission path length has a double restriction. It is either 2 metres x the number of instruments or 20 metres total, *whichever is less*. To understand why there are two restrictions, keep in mind there are drivers and terminations inside each product. Therefore, when there are few products involved, the termination impedance is relatively high and there is the possibility of standing waves (peaks and nulls) on the cables. As more instruments are added, the line termination impedance drops and more current flows into the lines. Eventually the reduced termination resistance plus the combination of signal-to-noise plus the cable resistive losses become dominant.

If necessary, the distance limitation can be overcome by using distance extenders as is shown in Figure 2. Here, speed is traded for distance as the byte-serial, bit-parallel structure is converted to a serial format. Typical extenders today are completely transparent and full duplex. They are fairly sophisticated and provide against errors introduced by poor quality data-circuit problems such as dropouts, line breaks, and sync loss.

## CASE HISTORIES

In order to put theory into practice it may be helpful to see how this bus concept has been applied in the Aerospace industry. First, it is important to point out that it has been applied in a very wide range of situations, and the two examples shown here were chosen as being interesting and representative.

The first example is the Watkins-Johnson Company in Palo Alto, California. Among other products Watkins-Johnson manufactures YIG oscillators for microwave sources, such as sweep oscillators and spectrum analyzers. Testing the YIG (Yttrium-Iron-Garnet) oscillator to customer specifications requires a skilled microwave test technician and elaborate test equipment. Only one YIG could be tested at a time. With orders growing rapidly and with skilled microwave technicians in short supply, Watkins-Johnson realized they were a good candidate for automation. The senior measurement engineer who put the system together had had no prior systems experience and did not know computer programming. He very carefully laid out his objectives:

Full Automation of 60 Basic Tests

8.5 Hour Static Heat-Soak and 2 Hour Dynamic Temperature Cycling

Automatic Test Repetition Up to 3 Times

Ability to Custom Interface Existing Oven and Custom Designed Interface Box (GP-IO)

Minimize Test Costs

Conserve Skilled Manpower

Improved Product And Test Quality

Hard-copy Test Results (along with device serial number, test data and time)

Easy Future Expansion Based on Results

An important decision was to buy all the system products from one supplier so he could minimize disagreements between vendors if problems arose. Another key was to choose a desktop computer that is optimized for the single station and is more friendly for the unskilled computer user. The chosen system solution could test 4 devices at a time, run overnight and automatically repeat failed tests three times in an attempt to recover the device. Cost-per-tested unit fell, test quality improved, and the need for more skilled microwave technicians was eliminated. In short, all items which influenced productivity were enhanced and productivity improved 4:1. The price of a YIG device is between $1200 and $2000 for each frequency and power level specified. A simple economic evaluation of the system design showed:

| SAVINGS | |
|---|---|
| 4 Less Test Technicians @ $25K/Yr/Tech | $100K |

PAYBACK
APPROX. 7 MONTHS

| COSTS (approximate) | |
|---|---|
| Equipment | $35K |
| Software Costs (no prior experience) | $20K |
| Test Fixtures | $ 3K |
| Miscellaneous | $ 2K |
| | $60K |

Although this simplified analysis does not take the time value of money into consideration it does provide an excellent insight to the approximate dollar savings for this automated system.

After the system was successfully completed it was expanded to handle 12 YIG's overnight which increased the productivity from the 4:1 initial level to 15:1. An evaluation of the system brought out the point that if it is your first system, don't try to automate too much on the first pass or you may unduly complicate the design and reduce your chances of success.

Hughes Aircraft was chosen for the second example and involves testing RF devices for artillery locating radar systems and also testing devices for a spread-spectrum frequency hopping system. Hughes used high-frequency ATS systems to do production testing on low-volume production of unevenly-spaced contractual programs with various U.S. government agencies.

What Hughes proposed to do was to expand and combine two existing ATS systems into one large system which was (fondly) called ANA. This Automatic Network Analysis System would cover a frequency range of 500 kHz to 18 GHz. The team putting this system together did have some computer skills but were primarily test and measurement personnel. The objectives they laid out for the proposed system were:

Full Automatic Testing of 9 Different RF Devices to Contractual Requirements

Easy Expansion to Include Newer Devices as Required

Hard-copy NBS-Traceable Test Results by Device Serial Number

Pass/Fail Capability

Maximize Measurement Range, Accuracy and Repeatability

Conserve Skilled Manpower and Test Costs

Automatic RF Switching

Automatic Cable Phase-Trim

Graphic Display of Test Results

The Hughes team started with two standard automatic network analysis systems that used the IEEE-488 interface bus. One was a low-frequency system and the other a high-frequency system. These were combined into one 500-kHz-to-18-GHz system. A desktop computer with 250K bytes of user memory and assembly language subprogramming was chosen for the controller. This amount of memory was needed because of the large amount of 8-12 term error correction that is done in state-of-the-art microwave test systems. About 5.5 man-months of software time was needed to merge and expand the existing software.

Many benefits resulted from this system. One significant one was the improved test and device quality resulting from the highly accurate, repeatable, and error-corrected software that was generated. The improved “test integrity” from the hard-copy test results has helped secure additional government contracts. The major savings came from eliminating the 3 proposed manual test-stations and not hiring their accompanying highly-trained technicians.

A rough economic evaluation shows costs of about $175K and saving about $85K (using 5 year straight-line depreciation). This results in a payback period of about 25 months. An extra benefit was that the new system made some measurements that were not made before.

## CONCLUSION

There are several common threads that run thru both these examples. One is to plan your objectives carefully and design the system in an evolutionary way so you can come on-line relatively quickly and with a high probability of success. Other key points include trying to buy everything, if possible, from one supplier and preparing a simple payback analysis in order to visualize the financial impact.

# REFERENCES

1. IEEE Std 488-1978 “Standard Digital Interface for Programmable Instrumentation”, Nov. 30, 1978. Available from the Institute of Electrical and Electronics Engineers, Inc., 345 E. 47th Street, New York, N.Y. 10017.

2. Evans, J.G. and Merrill, H. “A Standard Interface Applied to Measurement Systems”, ISA Proc. of 22nd Int. Instrumentation Symposium, Fundamentals of Test Measurement, Vol. 3, pp. 17-27, 1976.

3. Loughry, D.C., “What Makes a Good Interface?”, IEEE Spectrum, Nov., 1974, pp. 52-57.

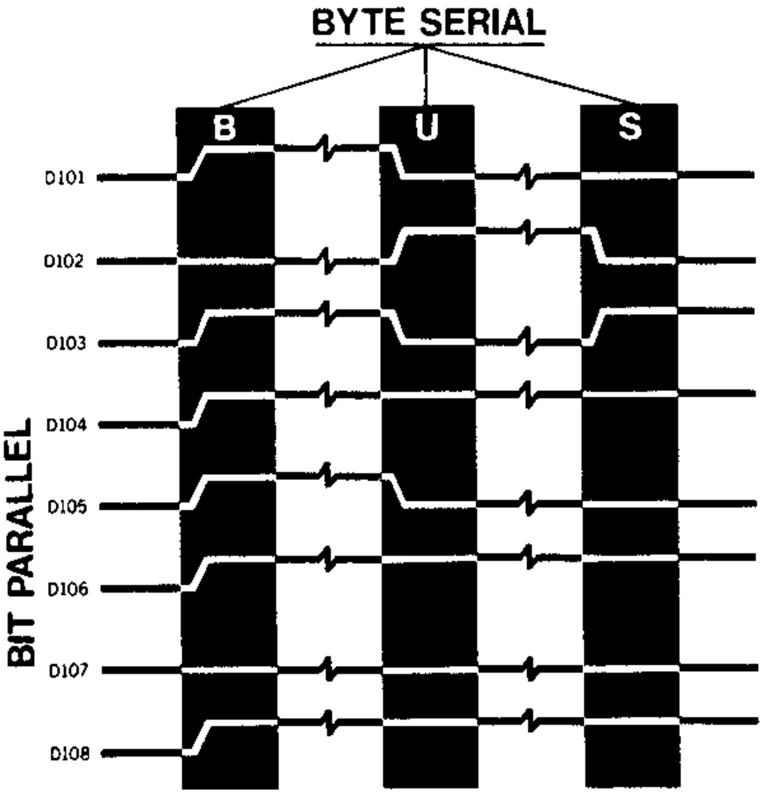
4. Moncrief, F. “Boost Productivity and Slash Errors with Computer Controlled Measurements”, Microwaves, Jan. 1976, pp. 50-61.

5. Ricci, D., “IEEE-488: Its Impact on the Design, Building, and Automatic Test of Measurement Systems, WESCON paper, May 1980.

6. Hewlett-Packard Co., Palo Alto, California, “Tutorial Description of the Hewlett-Packard Interface Bus”, part no. 5952-0156.

7. Kuhn, N., “Noise-Figure Test Setup Provides High Accuracy”, Microwaves, Nov. 1979, pp. 73-76.

8. Lane, Richard Q., “Derive Noise and Gain Parameters in 10 Seconds”, Microwaves, Aug. 1978, pp. 53-57.



| ASCII/ISO | B | U | S |
|---|---|---|---|
| Decimal | 66 | 85 | 83 |
| Octal | 102 | 125 | 123 |
| Hexadecimal | 42 | 55 | 53 |

Figure 1. IEEE-488 DATA STRUCTURE

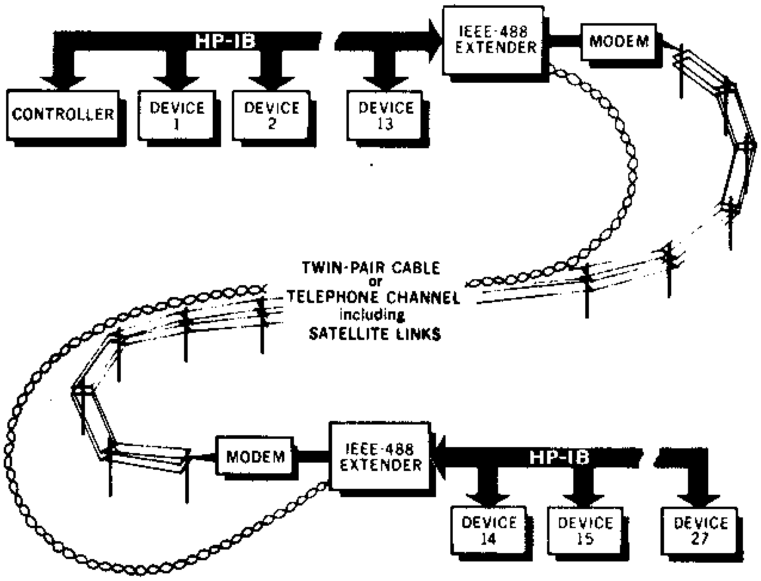


Figure 2. IEEE-488 DISTANCE EXTENDERS