FINAL TECHNICAL REPORT

WELDING PROCESS MODELLING AND CONTROL

26 February 1992 through 25 February1993
Contract Number NASA-38609
Delivery Order #24

Prepared for:

George C. Marshall Space Center
Marshall Space Flight Center, Alabama 35812

16 April 1993

by

Peter L. Romine
Jinen A. Adenwala

Electrical and Computer Engineering Department
The University of Alabama in Huntsville
Huntsville, Alabama 35899

(NASA-CR-192549) WELDING PROCESS MODELLING AND CONTROL Final Technical Report, 26 Feb. 1992 - 25 Feb. 1993 (Alabama Univ.) 44 p

N93-27593

Unclas

G3/37 0171076

TABLE OF CONTENTS

1.0 INTRODUCTION

This report documents the research and analysis performed, and software developed, and hardware/software recommendations made during 1992 in development of the PC-based data acquisition system for support of Welding Process Modeling and Control.

A need was identified by the Metals Processing Branch of NASA Marshall Space Flight Center, for a mobile data acquisition and analysis system, customized for welding measurement and calibration. Several hardware configurations were evaluated and a PC-based system was chosen. The Welding Measurement System (WMS), is a dedicated instrument, strictly for the use of data acquisition and analysis. Although the WMS supports many of the functions associated with the process control, it is not the intention for this system to be used for welding process control.

1.1 WELDING MEASUREMENT SYSTEM SPECIFICATION

The following is the initial specification for the Welding Measurement System.

High-speed Differential Input Analog To Digital (12bit)

	Signal	Range	Samples/Sec(Min)
1	Arc Voltage	+/- 400V	4000
2	Arc Current	+/- 500mV	4000
3	Pilot Arc Voltage	+/- 400V	4000
4	Pilot Arc Current	+/- 500mV	4000
5	Phototransistor-Arc Light		

Low-speed Single Ended Input Analog To Digital (8-12 Bit)

	Signal	Range	Samples/Sec(Min)
1	Shield Gas Flow	0-5V	10
2	Shield Gas Pressure	0-5V	10
3	Plasma Gas Flow	0-5V	10

4	Plasma Gas Pressure	0-5V	10
5	Backpurge Gas Flow	0-5V	10
6	Backpurge Gas Pressure	0-5V	10
7	Wire Feed Speed	0-5V	10
8	Temperature	0-5V	10
9	Travel Speed	0-5V	10
10	LVDT Torch Position	0-5V	10

Digital Inputs (0,5v)

Encoders: Initially 1 -- Travel Position

Future: upto 4 1 axis travel
1 axis rotation
1 axis position
1 wire speed

Output Data Requirements

The following is a list of processed data to be generated.

The data is to be output in a LOTUS .PRN (tabed ascii) file.

1 Avg Straight Voltage
Running Avg,Mean,StdDev Updated @0.25s
2 Avg Straight Current "
3 Avg Reverse Voltage "
4 Avglpql Reverse Current "

5 Straight Polarity Time
6 Reverse Polarity Time
7 Phase Shift Arc V,I
1 per cycle
8 Phase Shift Pilot Arc V,I
1 per cycle
9 Arc Ripple V,I
10 Pilot Arc Ripple V,I

ASSUME:

Normal weld time <= 10 minutes/run. Possible to have 2-6 Hours/run.

2.0 CRITERION FOR SELECTING DATA ACQUISITION BOARD/SOFTWARE

PC based data acquisition is compromised of analog and digital sensors, analog circuitry, signal conversion technology, digital logic, computer architecture and software. This section briefly explains some of the key specifications and concepts in each of the applicable technologies, and provides some directions for selection of the product.

2.1 ANALOG INPUTS

Analog inputs are a common criterion for the preliminary assesment of a data acquisition board. Most analog input boards are designed to measure voltage with additional signal conditioning and/or some software scaling allows virtually any type of input signal can be converted into the analog unit desired. (e.g. thermocouple inputs are easily converted into ^{o}C or ^{o}F.) . Following factors pertaining to the analog inputs should be considered :

2.1.1 Input Resolution

Resolution is specified in "bits". The available products range from 8–bits to 16–bits, with the majority of commercially available products offering 12–bit resolution.

$$\text{Resolution} = \text{one part in } 2^{(\#\text{ of bits})}$$

To determine the resolution in volts, take the total input range and divide it by result of the above equation.

e.g. For 12–bit resolution with an input range of -5 V to +5 V,

$$\begin{aligned}\text{Resolution (in volts)} &= 10\text{ V FS}/\,2^{12}\\ &= 10/4096\\ &= 0.00244\text{ V (2.44 mV)}\end{aligned}$$

Higher resolution A/D converters are more expensive and may not be required for a particular application. For example, if the sensor has an accuracy of 1%, using 16-bit A/D board will add unnecessary expense to the overall system. To avoid this, the desired resolution of the measurement should be matched with the resolution of the A/D board.

2.1.2 Input Accuracy

Input accuracy is related, but not equal to input resolution. Both resolution and accuracy should be checked carefully as it is possible to have a 16-bit A/D converter which is only 12-bit accurate.

Specification : +/- 1 bit

Measurement accuracy = $10\ V^* (0.024/100 + 1/2^{12})$

$= 4.8\ mV$

Specification : +/- 2 bit

Measurement accuracy = $2^* (10\ V/ 2^{12})$

$= 4.8\ mV$

Specification : 0.048% of FSR (Full Scale Range)

Measurement accuracy = $10\ V * (0.048/100)$

$= 4.8\ mV$

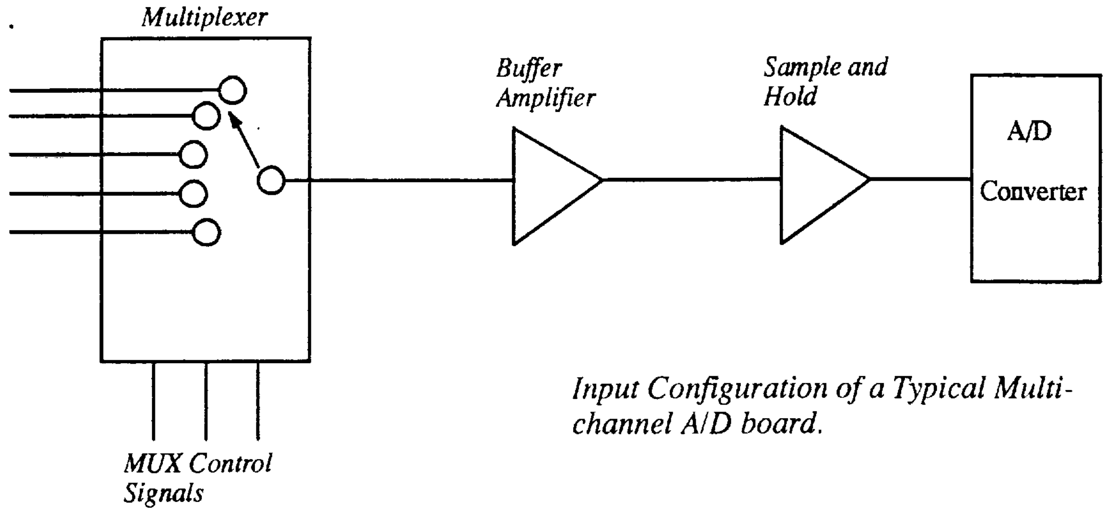
2.1.3 Maximum Sampling Rate

This is often the most important criterion for selection of an A/D board. The maximum sampling rate is specified in samples per second. Most multi-channel A/D boards consist of a single A/D converter and input multiplexer. The multiplexer acts as a switch allowing each input channel to be sampled independently. The maximum sample rate per channel is equal to the maximum sample rate of the A/D board divided by the number of channels being sampled. As an example, if an 8 channel A/D board is specified at 100K samples/sec, and if 4 channels are being sampled, then

Maximum sampling rate per channel = 100,000/4

= 25,000 samples/sec

a



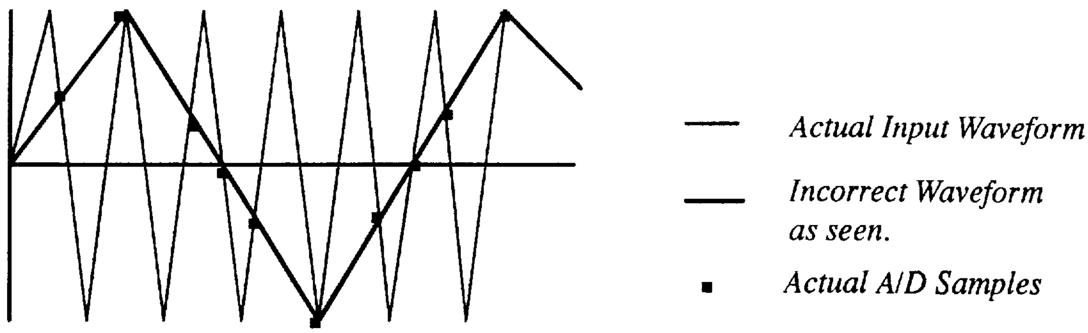
Input Configuration of a Typical Multi-channel A/D board.

At high sample rates, computer memory is filled quickly. The A/D card can sample at a desired speed and for a desired amount of time. For high speed acquisitions, special high-speed disk access software (disk "streamers") is used for streaming data directly to the hard-drive. Many very fast data-acquisition systems are designed with on-board memory so that sample rates are not limited by computer speed. As per Shannon's sampling theory, the minimum sampling rate should be at least twice as fast as the highest frequency component of the input signal for accurate information to be acquired. Higher sampling rate is recommended, if possible.

Another sampling rate factor is aliasing. This is the phenomenon that makes a helicopter's rotor blades appear in movies as slowly moving backwards. In a data acquisition system, the same process can occur, and the analog input can incorrectly show a slowly moving input signal which actually is a high-frequency phenomenon. This is predominant if the input signal contains frequencies higher than the systems sampling rate. Then an anti-aliasing filter is recommended. This filter is a very sharp roll-off, low-pass filter that allows the valid signals to pass while removing the undesired high-frequency error signals. Typically the anti-aliasing filter is set with the cut-off frequency of half the sample rate.

2.2 A/D Converter Types

There are four common types of A/D converters used in data acquisition equipment. These



are: Successive Approximation, Integrating, V/F counting, and Flash Converter. The Successive Approximation type is most commonly used.

Converter Type	Speed	Resolution	Noise Immunity	Cost
V/F Counting	Slow	16 - 24 bits	Very Good	Medium
Integrating	Slow	12 - 18 bits	Very Good	Low
Successive Approximation	Medium	10 - 16 bits	Little	Low
Flash	Very Fast	4 - 8 bits	None	High

In low speed applications, an integrating converter (often referred to as Dual-slope converter) may be a better choice. It has a maximum sampling rate of 100 sample per second and is less susceptible to noise than successive approximation devices and is a better choice if input signals are slowly changing. For extremely slow sampling rates, (less than 100 samples per sec) a V/F counting converter is the best choice. This converter has very good noise immunity and offers extremely high resolution. The resolution is obtained by simply counting the V/F converter's output. Flash converters are extremely fast (upto 10 million samples/ sec or more), but must compromise with a have very limited input resolution (available with 4, 6 and 8 bit resolution). These type of converters are used in oscilloscope and video frame grabber products. This converter is used in MetraByte's ultra high speed PCIP-SCOPE.

2.3 A/D Triggering and Data Transfer Methods

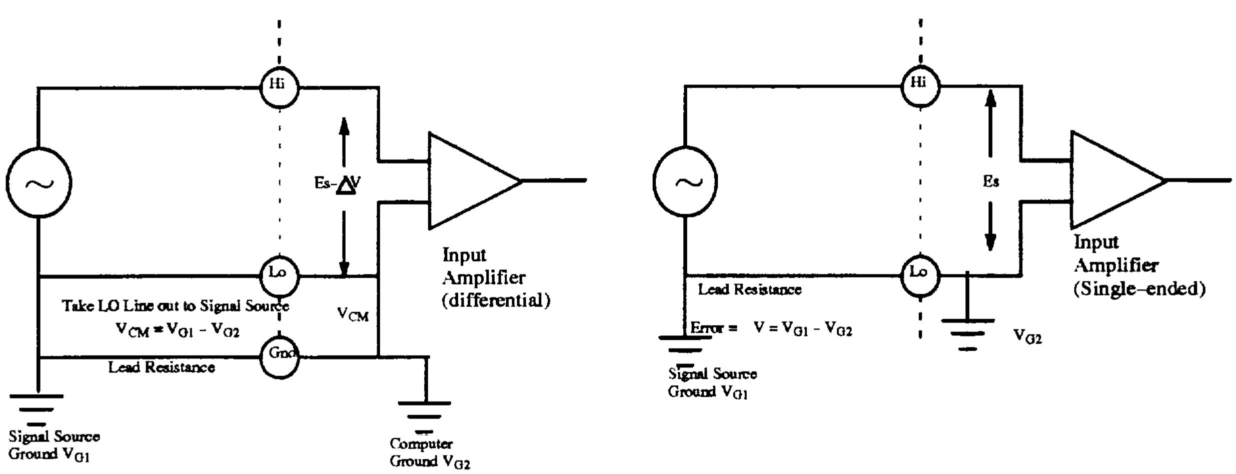
This is also an important consideration. In many applications, the presence of any jitter in the inter-sample timing will cause large errors in any subsequent analysis. The loss or skipping of a sample could easily render the data useless and the user may not even know the the sample has been lost. The following care should be taken to avoid inter-sample jitters . The A/D conversion should be initiated directly by the pacing clock in the board's hardware. Systems that use software routines to start conversions will always be susceptible to jitters. This is because most of today's computers periodically interrupt the computers operation to perform such tasks as memory refresh, disk access , real-time clock updates, etc. Though these interrupts occur too quickly to be noticeable to the human eye, they can drastically alter the timing of the sampled data. There are three types of sample modes used to acquire data. The Post-trigger mode can start taking data on a trigger, and stop taking data based on a trigger (Pre-Tigger). Some boards take data before and after the trigger (Trigger-about mode). Pre-Trigger and Trigger-about modes are extremely useful when part of the data of interest is the condition of experiment before the event occurred.

High speed A/D boards use either DMA(Direct memory access) or interrupt driven data transfer. DMA transfer takes the data from the data acquisition boards and puts it directly into the computers memory. This transfer is completely hardware controlled and all software executions are suspended during this transfer. Since DMA transfers are completely hardware controlled, they are extremely fast. An interrupt causes the computer to halt the current program and jump to a different program. This routine will then take the data from the data acquisition board and put it in memory and then give control of the computer back to the original program. For lower speed applications , interrupt driven applications are perfectly adequate. Extremely high speed boards require on-board memory so that their speed is not restricted by the computers bus speed.

2.4 Input Signal Conditioning

Most acquired real–time signals are not suitable as direct A/D inputs. They have to be amplified, attenuated, or otherwise modified before conversion. This section describes some important signal conditioning considerations as they pertain to data acquisition systems. Input gain and attenuation is the signal conditioning required by most systems. For accurate use of the systems A/D converter, the input range must match the full–scale deflection of the input signal. Some boards have a fixed input range while others have a selectable input range. For boards with a fixed input range, signal conditioning must be done outside the board or at the sensors. For boards with selection switches, select the range that matches the full scale deflection of the input signal. Software programmable input ranges can be used in any application , but are more expensive. Some systems offer a choice between bipolar (+/- 5V) and unipolar (0 - 5V). Some boards are custom made for a particular application.

There are two types of input configurations, the Differential input configuration and the Single–ended input configuration. Differential input is a little more complex to use and is more expensive, but offers better noise immunity. This is particularly important when data is being acquired from several different sources and/or the sensors are located at large distance from the board.



Errors Caused by V_{CM} are reduced by Common-Mode rejection of the Input Amplifier. (Typically 80 db or greater)

2.5 Input Isolation

Input Isolation is essential to protect the computer and expensive hardware from input voltage surges that could cause the system damage. High voltage inputs are not always accidental. For example, a temperature sensor mounted on a motor stator could be at 120VAC although it would not affect the performance of the temperature sensor. Without isolation, the data acquisition system could be easily destroyed by the AC line voltage. Isolation also eliminates the effect of ground loops.
Systems that operate at high gains, are made up of a large number of different sensors/instruments can often be plagued by excessive ground noise. Isolating the data acquisition inputs from a noisy ground will greatly increase system accuracy.

Isolation can be provided in a number of ways. The most common way of isolating the analog signals is by using a transformer. Optical and capacitive isolators are becoming popular. Some sensors give the already isolated output and do not require further isolation. Sensors based on the Hall Effect to measure currents, optical encoders , capacitive encoders are examples of these. Sometimes, equipment with isolated inputs can cost twice as much as the non isolated ones, but the added advantages such as long term reliability and accuracy, compensates for the initial cost involved.

2.6 Special Signal Conditioning

Some sensors require special signal–conditioning circuitry. For example, thermocouples require cold–junction compensation, RTD's and strain gages require excitation circuitry. Care should be taken that the system selected provides proper inputs and outputs for each application. Some data acquisition boards provide these facilities to simplify taking accurate measurements.

2.6.1 Non–A/D Functions

Most Data Acquisition systems require some combination of digital inputs and outputs, analog outputs, counters/timers , motor controllers, etc ... . Digital I/O is the most commonly used non–A/D function. Digital inputs are used to monitor switch closures, sense power on–off conditions, control motor/heaters, activate relays. Position signals from optical encoders are also digital inputs. Hence pertaining to the application, selection of A/D boards should be such that they provide some level of digital I/O capability. For control applications, along with the digital outputs, the computer should also provide analog outputs. These outputs are used to generate excitation voltages, control valves, generate waveforms and simulate outputs from other devices. Boards featuring analog outputs have additional D/A converters. These unnecessarily elevate the cost if the system application is purely data acquisition and data monitoring. Counting and timing is another data acquisition function. The timers are used to accurately set sample rates on analog inputs and outputs. They are also used to measure frequency, count events, measure time and delays and generate known output frequencies.

2.7 Software

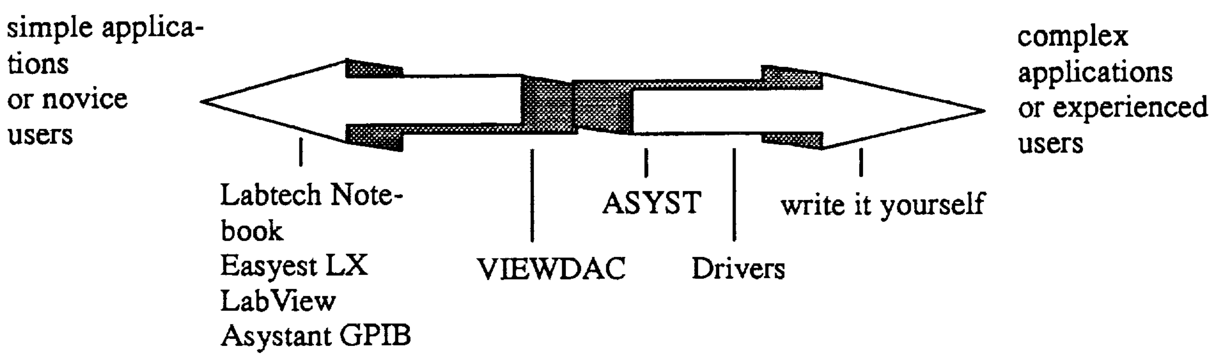
Once the hardware is selected, software is necessary to pull the application together. There are several integrated packages available. One can also write his own program but it is not recommended as it is a very time consuming process and requires excellent programming skills. There are some very reasonably priced packages which require little programming. The software decision will greatly affect the outlook of the overall data acquisition system, its functionality, and the effort required to get it running. Apart from acquiring and manipulating the data, the user should have other flexibility. The software should be very user–friendly as most of the time, the person using it may be a non–technical person. Factors to be considered while making the decision are :

- Capabilities needed : Acquisition, Graphics, Analysis.
- Hardware used : Plug–in, GPIB or RS–232 instruments.
- User interface required: Displays, Automation, Window environment.
- Time and expertise available : Language, Knowledge.
- Cost factor .

2.7.1 Software Selection

Ease of use: Simple systems are intutive, and using them requires minimum study. Systems that fall in this category are based on interactive state of art user interfaces (i.e. menus or windows). More complex systems are based on programming languages, the most difficult being assembly language coding for a particular processor.

Adaptability: 'Complex' systems are designed with high priority on a few aspects of data acquisition and control. For instance, a package can simplify data logging through tightly defined menus and may even be able to plot real-time graphics. However adding control tasks and even changing the type of graphics can be difficult. These systems generally have a predefined model from which they cannot deviate easily. They are limited by their ability to make descisions and to adapt themselves to a different application. "Flexible" systems are usually based on some type of programming language and allow the user to make decisions on what the program will do under various conditions. The system can also be altered to adapt to changing requirements. A package offering optimum compromise between flexibility, performance and ease of use, is the best choice.



Processing speed: This is a critical consideration for real-time systems. How fast can the system acquire and store data? How fast can the system analyze data? How fast can the system display data? (i.e. can it do real-time graphics or only post acquisition display of data?) Can this system perform control functions (can it make decisions in real-time and how powerful is the decision making capability?). 'Slow' systems are generally useful for slower decision making or control application.

These three factors, define a spectrum with simple systems at one end and complex systems on the other. The chart above shows the spectrum and places some of the commercially available software packages. The placement of these packages on the diagram is based on eaches generally accepted strong points. To choose a software, determine where the application lies along the software selection spectrum, the time spent in trouble shooting the applications and future application needs.

Then after reference to detailed software description, make the software selection. The vendor supplied specifications of some of the popular data acquisition packages are attached . These software also mention the boards that they are compatible with. This makes the selection procedure a lot easier for the system designer.

With these basics, it is a straight forward process easy to design a data acquisition system to match one's need and budget. The project discussed in these report puts these things together to design a data acquisition system for a A Welding Robot .

2.8 Computer Considerations

IBM PC/XT and IBM PC/AT and compatibles are by far the most popular host computers for data acquisition. PC/AT bus is referred to as ISA (Industrial Standard Architecture) bus. The PC/XT bus performs bit data transfer and is capable for transfer upto 100K words (16 bits) per second. The ISA bus performs 16-bit transfers and can perform upto 300K transfers per second. Boards with on-board memory offer 1M samples/sec or greater even in PC/XT bus computers.

Several data acquisition products are available for IBM PS/2 (Micro Channel) computers and compatibles. In general, the data acquisition products donot require powerful computers. However when combined with analysis and graphical display, powerful computers (80386 and 80486) are strongly recommended. Often the softwares selected for application will dictate the type of computer to use. For example, extremely powerful VIEWDAC package requires atleast a 386 with 6 MBytes of memory and a 387 coprocessor).

Standard PCs may not be suitable for certain industrial or harsh lab environments, due to factors such as heat, shock, electrical noise, vibrations. To deal with such environments, many manufactures are producing rugged PCs for the market. Some of this PCs use standard motherboard with expansion slots, and many others use passive backplane. A passive backplane is the backbone of many industrial computer bus systems (such as VME and STDBus) as well as many mainframe computers and early microcomputers. It is simply an array of connectors wired together to form a bus, without any active circuitry present. The CPU is on card, plugged into the bus. This adds extreme flexibility since upgrading to another processor simply involves switching the CPU card. This also guarantees all signals required by the CPU are present on the passive backplane, adding flexibility for having multi-processors. The major disadvantage for this approach is the added cost involved.

A continuing problem in the area of industrial PC systems is how to enhance the ISA standard while maintaining compatibility with products from different manufacturers. A new standard called PCXI from Rapid Systems is a potential solution. It is intended for multivendor standard for data acquisition and industrial instrumentation systems. It incorporates a standard a standard ISA passive backplane and power supply into a modified PC chassis. PCXI is PC equivalent to VMXI, the VME bus instrumentation. It supports several manufactures and will probably become established as a true standard.

2.8.1 The VXIbus

The VXIbus is a fast growing platform for instrumentation systems. It was introduced in '87 and has become very popular since then. VXI is based on worldwide VMEbus standards and so VME modules can be used in VXI systems. The VXI backplane includes a 32-bit VME computer bus as well as high performance instrumentation buses for precision timing and synchronization between instrument components. VXI benefits user in following the ways:

- Increased system throughput
- Smaller size and higher density
- Reduced cost
- More precise timing and synchronization
- Standardized, multivendor protocol for systems configuration and pro gramming

Data acquisition and control systems based on the VMEbus have long been popular in industrial research environments. With VXI, we can use existing VME modules and also enhance VME to improve performance and reliability. VXI's packaging handles very high density in a single module and therefore very attractive for applications with high channel counts. A variety of self-processing modules, including digital signal processing (DSP), are available with VXI and VME. With VXI's multiprocessing architecture, high data transfer rates and shared-memory capability, we can process multiple channels of acquired data can be processed in real time.

2.8.2 Plug-in boards Versus external Systems

Data acquisition systems can be classified in two basic types. The plug-in boards where the data acquisition boards plug directly into the computer , and the external systems where the whole data acquisition system is mounted outside the computer on an external chassis and is connected to the computer through some type of communication interface (e.g. RS-232 , RS-422 , IEEE 488). Depending upon the application, one or the other or sometimes both (hybrid systems) are selected to best match the application. The following table describes the features of both systems:

External systems also have several advantages. Since they are not tied to a specific computer bus, they can be used by "closed architecture" computer that have no usable slots. External systems are preferred when number of I/O channels are more than 50 - 100 since it is very difficult to physically connect large number of wires to a personal computer. Their enclosure and their power supply are specifically designed for data acquisition applications. This results in more accurate measurements at high speeds.

As the plug-in boards plug directly in the computer, use its power supply and also does not require an external chassis , they are less expensive. Also, as they plug directly into computers bus, they have ability to transfer data directly into host computers memory at full bus speeds. Though external systems have ability to acquire data in local memory at high speeds,

FACTOR	PLUG-IN BOARDS	EXTERNAL CHASSIS
Cost	Low	Moderate
Expandability	To 100 Channels	Almost Unlimited
Portability bet. Computers	Only with Compatible Slots	Bus independent
Data Transfer to Computer memory	Full computer Bus Bandwidth	Usually less than 60 - 70 KHz

the actual transfer of data to computer is limited by the communication link. Also where space is the restriction, plug-in boards are ideal choice. There are wide range of data acquisition boards available by various manufacturers. These boards are designed to be good for a particular application. They all vary in their specifications as shown in the attached data sheets. Based on ones need, selection of the data acquisition board and the relevant software has to be done as described above.

After the data sheets, we will briefly explain why we opted for CIO-AD16Jr and Driver-LINX combination for the project.

IBM PC/XT/AT ANALOG & DIGITAL I/O BOARDS

	ANALOG INPUTS						ANALOG OUTPUTS		DIGITAL I/O			
BOARD	COMPUTER	NO. CHANNELS	BITS	MAX SAMPLE (S/S) RATE	INPUT RANGE (VOLTS)	RANGE SELECTION	CHANNELS	RESOLUTION	INPUTS/ OUTPUTS	PACER CLOCK	COUNTER/ TIMERS	PAGE
12-BIT RESOLUTION												
DAS-58	PC/XT/AT	8 SE	12	1 M	±10, ±5, ±2.5 +10, +5	Software	—	—	—	Y	0	29
DAS-50	PC/XT/AT	4 SE	12	1 M	±10, ±5, ±2.5 +10, +5	Software	—	—	—	Y	0	33
DAS-40G1	AT Only	16 SE/ 8 Diff	12	45 K	±10, ±1, ±0.1, ±0.02 +10, +1, +0.1, +0.02 V	Software	2	12	3 out, 8 in	Y	0	37
DAS-40G2	AT Only	16 SE/ 8 Dif	12	250 K	±10, ±5, ±2.5, ±1.25 +10, +5, +2.5, +1.25 V	Software	2	12	3 out, 8 in	Y	0	37
DAS-20	PC/XT/AT	16 SE/ 8 Dif	12	100 K	±10, ±5, ±0.5, ±0.05 +10, +1, +0.1 ±0.1 V	Software	2	12	3 out, 8 in	Y	2	41
DAS-1601	PC/XT/AT	16 SE/ 8 Diff	12	100 K	±10, ±1, ±0.1, ±0.02 +10, +1, +0.1, +0.02 V	Software	2	12	32	Y	1	47
DAS-1602	PC/XT/AT	16 SE/ 8 Dif	12	100 K	±10, ±5, ±2.5, ±1.25 +10, +5, +2.5, +1.25 V	Software	2	12	32	Y	1	47
DAS-16F	PC/XT/AT	16 SE/ 8 Dif	12	100 K	±10, ±5, ±2.5, ±1.25 +10, +5, +2.5, +1.25 V	Switches	2	12	4 in, 4 out	Y	1	51
DAS-1401	PC/XT/AT	16 SE/ 8 Dif	12	100 K	±10, ±1, ±0.1, ±0.02 +10, +1, +0.1, +0.02 V	Software	—	—	4 in, 4 out	Y	1	55
DAS-1402	PC/XT/AT	16 SE/ 8 Dif	12	100 K	±10, ±5, ±2.5, ±1.25 +10, +5, +2.5, +1.25 V	Software	—	—	4 in, 4 out	Y	1	55
DAS-16G1	PC/XT/AT	16 SE/ 8 Dif	12	70 K	±10, ±1, ±0.1, ±0.02 +10, +1, +0.1, +0.02 V	Software	2	12	4 in, 4 out	Y	1	51
DAS-16G2	PC/XT/AT	16 SE/ 8 Dif	12	70 K	±10, ±5, ±2.5, ±1.25 +10, +5, +2.5, +1.25 V	Software	2	12	4 in, 4 out	Y	1	51
DAS-16	PC/XT/AT	16 SE/ 8 Dif	12	50 K	±10, ±5, ±2.5, ±1.25 +10, +5, +2.5, +1.25 V	Switches	2	12	4 in, 4 out	Y	1	51
DAS-8/AO	PC/XT/AT	8 Diff or SE	12	4 K	±10, ±5, ±0.5, ±0.05, ±0.01 +10, +1, +0.1, +0.02 V	Software	—	—	4 out, 3 in	Y	3	58
DAS-8PGA	PC/XT/AT	8 Diff or SE	12	4 K	±10, ±5, ±0.5, ±0.05, ±0.01 +10, +1, +0.1, +0.02	Software	—	—	4 out, 3 in	Y	3	58
DAS-8 LT	PC/XT/AT	8 SE	12	4 K	±5	Fixed	—	—	4 out, 3 in	Y	3	81
µCDAS-16G	Micro Channel	16 SE/8 Diff	12	70 K	±10, ±1, ±0.1, ±0.02 +10, +1, +0.1, +0.02 V	Software	2	12	4 in, 4 out	Y	1	134
µCDAS-8PGA	Micro Channel	8 Diff	12	4 K	±10, ±5, ±0.5, ±0.05 ±0.01, +10, +1, +0.1, +0.02 V	Software	—	—	3 in, 4 out	Y	3	140
HIGH RESOLUTION												
DAS-HRES	PC/XT/AT	8 Diff	16	47.6 K	±10, ±5, ±2.5, ±1.25 +10, +5, +2.5, +1.25 V	Software	2	16	8 out, 8 in	Y	1	64
ADC-16	PC/XT/AT	8 Diff	16	16	±5, ±0.5, ±0.05 ±3.27, ±0.327, ±0.0327	Software	—	—	5 out, 2 in	Y	0	67
PCIP-DMM	PC/XT/AT	1	15	2.5	±200, ±20, ±2, ±0.2	Software	—	—		N	0	214
CHROM-1AT	PC/XT/AT	2 SE	14	350	+10, +5, +2, +1	Software	—	—	4 in, 4 out	N	0	77
DASCON-1	PC/XT	4 Diff	13	30	±2.047	Switches	2	12	12 I/O	N	0	70
SPECIAL PURPOSE												
PCIP-SCOPE	PC/XT/AT	2 Diff	8	20 M	±20, to ±0.02 in 10 ranges	Software	—	—	—	Y	0	218
DAS-TEMP	PC/XT/AT	32	0.1°	200	-25 to +105°C	Software	—	—	—	N	0	79
DAS-4	PC/XT/AT	8 SE	8	S/W Limited	±5	Fixed	—	—	4 out, 3 in	N	0	62
AT-CODAS	AT Only	16 SE/8 Diff	12	50 K	±5, ±2.5, ±1, ±0.5	Software	1	12	8 in, 8 out	Y	0	74
MCA-CODAS	Micro Channel	16 SE/ 8 Diff	12	50 K	±5, ±2.5, ±1, ±5	Software	1	12	8 in, 8 out	Y	0	137

ORIGINAL PAGE IS OF POOR QUALITY

PC/XT/AT/EISA Data Acquisition Boards

	AT-MIO-16X	AT-MIO-64F-5 / AT-MIO-16F-5	AT-MIO-16DH / AT-MIO-16DL	AT-MIO-16H-9 / AT-MIO-16H-25	AT-MIO-16L-9 / AT-MIO-16L-25	Lab-PC	PC-LPM-16	EISA-A2000	AT-A2150	AT-AO-6 / AT-AO-10	AT-DIO-32F	PC-DIO-96	PC-DIO-24	PC-TIO-10	AT-DSP2200
Computer Bus	AT	AT	AT	AT	AT	XT/AT	XT/AT	EISA	AT	AT	AT	XT	XT	XT	AT
ANALOG INPUT															
Channels*	16SE 8DI	64SE, 16SE 32DI, 8DI	16SE 8DI	16SE 8DI	16SE 8DI	8SE 4DI	16SE	4SE, SS	4SE SS	—	[illegible]	[illegible]	[illegible]	[illegible]	2SE SS
Max Sampling Rate (Hz)	100 k	200 k	100 k	100 k	100 k	75 k	50 k	1 M	51.2 k	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	51.2 k
Resolution (Bits)	16	12	12	12	12	12	12	12	16	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	16
Ranges (V)	±10, 0 to 10	±10, ±5, 0 to 10	±10, ±5, 0 to 10	±10, ±5, 0 to 10	±10, ±5, 0 to 10	±5, 0 to 10	±5, 0 to 10, ±2.5, 0 to 5	±5	±2.828	—	[illegible]	[illegible]	[illegible]	—	±2.828
Gains	1, 2, 5, 10 20, 50, 100	0.5, 1, 2, 5, 10 20, 50, 100	1, 2, 4, 8/ 1, 10, 100, 500	1, 2, 4, 8	1, 10, 100, 500	1, 2, 5, 10 20, 50, 100	1	1	1	—	—	[illegible]	—	—	1
ANALOG OUTPUT															
Channels	2	2	2	2	2	2	—	—	—	6 10	—	—	—	—	2
Resolution (Bits)	16	12	12	12	12	12	—	—	—	12	—	—	—	—	16
Digital I/O Channels	8	8	32	8	8	24	16†	—	—	8	32	96	24	16	—
Counter/Timers	3	3	3	3	3	3	2	—	—	—	—	—	—	10	—
RTSI	√	√	√	√	√	—	—	√	√	√	√	—	—	—	√
SOFTWARE															
LabVIEW for Windows	√	√	√	√	√	√	√	√	√	√	√	√	√	√	—
LabWindows for DOS	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
NI-DAQ	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
DAQWare	√	√	√	√	√	√	√	—	—	—	—	—	—	—	—
Measure	—	—	—	√	√	√	—	—	—	—	—	—	—	—	—
Third-Party	—	— √	—	√	√	√	√	—	—	—	√	—	√	—	—
Page Number	3-17	3-22	3-28	3-28	3-28	3-34	3-38	3-41	3-46	3-51	3-54	3-57	3-57	3-61	3-136

* SE — Single-Ended, DI — Differential, SS — Simultaneous Sampling

† 8 Channels In, 8 Channels Out

Table 1. Analog, Digital, and Timing I/O Boards for the IBM PC/XT/AT, EISA, and Compatibles

SELECTING THE RIGHT SOFTWARE

In order to determine which package is best for you, it is necessary to review your data acquisition system needs.

Acquisition What capabilities will be used?

- A/D, D/A, Digital I/O?
- GPIB or RS-232 instruments?
- Numbers of Channels?
- Maximum number of samples required?
- Maximum sample rate?

Analysis/Graphics What will you be doing with the data? Will the data be sent to another package for analysis? What formats are required? Will it be useful to analyze and plot the data where it is collected? Built-in analysis and graphics can provide rapid insight into results as the data is collected. Are hard copy graphics required? Do you need to incorporate custom routines? Do you want to generate custom graphics or reports?

User Interface Who will be using the system? The software is the interface to the system. If the system is to be used by inexperienced operators, it must be easy to use. Automation can help reduce operator errors. A menu driven interface can be used to guide the selections and save on repetitive tasks. A custom control panel can be the easiest interface of all.

Once you've answered these questions, the following software feature guide will help in choosing the most appropriate package for your application.

The hardware compatibility chart lists the boards and accessories supported by each software package. We are constantly adding new products and capabilities to this list, so please call if your board isn't supported.

SOFTWARE FEATURE SELECTION GUIDE

Feature	VIEWDAC	ASYST	EASYEST LX	EASYEST AG	LABTECH NOTEBOOK	Snap-Master	Control EG
Computer Requirements	386/486 w/ copr. & 6 MB RAM	PC XT/AT w/ copr. & 1 MB RAM	PC XT/AT w/ copr. & 2 MB RAM	PC XT/AT w/ copr. & 2 MB RAM	PC XT/AT	PC XT/AT w/ 4 MB RAM	PC XT/AT w/ 512 kB RAM
Operating Environment	DOS	DOS	DOS	DOS	DOS/ Windows	Windows	DOS
Max. A/D (D/A) Channels	100s	100s	160	160	256/16	Same as board	256
Max. A/D Boards	20	10	10	10	16	8	5 plus EXP's
Max. Acquisition Speed	Max. HW	Max. HW	Max. HW	Max. HW	Max. HW	Max. HW	10 I/O points per sec
Create macros, sequences, subroutines, procedures	Yes	Yes	Yes	No	Yes	Yes	No
Calls to standard programming languages	Yes	Yes	No	No	No	No	No
Integrated Analysis	Excellent	Excellent	Good	—	—	Excellent	Good
Max. Real-time display channels	Many [1]	Many [1]	8	8	50	100	16
Integrated graphics	Excellent	Excellent	Good	Good	Good	Excellent	Very Good
Custom User Interface	Yes	Yes	Yes	No	Limited	Yes	No
Graphic output to printers & plotters	Yes	Yes	Yes	Yes	Yes	Yes	Limited
RS-232 & GPIB support	Yes	Yes	No	No	Yes	Yes	No
Page Number	162	168	164	166	170	172	174

[1] Limited by screen resolution only.

ORIGINAL PAGE IS OF POOR QUALITY

HARDWARE/SOFTWARE COMPATIBILITY CHART

	VIEWDAC	EASYEST LX	EASYEST AG	ASYST	Control EG	Snap-Master	Labtech NOTEBOOK
ADC-16	C	C	C	C			
CHROM-1AT							Y
CTM-PER							
CTM-05	Y			Y			
DAC-02					Y		Y
DAS-16/F/G	Y	Y	Y*	Y	Y	Y	Y
DAS-1600/1400	C	C	C*	C	C	C	C
DAS-20	Y	Y	Y	Y	Y		Y
DAS-4						Y	Y
DAS-40	Y	Y	Y	Y			Y
DAS-50	Y	Y	Y	Y		Y	
DAS-58/SSH-58	C	C	C	C			
DAS-8/PGA/LT	Y	Y	Y*	Y	Y	Y	Y
DAS-8AO	C	C	C	C			
DAS-HRES	Y	Y	Y	Y		Y	Y
DDA-06	Y	Y	Y	Y	Y		Y
EXP-16	C	C	C	C	Y		Y
EXP-20	C	C	C	C		Y	Y
EXP-GP	C	C		C			Y
ISO-4					Y		
KPC-488.2	C			C		Y	Y
KPC-488.2AT	C			C		Y	Y
KPS-488.2	C			C		Y	Y
MB-01 & MB-02	C	C	C	C	Y		
PDISO-8							Y
PDMA-16/32	Y						
PIO/12/24/HV					Y		Y
SSH-4	C	C		C			
µCCTM-05	Y			Y			
µCDAS-16G	Y	Y	Y	Y		Y	Y
µCDAS-8PGA	Y	Y	Y	Y		Y	Y
µCDDA-04	Y	Y	Y	Y		Y	Y
µCPDISO-8							Y
µCPDMA-32							
µCPIO-12/24/72	Y	Y	Y	Y			Y
500A	Y	Y	Y	Y			Y
575	Y	Y	Y	Y			Y
576	Y			Y			Y

Y = Product is supported
C = Call for details
* = Start-up manual available

ORIGINAL PAGE IS OF POOR QUALITY

OMEGA Software Descriptions and Requirements

Software Title	Price	Data Acquisition Hardware Supported	Description	Part No.
Labtech® Notebook	$995	OM-700, OM-900, OM-1050, UCDAS-8-PGA, WB-800/815/802/817/820, DAS-8/16/20, PIO-12/24, CTM-05, DDA-06, DAC-02, CHROM-1, WB-AAI/FAI/ASC/AVO, OM-480/481 (*See pg. B-5 for add'l. hardware)	General Purpose Laboratory Data Acquisition, Control and Analysis Software	SWD-LTN
Real-Time Access™	$295	Requires Labtech Notebook	Labtech Notebook Accessory Program	SWD-RTA
Labtech Chrom	$495	No Hardware Supplied	Chromatography Analysis	SWD-LTC
CHROM+®	$645	Same as Acquire	Chromatography Analysis	SWD-LTP
WorkBench PC™	$995	WB-AAI/ASC/AVO/DIO/FAI, OMB-PER-488, DAS-8 DAS-16/16F	Icon-Based Data Acquisition and Control Software	SWD-WBP
SnapMaster	$995	CIO-AD16/16F, CIO-AD16JR, CTM-05, DAS-8PGA, DAS-16/16F/16G, DAS-50, DAS-HRES, PCL-718, PCL-818, UCDAS-8PGA	Data Acquistion, Analysis and Display Software	SWD-SNMA
IoCalc	$550	DAS-16/16F, PC-30, PC-60, PC-61, PC-66, OM-272, DP41, CDP-75, PHI-40/45	Real-Time Spreadsheet for DOS and OS/2	SWD-IOCA
Easyest LX	$995	DAS-8/8 PGA, UCDAS-8PGA, DAS-16/16F/16G, UCDAS-16G, DAS-20, DAS-50, CTM-05, UCCTM-05, PDMA-16, WB-800, WB-815, WB-820	Data Acquisition, Analysis and Graphics Software	SWD-EZLX
Unkelscope™	From $349	WB-800/815, DAS-8/16/16F, WB/AAI/ASC/FAI	High Speed Data Acquisition Software	SWD-US
Snapshot™	$495	WB-800, DAS-8/16/16F/16G, DAS8	Storage Oscilliscope Emulator Software	SWD-SNS
Snapshot with Snap-Calc™	$990	DAS-8, DAS-16, WB-800	Mathematical Analysis for Snapshot	SWD-SNC
Snap-FFT™	$495	No Hardware Supported	Frequency Analysis for Snapshot	SWD-SNF
LT/Control™	$3995	Same as Labtech Notebook, does not support OM-480 Series	Industrial Monitoring and Control Software	SWD-LCT

PRECEDING PAGE BLANK NOT FILMED

Software Title	Price	Data Acquisition Hardware Supported	Description	Part No.	Page
IPC-Expert	**$2250**	DAS-8, DAS-16/16F/16G, DAS-20, WB-800/802/815/817/820, OMEGA Workhorse, OM-1050	High-Speed Data Acquisition Software. Industrial Monitoring and Control Software	SWD-IPC	B-23
Slide Write Plus™	**$445**	No Hardware Supported	Presentation Graphics for Scientists and Engineers	SWP-SLP	B-25
STREAMER	**$275**	DAS-16/16F/16G, DAS-20, PDMA-16, PDMA-32, DAS-HRES, UCDAS-16G	High-Speed Disk Streaming Software	STREAMER	B-26
Acquire™	**$195**	DAS-16/16F, CHROM-1, WB-800, WB-AAI/FAI/ASC	Economical Data Acquisition Software	SWD-LAC	B-27
Control EG™	**$500**	WB-800/815/802/817/820	Data Acquisition and Control	SWC-CED	B-28
DADiSP™	**From $895**	No Hardware Supported	Data Analysis Software	SWD-DSP	B-29
Quality Analyst™	**$595**	No Hardware Supported	Statistical Quality and Control Charting and Analysis	SWA-NWA	B-31
Programming Tools	**$200**	No Hardware Supported	Real-Time Graphics, Measurement and Control Tools	SWV-IPC	B-32
Tech Graph Pad™	**$395**	No Hardware Supported	Data Plotting Software	SWP-TGP	B-33
Anti-Virus Plus	**$100**	No Hardware Supported	Virus Protection Software	SWU-ANTI	B-34
Plotz™	**$350**	No Hardware Supported	Graphics and Analysis Software	SWP-PLT	B-35
Atoms™	**$25**	No Hardware Supported	Atomic Reference Database	SWU-ATM	B-32
Units™	**$25**	No Hardware Supported	Linear Conversion Factors	SWU-UNT	B-32
MathCAD®	**$349**	No Hardware Supported	Mathematical Equation Solving	SWE-MCD	B-37
OMEGAMAC™ 232	**$595**	Most RS-232/422 Instruments	Macintosh Data Acquisition Software for RS-232/422 Instruments	SWD-MAC232	B-38
Work Bench Mac	**$995**	WB-FAI SE or M2, WB-AAI SE or M2, OMB-MAC2-488	Icon-Driven Data Acquisition and Control	SWD-WBM-2	B-39

SELECTOR GUIDE

		VIEWDAC	ASYST	KDAC500	EASYEST LX	Labtech Notebook	ASYSTA GPIB
Application	Primary Functions	Development Package for Data Acquisition Control Analysis & Graphics	High-Level Programming for Demanding Scientific & Engineering Applications TurboPascal	Data Acquisition Extension of BASICA QuickBasic QuickC TurboC & Graphics	Toolkit for General Purpose Data Acquisition Control Plus Analysis &	General Purpose Data Acquisition	Interacti Interface IEEE-48 Instrume Plus Analysis Graphic
	User Interface	Windowed	Language	Language	Icon	Menu, Icon	Menu
Computer	Platform	386, 486	XT & Up	XT & Up	XT & Up	AT, 386, 486	XT & U
	Math Coprocessor	Required	Required	Optional	Required	Optional	Require
	RAM, Minimum	4M	640K	640K	2M (4M for OS/2)	640K	640K
	Color Monitor	EGA or better	CGA or better	CGA or better	EGA or better	EGA or better	CGA or be
	Mouse	Required	Optional	Not Used	Required	Optional	Not Use
	Copy Protect Key	Yes	Yes	No	Yes	Yes	Yes
	Operating System	DOS	DOS	DOS	DOS	DOS, OS/2	DOS
	DOS-expanded Memory Manager	Required	Optional	Not Used	Required	Optional	Not Use
System Compatibility	Memory-Mapped System Supported	500A, 500P 575	500A, 500P 575	500A, 500P 575	500A, 500P 575	500A, 500P 575	—
	Maximum Systems	2	1	4	1	4	—
	Maximum Analog Inputs	608	304	608	160	250	—
	Use Together with Other Keithley MetraByte Hardware*	PIO, DAS boards	GPIB instruments	—	PIO, DAS boards	PIO, DAS boards	GPIB instrumen
	GPIB Support (IEEE-488)	Yes	Optional	No	No	Optional but limited	Yes
	Recommended Models	576	576	—	—	—	576
	RS-232 Support	Yes	Yes	Via Language	No	Optional	No
Software Features	Engineering Conversions						
	mX+b	Yes	Yes	Via Language	Yes	Yes	Yes
	TC Linearization	J K T E B R S	Polynomial	J K T E B R S	J K T E B R S	J K T E B R S	Polynomi
	RTD Conversion	Polynomial	Polynomial	100Ω Pt	Polynomial	Yes	Polynomi
	mv/v/full scale units	No	No	Yes	No	No	No
	PID Loops	Yes	Via Language	Via Language	Yes	Yes	No
	Real-Time Analysis	Yes	Yes	Via Language	Yes	RTA Option	—
	Post-Acquisition Analysis	Yes	Extensive	Via Language	Yes	Yes	Yes
	Calls to Languages	Contact Factory	Yes	Yes	No	No	No
	Real-Time Display Channels	> 100	> 100	16	16	50	—
	Custom Front Panels	Yes	Via PCX	Via PCX	Via PCX	No	—
	Run Time Versions	Yes	Yes	No	No	Yes	No
File I/O	Input File Formats	Binary, ASCII, ASYST, EASYEST ASYSTANT	Binary, ASCII, ASYST, EASYEST ASYSTANT	Any Via Language	ASCII, EASYEST	Binary, ASCII,	ASCII, DIF ASYSTANT
	Output File Formats	Binary, ASCII, ASSYST, EASYEST	Binary, ASCII, ASYST, EASYEST	Binary, ASCII, ASYST, Lotus DADiSP	ASCII EASYEST	Binary, ASCII, Lotus, Character	ASCII, DIF ASYSTANT
	Graphics Output Files	PCX	PCX	Via Language	PCX	No	No
	Graphics to Plotter	Contact Factory	HPGL	Via Language	HPGL	No	HPGL

CALL DRIVER INTERFACE

The standard Call Driver is designed to be used from Interpreted BASIC, Compiled BASIC or QUICKBASIC. The calls are a collection of functions that are accessed from a BASIC program, each through a single line Call statement. The various modes of the call routine select all of the functions of each board, format and error check data and perform frequently used sequences of instructions. To use the Call commands, simply select the applicable call routine and pass the appropriate parameters. The driver handles all low level hardware manipulation. A consistent set of defined call command names are used in the new DAS drivers so you can write board-independent programs and easily change from one board to another. The Call commands allow a great deal of flexibility and allow background data acquisition. Most commands execute faster using the Call Driver versus the File I/O Driver. Many of Keithley MetraByte's data acquisition boards have optional Pascal, C and Fortran (PCF) interfaces with similar functionality. Some of the more simple boards include the PCF capability in the price of the standard board.

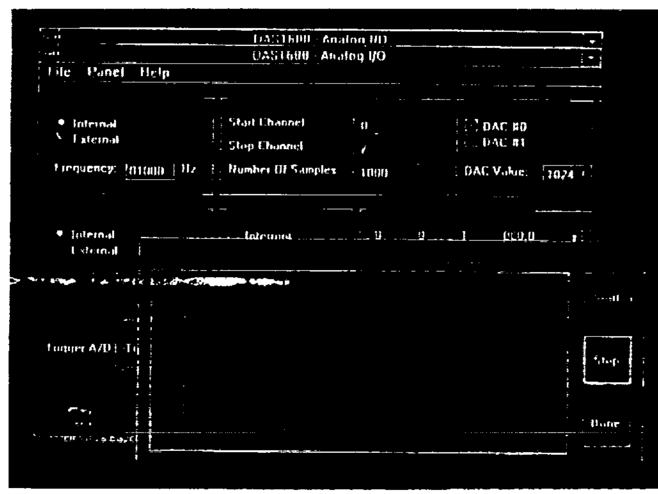
The following example shows the Call Driver interface using the BASIC language. This program performs an analog to digital conversion on channel 4 of the ADC-16.

```
10                                              ' Configure and initialize
                                                  the ADC16
20  CALL ADC16DevOpen(Filename$, NumOfBoards%,ErrFlag%)
30  IF ErrFlag% < > 0 THEN STOP                  ' Stop if Configuration Error
40  Start% = 4                                   ' Set up channel number
50  Gain% = 0                                    ' Set up gain
60  CALL KAD(Start%, Gain%, ErrFlag%)            ' Get a reading
70  IF ErrFlag% < > 0 then PRINT
    "AD Error Occurred"
```

Call Driver interface gives the highest performance.

WINDOWS 3.0 DLL INTERFACE

Microsoft Windows Dynamic Link Libraries or DLL standard allows you to use any language that supports the DLL construct, including Microsoft's Visual Basic and C for Windows and Borland's C++ and Turbo Pascal for Windows. Using the DLL driver option, all the features of Microsoft Windows are accessible. This includes running multiple programs and using extended memory all through the consistent graphical user interface of Windows. Microsoft Windows lets you set up and run your data acquisition program and automatically transfer collected data to another application through the standard Dynamic Data Exchange (DDE).



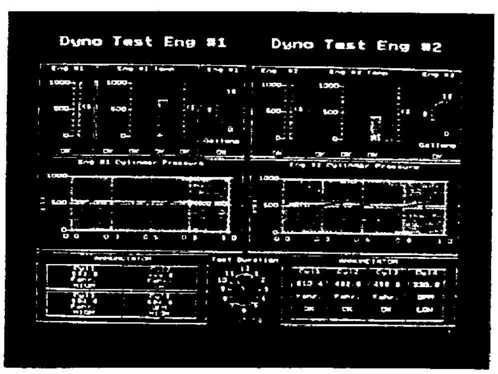
The DAS-1600 and Visual Basic make creating custom interfaces simple

A typical example is collecting temperature measurements with an ADC-16 and transferring the data to a spreadsheet package, such as Excel, for automatic scaling and plotting. All features and commands available in the Call Driver interface are accessible in the DLL interface. The same command set is used for both DOS and Windows based programming. Windows also gives access to new languages such as Visual Basic. A Visual Basic example is included with the DLL Interface to show how easy it is to set up a control panel for your DAS board and to acquire and graph data.

DIRECT REGISTER I/O

The direct register I/O programming should be used only if you have a high level of programming expertise and understanding of your PC's architecture and the PC's Peripheral Controllers (the 8259 Interrupt Controller, the 8237 DMA Controller) or if you are programming a very simple board such as the PIO-12 digital I/O interface. This method can be used with most languages. Direct Register I/O has the advantage of being the fastest method to communicate with the board. A disadvantage with programming in BASIC is that BASIC has no Interrupt or DMA processing functions, so background data acquisition is not available when using this method. Our user manuals provide the details needed for implementing register programming. This can be used with operating systems, such as UNIX, that are not presently supported by our drivers.

The Driver Table to the right illustrates driver support for our data acquisition boards. A PCF package includes Pascal, C and Fortran Call drivers and a user manual for the supported board. The Advanced Software Option for a board includes the File I/O Driver, Pascal and C Call Drivers, the Windows 3.0 DLL, comprehensive user manuals and examples.



Programming Tools help create custom graphics.

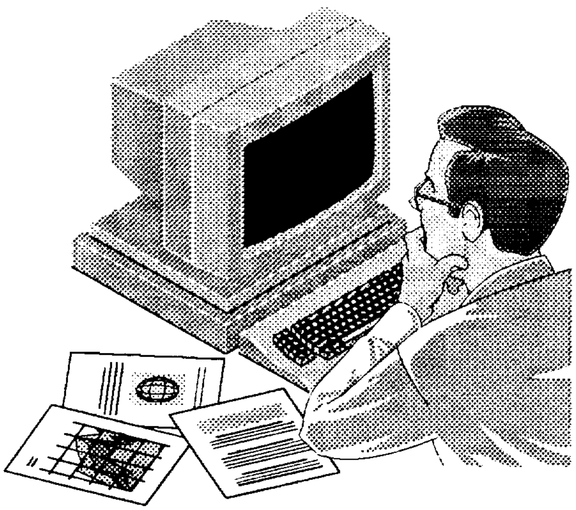
DRIVER AND UTILITY SUPPORT

	POP UP	FILE I/O	Basic Call	DLL	Register I/O	STREAMER	PCF Calls
ADC-16	√	√*	√	√*	√		√*
CHROM-1AT			√		√		
CTM-PER			√		√		√
CTM-05			√		√		√
DAC-02					√		
DAS-16/F/G	√	√	√	√D	√	√	√
DAS-1600/1400	√	√*	√	√*	√	√*	√*
DAS-20			√		√	√	√
DAS-4			√		√		
DAS-40			√		√	√	√
DAS-50	√	√*	√	√*	√		√*
DAS-58	√	√*	√	√*	√		√*
DAS-8/PGA/LT			√	√D	√		√
DAS-8/AO			√	√D	√		√
DAS-HRES			√		√	√	√
DDA-06					√		
KPC-488.2		√	√	√			√
KPC-488.2AT		√	√	√			√
KPS-488.2		√	√	√			√
PCIP Family	√	√					
PDISO-8					√		
PDMA-16/32			√		√	√	√
PIO-12/24/HV					√		
µCCTM-05			√		√		
µCDAS-16G			√		√	√-	√
µCDAS-8PGA			√		√		√
µCDDA-04			√		√		
µCPDISO-8					√		
µCPDMA-32			√		√		√
µCPIO-12/24/72					√		

* = Part of Advanced Software Option
D = DriverLINX support
‡ = If used as a DAS-16
- = µCSTREAMER

ORIGINAL PAGE IS OF POOR QUALITY

3.0 Data Acquisition System Hardware



Why we selected CIO–AD16Jr and DriverLINX combination

To design a Data Acquisition System for a welding Robot. Signals from various sensors of the robot are acquired and after proper signal–conditioning and isolation (if required) are fed to the data acquisition card. There are basically three types of input signals , analog signals, digital signals and the counter–timer signals. The analog signals are digitized after proper conditioning to take full advantage of the input range. The digital signals are raised to TTL levels and fed to the digital I/O port of the Data Acquisition card. Based on the selection criterion as described above, a device driver is selected (DriverLINX) to communicate with the Data Acquisition Board (CIO AD–16). Programming for the Driver and the File I/O command interface is done in Visual Basic. Acquired data can be stored to a file in binary or ASCII format for importing into commercial and custom analysis packages. The analysis package is also programmed in the Windows environment using Visual Basic.

3.1 Sensors/Signals

Data	*Sensor*	*Signal*
Position	Optical Encoders	Digital Pulses
Arc Current	Hall–effect	4 - 20 mA
Arc Voltage	Circuit	Voltage
Wire–feed	Encoders	Digital

These are tentative signals on the basis of which, the data acquisition card is selected.

3.2 The Data Acquisition Card

CIO-AD16 is multifunction analog and digital I/O board. It is small in size, acquires data very fast (110 KHz for CIO-AD16Jr and 330KHz for CIO-AD16Jr - AT). It is a plug-in data acquisition board and can be installed on any IBM PC/AT or compatible. It can turn the personal computer into high-speed data acquisition system.

CIO-SSH16 is a 16-channel simultaneous sample and hold accessory board. It acts as a front-end signal amplification and capture for CIO-AD16 series of analog input boards. This board provides two major functions. Sixteen differential amplifiers have individualswitch selectable gains of 1, 10, 100, 200, 300, 500, 600, 700, and 800 providing very flexible amplification for individual signals. After amplification, each channel has a sample and hold which is controlled by the CIO-AD analog input board. In applications where, a number of signals must be analyzed and compared, such as high speed transient analysis and spectrum analysis, a channel to channel skew may be unacceptable. The CIO-SSH16 eliminates the channel to channel skew associated with multiplexed A/D inputs. The data sheets for CIO-AD16 and CIO-SSH16 are attached.

3.3 Software

There are various software packages available for data acquisition, analysis and display. These packages are very expensive depending on various features it offers. Most of the features are extra for a specific application and still may not be to the complete satisfaction of the user. For a specific application, it is best to design a custom-made software which exactly meets the requirements. To help us communicate with the data acquisition boards at high level, we use Drivers. This saves us the trouble of writing directly on board registers.
For our application, we use windows 3.0 Dynamic Link Libraries (DLL) interface. Microsoft Windows DLL standard allows the use of any languages that supports DLL construct. For our application, we use Microsoft's Visual Basic. Using the DLL driver

option, all features of Microsoft Windows are accessible. This includes running multiple programs and using extended memory all through the consistent graphical user interface of Windows. Microsoft Windows gives us flexibility and lets us set up and run the data acquisition program and automatically transfers collected data to another application through the standard Dynamic Data Exchanga (DDE).

For our application, we are using DriverLINX as our Driver. It has following features:

- Language Independent dynamic link libraries (DLLs).
- Multitasking and multiuser capabilities.
- More than 70 high level functions supported.
- Allows maximum data acquisition speeds.
- Comprehensive on-line help.

DriverLINX allows us to use CIO-AD16Jr with MS Windows 3.0 environment. It provides us high-level interface with data acquisition board. This driver contains all the intelligence necessary to to manage the details of data acquisition tasks, insulating the developers program from hardware and implementation strategy. Applications communicate with DriverLINX by passing a "service request" that contains the specifications for the data acquisition task.

DriverLINX supports all functions of CIO-AD16Jr including analog input and output, triggering, gain setting and DMA transfers. It can support upto 6 boards and 10 concurrent tasks. Special version of DriverLINX(DriverLINXVB) is available to program in Visual Basic.

4.0 The Data Acquisition System Software

This software incorporates window programming techniques and operates in Microsoft's Windows 3.0 environment. DOS programs are written sequentially, i.e. one event follows the other. In DOS programs, controls goes down the list of statements, more or less the order which the programmer designed. However, windows is different.

An application under Windows typically present all possible options (in the form of visual objects) on the screen for the user to select for themselves. In this way, it represents entirely new kind of programming -- event-Driven, and object-oriented programming. That is to say that, a programmer is no longer completely responsible for the flow of program. Rather the user is. The user selects among all the options present to them, and it is upto the program to respond to them. The code is specifically designed to respond to a particular event called-on by the user. Our program will typically be the collection of code sections like this, one after the other. That is how event driven program works. Besides being event driven, window programming is also object-oriented. That is easy enough to see on screen: Just pick up an object or a paint brush on the screen and move it around. This corresponds closely to what's called object oriented programming. This type of programming breaks a programming up into discrete objects, each of which has its own code and data associated with it. In this way, each of the objects can be somewhat independent from others.

Window environment is extremely friendly to the user, but programming Windows was often excruciating -- until recently. Visual Basic environment is very friendly even for the programmer. Visual Basic is the new BASICA or GW-BASIC. There are three major steps in writing application in VIsual Basic. They are :

- Draw the Window(s) you want.
- Customize the properties of buttons.
- Write the code for associated events.

In first step, complete with buttons and menus – this is where Visual Basic really shines. Before, it was tedious process to design the appearance of the windows, where the buttons would go, how large it would be and all types of other considerations. Adding or removing features were also difficult.

Under Visual Basic, the whole process has been extraordinarily easy. Just like using paint–brush in windows. Visual Basic allows us to simply draw the windows we want, as well as the buttons, boxes and labels we want. Adding or removing buttons or boxes wo rks just like it would in a paint program. There is no difficult programming involved. The next step involves customizing the properties of what we have drawn; for example, we might give the window or a button a certain caption, or change its color (or even whether or not it is visible). Finally, writing a code that responds to events we consider significant. This is how it works in outline. In other words – Visual Basic is like Window programmer's dream.

5.0 CONCLUSIONS AND RECOMMENDATIONS

The specified welding parameters have been evaluated and a data acquisition scheme was developed to support weld modeling and feedback control development. The initial data acquisition requirements for VPPA welding were determined.

The Mid-South VME-based PC/DOS system was evaluated and found to be unacceptable for this task. Written requirements for a new PC/DOS/Windows system were prepared and the specified hardware and software were purchased by NASA.

Overall software requirements were determined and the combination of DOS 5.0, Windows 3.1, Visual Basic 1.0, and Driver Lynx 1.0 was recommended. All of these are commercially-developed software products.

The Preliminary software and hardware was demonstrated. The computer hardware was not received by NASA in time for complete integration of the hardware and software. I recommend that this work be further developed in a follow-on task.

A1.0 VISUAL BASIC

This section briefly defines the important features of the Visual Basic
To create an application in Visual Basic, follow this suggested sequence:

1. Create a new project (or use the new project created when you start Visual Basic) to organize the parts of your application.
2. Create a form for each window in your application.
3. Draw the controls for each form.
4. Create a menu bar for the main form.
5. Set form and control properties.
6. Write event procedures and general procedures.
7. Save your work.
8. Debug your code.
9. Create an executable file to turn the project into an application.

A1.1 Menus

File Menu Controls Visual Basic projects and files.

Edit Menu Alters Form window and Code window contents, sets up DDE links, and controls use of the drawing grid.

Code Menu Creates, alters, displays or prints code. Searches for specific text.

Run Menu Controls application execution and provides debugging tools.

Window Menu Opens or closes Visual Basic windows: the Color palette, the Immediate window, the Menu Design window, the Project window, and the Toolbox.

Help Menu Provides access to Help topics, the Tutorial, and version information for Visual Basic.

A1.2 Event Procedures

The code you attach to a form or control is called an event procedure. Every form and control has a set of predefined events that it can recognize. You attach event procedures only for events to which you want a form or control to respond.To attach an event procedure to a form or control:

1. Double-click a blank part of the form to open the form's Code window. Or, to attach code to a control, double-click the control.
2. In the Procedure box, select the event to which you want to attach code.
3. Enter the code you want in the template provided, following basic guidelines for entering and editing code and declarations.
4. Repeat steps 2 and 3 as necessary to attach additional event procedures to the item.

Syntax

Use this syntax when writing an event procedure:

```
Sub ItemName_EventName (arguments)
        local variable and constant definitions
        statements
End SubNote
```

You can also select controls or the form itself from the Object box in the Code window. The information displayed in the Procedure box changes to reflect the predefined events for the selected item. Bold text in the Code window's Procedure box indicates event procedures you have attached to a form or control. Instead of using the template provided for you by Visual Basic, you can also create a new procedure by typing Sub ProcedureName in the Code window. If you change the CtlName of a control after attaching a procedure to it, you must also change the name of the procedure to match the name of the control. Otherwise, Visual Basic won't be able to match the control to the procedure. You can find the procedure by selecting (general) from the Code window's Object box and then selecting the procedure from the

Procedure box. For a list of the events that apply to forms and each type of control, see the topics under Properties, Events, and Methods index. For a list of all Visual Basic events, see the Events index.

See Also
Help:
Creating a General Procedure
Tutorial:
"Working with Visual Basic"
Programmer's Guide:
Chapter 8, "Attaching Code"
Chapter 9, "Language Elements"
Chapter 10, "Responding to Commands"
Chapter 11, "Getting Information from the User"

A1.3 Use of code window

The Code window is used to write, display and edit code. Each form or module has one code window. You can open as many Code windows as you want, so you can easily view the code in different forms and modules and copy and paste between them.

To open a Code window from the Project window, select a form or module name and click the View Code button. Click the View Form button to see the form.

To open a Code window from a Form window, double-click a control or the form itself.The Code window consists of:The Object box, located at the upper-left corner of the Code window. It lists all the forms and controls in your project.The Procedure box, located at the upper-right corner of the Code window. If you are editing form code, it lists all the events Visual Basic recognizes for the form or control displayed in the Object box. When you select an event, the event procedure associated with it or a code template is displayed in the bottom part of the Code window. If (general) is displayed in the Object box, the Procedure box lists all of the general procedures that have been created for the form. If you are editing module code, the Procedure box lists all of the general procedures included in the module. In either case, the procedure you select in the Procedure box is displayed in the bottom part of the Code window. The Split bar, located across the window, below the title bar and at the top of the vertical scroll bar. Dragging this bar down splits the Code window into two horizontal panes, each of which scrolls separately. This enables you to view different parts of your code at the same time. The information in the Object box and Procedure box applies to the code in the pane that has the focus. Dragging the bar to the top of the window closes a pane.

Note

If you need help on syntax for functions, statements, properties, events, or methods while working in the Code window, type the keyword or the property, event, or method name, and press F1.

See Also Help:

Attaching an Event Procedure to a Form or Control

Entering Declarations in Code

Guidelines for Entering and Editing Code

Tutorial:

"Working with Visual Basic"

Programmer's Guide:

Chapter 8, "Attaching Code"

A1.4 Using Microsoft's Visual Basic:

This document contains release notes for version 1.00 of Microsoft Visual Basic for Windows Version 3.0, and later. Information in this document is more current than that in the manuals. Information in online Help may also be more current than that in the manuals. Microsoft revises its languages documentation at the time of reprinting, so some of the information in this online file may already be included in your manuals. Content

Part Description

1	Software Installation Information
2	Notes for "Microsoft Visual Basic Programmer's Guide"
3	Notes for "Microsoft Visual Basic Language Reference"
4	Notes for Tutorial5 Miscellaneous Notes and Tips

A1.4.1 Part 1: Software Installation Information

Before installing Visual Basic you should make backup copies of all the distribution disks. Do not write-protect the distribution disks you use to install Visual Basic. If you do, Visual Basic cannot be successfully installed.Also ensure that the Windows directory, including drive letter, is in your PATH. If the drive letter is not included in your PATH statement, Visual Basic will be unable to determine where Windows resides. SETUP.EXE is a Windows application; that is, it is run from Windows, rather than from the DOS prompt. SETUP.EXE will only run in Windows Standard or Enhanced mode. It will not run in Real mode. You can determine how Windows is configured on your computer by choosing About from the Help menu in the Program Manager.To install Visual Basic, use Program Manager or File Manager to start SETUP.EXE as you would any other Windows application. For example, if you are installing from A:

1. From the Program Manager File menu, choose Run.
2. In the Run dialog box, type A:SETUP and choose OK.

Most of the files on these disks are compressed and must be expanded before they can be used. For Visual Basic to work properly, you must install the files using SETUP.EXE. You cannot simply copy the files to your hard disk.

A1.4.2 Part 2: Notes for "Microsoft Visual Basic Programmer's Guide"

Page	Section\Note
117	*An Application that Adds and Deletes Menu Commands* Change the last item in the table near the bottom of the page so

that CtlName is SepBar and Index is blank. At the bottom of the table add another line with a blank caption, AppName as the CtlName, indented once, with 1 as the Index.
Insert a sentence immediately following the table that reads:
Turn off the Visible property for AppName by toggling the Visible check box. Change "AppName" in the sentence following the table to "SepBar".

118 Delete the sentence that begins "Therefore, you cannot..." from the second paragraph on the page. Add the following line of code just above the End Sub for the AddApp_Click procedure:
AppName(LMenu).Visible = -1

170 *Creating a Control Array*

In the example code at the bottom of the page Format(I) should be Format$(I).

223 *Displaying the GroupChoice Form*

The last sentence on the page should read
: Then create a list box (named GroupList) large enough to hold six or more items.

353 *Chapter 22 – Communicating with Other Applications*

This chapter discusses the links that enable applications to exchange information through dynamic–data exchange (DDE). The correct names for the two kinds of DDE links are "hot link" and "cold link." On pages 360, 370, and 371 cold links are incorrectly referred to as "warm."

360 *LinkTimeOut*

The note near the bottom of the page incorrectly indicates that the Alt key is pressed to interrupt pending DDE operations. Actually, the Esc key is used to interrupt DDE operations.

364 *LinkExecute*

The example code shown will work correctly, however to be syntactically correct, it should appear as follows:

```
Sub Form_LinkExecute (CmdStr As String, Cancel As Integer)
   Const FALSE = 0, TRUE = Not FALSE
   If CmdStr = "[Quit]" Then
               Cancel = FALSE
               End
   Else
               Cancel = TRUE
   End If
End Sub
```

386 *Null Pointers*

The call to the FindWindow DLL routine near the top of the page should read as:

```
hWndExcel% = FindWindow%(ByVal 0&, ByVal "Microsoft Excel")
```

Insert the following paragraph immediately following the statement shown above:

The use of ByVal when passing a string is necessary because the data type of that argument was declared as Any. Including ByVal when passing a string declared as Any causes Visual Basic to convert the string to the null-terminated form expected by most DLL routines.

386 *Properties*

The Lib clause in the external function Declare statement should say "GDI" instead of "User."

A1.4.3 Part 3: Notes for "Microsoft Visual Basic Language Reference"

Page	*Section\Note*
9	Table 3 - Properties by Programming Task
	In the Windows category at the bottom of the page, the property for "Get handle for form" should be hWnd, not hWin.
27	*AutoRedraw Property*
	The Note should include the following paragraph: When you minimize a form whose AutoRedraw property is set to False (0), ScaleHeight and ScaleWidth are set to icon size. When AutoRedraw is set to True (-1), ScaleHeight and ScaleWidth remain the size of the restored window.
31	*BorderStyle Property*
	In the Description section change the word "picture" to "text." Add the following paragraph at the end of the Remarks section: Because of appearance, the BorderStyle for forms with a menu can only be set to Sizable (2) or Fixed Single (1). Setting the BorderStyle property to None (0) or Fixed Double (3) forces the BorderStyle property to Fixed Single (1).
147	*Icon Property*
	In the Description section change "read-only" to "read-write." Change the note to read as follows: For a form icon to be functional, the BorderStyle property must be set to either 1 (Fixed Single) or 2 (Sizable). The MinButton

property must be set to True (–1).
At run time, you can assign an object's Icon property to another object's DragIcon or Icon property. You can also assign an icon returned by the LoadPicture function. Doing this assigns an empty (null) icon, which enables you to draw on the icon at run time.176

LinkExecute Event

The default value for the Cancel% argument to the LinkExecute event
is True (–1). This is done so that if no LinkExecute procedure is written, Visual Basic properly returns a negative acknowledgement to any application that attempts to send a string to Visual Basic to be executed.

188 *List Property*

The last sentence in the Description should be changed to read:
The List property is not available at design time; it is read–only for drive, file, and directory list boxes and read–write for combo and list boxes.

285 *SetData Method*

The parentheses shown in the syntax example should be removed.

A1.4.4 Part 4: Notes for Tutorial

Tutorial Screen Conflicts

Some Windows programs that run in the background and automatically perform some action on the screen may behave unpredictably when the Visual Basic Tutorial is running. For this reason, we recommend that you turn off or unload screen savers and background clock–type programs before running the Tutorial.

A1.4.5 Part 5: Miscellaneous Notes and Tips

Using Frames

If you plan to group controls on a form using a frame, draw your frame first, then draw the controls in the frame. This allows you to reposition the frame and the controls it contains as a single unit rather than having to move each part separately.

Deleting or Renaming Controls

When you delete or rename a control for which you have written event procedures, the event procedures themselves are not deleted. All such event procedures become general procedures with their names preserved. If you create a new control of the same name, those general procedures will once again become attached to that control. If you rename those general procedures to match the name of an existing control, they too will become attached to that control. Note that while the event procedures are restored, the value of any properties you previously assigned to the deleted control are lost.Displaying Modal Forms from the Immediate Window

Forms cannot be displayed modally using the Show method in the Immediate window. You can, however, call a procedure from the Immediate window which contains a Show method to display a modal form. There are no restrictions on the display of non-modal forms.

Design–Time Dynamic Data Exchange (DDE)

If you establish a design–time DDE link (either as a client or as a server) between another application and a Visual Basic text box, changing any property which causes the text box to be destroyed and recreated will terminate the link. For example, changing a text box from single line to multiline terminates a design-time DDE link; the Visual Basic developer must re-establish the link if it's still needed.

Terminating DDE Links During Form_Unload

You must terminate all DDE links before you close a form that contains any controls involved in the link.

Submenu Visibility

If you have a menu with submenu items, at least one submenu item must always be visible, i.e., the Visible property must be set to True (-1).Calling DLL Routines by Ordinal Number Some DLLs export their routines by ordinal number rather than by name. To call one of these DLL routines, you must declare it with an alias string that includes the number sign character (#) followed by the ordinal number. For example, to declare the routine with ordinal number 234: Declare Sub AnyRoutine Lib "AnyDLL" Alias "#234" (ByVal Var As Long)

Default WindowState

Whatever WindowState (minimized, restored, or maximized) a form is when Visual Basic goes to Run mode may become the new default WindowState. If you want your forms to retain a specific WindowState you must either close the form before running or explicitly set the WindowState in the Form_Load event. However, this last technique prevents the application from being run in a state different from the state explicitly set.

Custom Controls

You must remove all instances of a custom control from project forms before removing the custom control file from the project.If you're replacing or updating a custom control file with a newer version, you should not remove (using File Remove File) the old version and then add the new version (using File Add File). Instead, simply copy the new version over the old version and reload the entire project.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle	5. Report Date
Welding Process Modeling and Control	April 1993
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Peter L. Romine Jinen A. Adenwala	10. Work Unit No.
9. Performing Organization Name and Address University of Alabama, Huntsville Electrical and Computer Engineering Dept. Huntsville, AL 35899	11. Contract or Grant No. NAS8-38609 DO#24
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546-0001 George C. Marshall Space Flight Center	13. Type of Report and Period Covered Final 26 Feb 1992 - 16 April 93
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This is the final technical report for the period 26 February 1992 through 16 April 1993.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
	Unclassified

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	41	