LA-10432-MS

C.3

Data Acquisition Interfaces for the IBM Personal Computer

Los Alamos

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

LA-10432-MS

UC-32
Issued: May 1985

Data Acquisition Interfaces for the IBM Personal Computer

Frank McGirt
Kurt Moore

Los Alamos
Los Alamos National Laboratory
Los Alamos, New Mexico 87545

DATA ACQUISITION INTERFACES FOR THE IBM PERSONAL COMPUTER

by

Frank McGirt and Kurt Moore

ABSTRACT

Data acquisition interfaces composed of standard bus structures such as CAMAC and IEEE-488 have been used with the IBM personal computer (PC). These interfaces have proved to be invaluable for applications that require automation of laboratory and field experiments and laboratory development stations. Software libraries have been developed to provide easy access to these interfaces from high-level programming languages such as C.

I. INTRODUCTION

Current work in the Earth and Space Sciences (ESS) Division at the Los Alamos National Laboratory requires data acquisition support for three general categories of applications. In these applications (described below), a personal computer (PC) served as both an intelligent controller and a primary user interface to each of the data acquisition systems. In Sections II and III, the pros and cons of using the PC for these functions are discussed. How selected data acquisition interfaces are supported in both hardware and software and what user interfaces are available are discussed in Sections IV and V.

A. Field Experiments

The first category is the support of field experiments. The most common type of field experiment support is ground support equipment (GSE). It is used to provide state-of-health information for satellite instrumentation. Most GSE work has required the development of custom interfaces for the PC because the data are not available from standard bus structures. Custom interfaces are not discussed in this report.

B. Laboratory Experiments

The second category is the support of laboratory experiments. This work usually involves the automation of a laboratory experiment, for example, the digitization of a detector response to monoenergetic x rays.[1] Because large numbers of fast pulses are required, the x-ray system was interfaced with a high-speed digitizer (100 MHz) using CAMAC. Almost all work in this area has been done using interfaces to standard bus structures such as CAMAC or IEEE-488.

C. Development Stations

The third category is primarily concerned with the development of new capability for later use in supporting either field or laboratory experiments. A typical application is a system to measure the dynamic properties of transient digitizers.[2] Such a system is important because digitizer specifications from vendors usually give only an indication of static properties, whereas a user really must determine how well a digitizer can represent the instantaneous amplitude of a varying input signal at precise sample times. This system requires that the digitizer interfacing be tested over either CAMAC or IEEE-488 to the PC. Almost all work in this area has been done using interfaces to standard bus structures.

II. WHY USE THE IBM PC FOR DATA ACQUISITION?

Since the introduction of the IBM PC in 1982, a large new industry has evolved that can supply second-source hardware and software for the PC. In many cases such sources allow the experimenter simply to buy the particular item required and then to integrate it into the data acquisition system. With the limited manpower available for most research projects, system integration is much more desirable than OEM development. For most applications, these items are well within the range of even quite modest research budgets.

IBM has published the complete hardware specifications and ROM source code for the PC. Therefore, when commercial products are not available, designing and developing interfaces for the IBM PC is easier than for other computers that are not open systems.

In addition to the flexibility of the IBM PC, the data acquisition performance is respectable with measured rates to memory of more than 300 kilobytes/second using direct memory access (DMA) and 50 kilobytes/second using programmed input.

A. Specific Data Acquisition Interface Hardware for the PC

The authors have used the interface hardware listed below with the IBM PC. This list is a small subset of the total products available, and no special endorsement of any item is intended.

1. Bus Interfaces.

CAMAC Controller - Transiac

IEEE-488 Controller - Ziatech, National

STD Controller - Ziatech

2. Data Acquisition.

Digital/Analog Conversion - Tecmar, Data Translation

Analog/Digital Conversion - Tecmar, Data Translation

Digital Input/Output - Tecmar, Data Translation

Timers - Tecmar, Data Translation

3. Graphical Output.

Monochrome Graphics - Hercules

Color Graphics - Tecmar

B. Software

Some examples of software used to support data acquisition for the IBM PC are given below. This list is a small subset of the total products available, and no special endorsement of any item is intended.

1. Operating Systems.

PCDOS, CP/M, UCSD, Xenix, Unix, Pick

2. Programming Languages.

C, FORTRAN, Pascal, Forth, Basic

III. WHY NOT USE THE IBM PC FOR DATA ACQUISITION?

Although using the PC for data acquisition has many advantages, several disadvantages also exist, which are discussed below.

A. Hardware Considerations

1. Architectural Design Deficiencies. IBM made several design decisions for the PC that penalize data acquisition performance. The first decision was to use the Intel 8088 CPU at 4.77 MHz with the attendant eight-bit data bus. A 10-MHz Intel 8086 with 16-bit data paths would be much preferable for data acquisition applications.

A second design decision that limits data acquisition performance is the fragmented memory map of the PC. The 8088 can address 1 megabyte of physical memory, but IBM, by the arrangement of ROM and reserved space, has limited the user to 640 kilobytes of memory.

The PC has both limited interrupt and direct memory access capability. The interrupt controller supports eight levels of interrupt; however, only three of these can be used for user applications. For user applications that require DMA, a single channel must be shared among all user DMA devices if a fixed disk is installed on the PC. With no fixed disk, two DMA channels are available for user devices.

2. Speed. The IBM PC has two speed-related problems: it uses relatively slow peripherals and lacks fast data analysis support. Even though it is possible to acquire data to PC memory quickly, little can be done with the data on a real-time basis. Therefore, use of the PC is limited to those data acquisition applications that do not require rapid continuous data acquisition or lengthy data analysis computations between data inputs.

3. Incompatibility. Finally, there are problems with incompatible second-source hardware. For data acquisition applications, the three most likely areas of incompatibility are memory conflicts (memory on separate interface adapters requiring the same address), interrupt conflicts (separate interface adapters requiring the same interrupt level), and DMA conflicts (separate interface adapters requiring the same DMA channel with no tristate channel acquisition scheme).

B. Software Considerations

The major problems with PC software for data acquisition have been in the area of high-level programming languages. Several PC languages have deficiencies for data acquisition; some of the deficiencies are slow execution speed, no hardware floating point support, generation of huge object modules, poor error detection, no byte data type, no masking operations, no hex or octal output formatting, no access to operating system functions, and complicated assembly language interfaces. However, this situation is rapidly improving particularly with the FORTRAN, Pascal, and C languages.

IV. DATA ACQUISITION INTERFACES

The ESS Division at Los Alamos has used the CAMAC and the IEEE-488 commercial data acquisition interfaces.[3-7] The implementation and support of these interfaces are described below.

A. CAMAC Interface

1. Hardware. The CAMAC interface hardware used for the PC was the Transiac Model 6002 CAMAC-Microprocessor Interface with DMA.[8] This unit consists of a dataway controller module with dataway display connected by ribbon cable to an interface adapter, which is installed in a PC expansion slot. Functionally, the interface can be used to control dataway modules by writing data to a module, reading data from a module, detecting look-at-me (LAM) signals from modules, and executing CAMAC initialize and clear cycles.

The 6002 uses 16 consecutive addresses on the IBM PC input/output channel as interface registers. These registers are defined as follows:

Register	Function
WHREGW	CAMAC w24-w17 (write)
WMREGW	CAMAC w16-w9 (write)
WLREGW	CAMAC w8-w1 (write)
AREGW	CAMAC subaddress, A (write)
FREGW	CAMAC function, F (write)
NREGW	CAMAC station, N (write)
ZREGW	CAMAC auxiliary register, Z, C, I (write)
SREGW	Start CAMAC cycle (write)
LREGR	CAMAC LAM, X, Q (read)
RHREGR	CAMAC r24-r17 (read)
RMREGR	CAMAC r16-r9 (read)

RLREGR	CAMAC r8-r1 (read)
STREGR	6002 status (read)
CONREG	PC control register (control)

Using the interface is therefore simply a matter of writing appropriate software to communicate with the hardware with the interface registers as described in the next section.

2. Software Support Library. To provide access to the CAMAC hardware interface from high-level languages such as FORTRAN and C, a library of subroutines and functions was developed. These functions are similar to those written in Basic, which Transiac supplies with their hardware, and include the following:

Function	Operation Performed
CAMO	Write data in w1-w24 to CAMAC dataway
CAMI	Read data in r1-r24 from CAMAC dataway
CAML	Return the station number of the highest priority LAM and return the status of Q and X from the last CAMAC cycle
CAMCL	Set the state of the CAMAC Z, C, and I signals
DMASET	Set parameters for DMA data transfer
DMAI	Transfer data from the CAMAC dataway to computer memory with DMA
DMAO	Transfer data from computer memory to the CAMAC dataway with DMA
INIT	Initialize the Transiac 6002 CAMAC crate controller

With these functions, a user can quickly and easily write a simple application program to control a CAMAC crate and to acquire data from any module resident in the crate. See Ref. 9 for details concerning the software support library.

B. IEEE-488 Interface

1. Hardware. The IEEE-488 interface hardware used (or equivalently, the GPIB) was the Ziatech Model ZT1488.[10] This unit is a single printed circuit card that is installed in a PC expansion slot. Functionally, the IEEE-488 modules can be controlled by using the interface and the full set of IEEE-488 functions:[4] specifying talkers and listeners, changing module status

from local to remote, writing data to or reading data from modules using programmed input/output or DMA, detecting service request (SRQ) signals from modules, conducting parallel or serial polls of groups of modules, etc. The ZT1488 uses the Texas Instruments TMS 9914A GPIB controller, which has 13 addressable registers. All communication between the GPIB and the PC is carried out with these registers, which are defined as follows:

Register	Function
INT STATUS 0	Interrupt conditions such as byte in
INT STATUS 1	Interrupt conditions such as bus error
ADD STATUS	Read ZT1488 talker/listener state
BUS STATUS	Unlatched GPIB control line monitor
ADD SWITCH	Read on board DIP switch address
1488 INT STAT	Interrupt status of board
CMD PASS THR	Unlatched GPIB data line monitor
DATA IN	Read data from GPIB
INT MASK 0	Enable interrupts for INT STAT 0
INT MASK 1	Enable interrupts for INT STAT 1
AUX CMD	TMS 9914A commands (talk only, etc.)
ADD REG	Set ZT1488 talker/listener address
SER POLL	Write ZT1488 serial poll response
PAR POLL	Write ZT1488 parallel poll response
DATA OUT	Write data to GPIB

Using the interface is therefore simply a matter of writing appropriate software to communicate with the hardware as described in the next section.

2. Software Support Library. To provide access to the IEEE-488 hardware interface from high-level languages such as FORTRAN and C, a library of subroutines and functions was developed. These functions are similar to those written in Basic, which Ziatech supplies with their hardware, and include the following:

Function	Operation Performed
INIT	Initialize interface
CHKDEV	Check if a specified device is present on the bus
SETPRI	Select primary IEEE-488 port
SETSEC	Select secondary IEEE-488 port
SENIFC	Toggle Interface Clear control line

DEVCLR	Send Device Clear message
TIMOUT	Set ZT1488 bus response wait period
SRQST	Return status of SRQ line
SPOLL	Conduct serial poll
WAIT	Delay for a specified number of milliseconds
RECBIN	Receive binary data from a specified talker
SENBIN	Send binary data to specified listeners
RECDMA	Receive binary data using DMA from a talker
RECSTR	Receive ASCII data from a specified talker
SENSTR	Send ASCII data to specified listeners
CMD	Send arbitrary IEEE-488 command
REMOTE	Set Remote Enable bus line true
LOCALZ	Set Remote Enable false and send Go To Local message to listeners
PPOLL	Perform a parallel poll and return status
PPUA	Unconfigure all devices from responding to a parallel poll
PPEN	Configure a single device for parallel polling
PPDS	Disable specified devices from responding to a parallel poll
LCKOUT	Disable Local switch on listener devices
TRIG	Address specified devices and send Group Execute Trigger
TERM	Select input stream terminator character
XFER	Enable data transfer between two devices other than the ZT1488

With these functions, a user can quickly and easily write a simple application program to control IEEE-488 devices and to acquire data from any device or devices present on the bus. See Ref. 11 for details concerning the software support library.

V. USER INTERFACE SUPPORT

At this time specific user interface support is limited to monochrome and color graphics. This support includes libraries of subroutines and functions callable from FORTRAN and C for the Hercules Graphics Adaptor[12] and the Tecmar

Graphics Master.[13] As an example of the support available, the Hercules library includes the following functions:

Function	Operation Performed
ARC	Draw a quarter circle arc
BLKFIL	Fill rectangular region
CIRC	Draw circle
CLRSCR	Clear currently active screen buffer
DISP	Display currently active screen buffer
DLINE	Draw line
FILL	Fill concave polygon
GETPT	Return intensity level of pixel
TMODE	Select text mode
GMODE	Select graphics mode
GPAGE	Select screen buffer
LEVEL	Select intensity level
GMOVE	Move imaginary cursor to selected pixel
PLOT	Set or clear pixel
TEXT	Write character string

Work is currently under way to develop other user interface libraries, for example, libraries to provide menu and window generation and to support graphical input devices such as a mouse.

VI. SUMMARIZING REMARKS

Interfaces to the IBM PC for data acquisition have proved to be simple and fast for a variety of applications. These interfaces are efficient because software libraries are being developed that provide high-level programming language access to the rather low-level data acquisition functions of the hardware interfaces. Also, the IBM PC has unsurpassed flexibility because hardware and software are widely available. Finally, because IBM has marketed the PC as an open system, the productivity of users who develop hardware and software for it has vastly increased.

REFERENCES

1. G. J. Berzins, J. E. Valencia, J. W. Gutierrez, F. McGirt, K. R. Moore, "Pulsed Fluorescent X-Ray System with On-Line Digitizer and Data Processor," Los Alamos National Laboratory report LA-10247 (December 1984).

2. K. R. Moore and F. McGirt, "A System for Measuring the Dynamic Properties of Transient Digitizers," Los Alamos National Laboratory, Group ESS-10 (1985).

3. "Modular Instrumentation and Digital Interface System (CAMAC)," IEEE Standard 583-1975.

4. "Serial Highway Interface System (CAMAC)," IEEE Standard 595-1976.

5. "Parallel Highway Interface System (CAMAC)," IEEE Standard 683-1976.

6. "Block Transfers in CAMAC Systems," IEEE Standard 683-1976.

7. "Standard Digital Interface for Programmable Instrumentation," IEEE Standard 488-1978.

8. Transiac Corporation, "Transiac Model 6001/6002 CAMAC-Microprocessor Interface," (1983).

9. Frank McGirt, "TRANIO, A Subroutine Package for the IBM Personal Computer to Support the Transiac 6002 CAMAC Crate Controller," Los Alamos National Laboratory report LA-10145-MS (June 1984).

10. Ziatech Corporation, "ZT1488 Multifunction GPIB Controller for the IBM PC," Operating Manual (June 1983).

11. Frank McGirt and Kurt Moore, "ZTL, A Subroutine Package for the IBM Personal Computer to Support the Ziatech ZT1488 IEEE-488 Controller," Los Alamos National Laboratory report LA-10141-MS (June 1984).

12. Frank McGirt, "HGRAPH, A Subroutine Package for the IBM Personal Computer to Support the Hercules Graphics Card," Los Alamos National Laboratory report LA-10144-MS (June 1984).

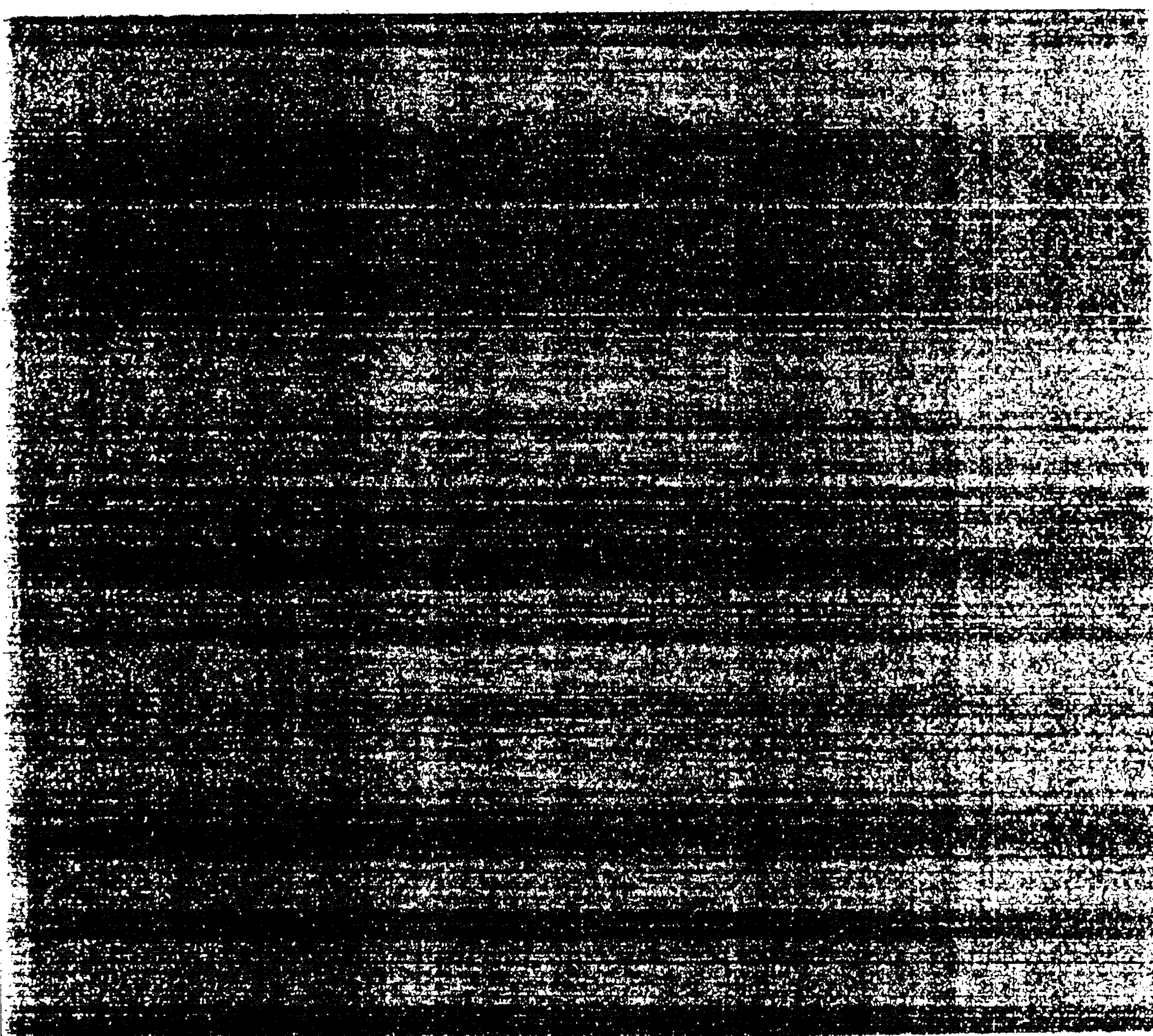
13. Frank McGirt, "TGRAPH, A Subroutine Package for the IBM Personal Computer to Support the Tecmar Graphics Card," Los Alamos National Laboratory, Group ESS-10 (1985).

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Microfiche (A01)

Page Range	NTIS Price Code	Page Range	NTIS Price Code	Page Range	NTIS Price Code	Page Range	NTIS Price Code
001-025	A02	151-175	A08	301-325	A14	451-475	A20
026-050	A03	176-200	A09	326-350	A15	476-500	A21
051-075	A04	201-225	A10	351-375	A16	501-525	A22
076-100	A05	226-250	A11	376-400	A17	526-550	A23
101-125	A06	251-275	A12	401-425	A18	551-575	A24
126-150	A07	276-300	A13	426-450	A19	576-600	A25
						601-up*	A99

*Contact NTIS for a price quote.



Los Alamos