NASA-CR-198833

FINAL TECHNICAL REPORT

Automated Data Acquisition Technology Development
Automated Modeling and Control Development

7 June 1993 through 6 June 1994
Contract Number NAS8-38609
Delivery Order #82

Prepared for:

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

16 May 1995

by

Peter L. Romine

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

TABLE OF CONTENTS

1.0 INTRODUCTION

This report documents the completion of, and improvements made to, the software developed for automated data acquisition and automated modeling and control development on the Texas Micro rackmounted Pcs.

This research was initiated because, a need was identified by the Metal Processing Branch of NASA Marshall Space Flight Center, for a mobile data acquisition and data 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 strickly for use of data acquisition and data analysis. In addition to the data acquisition functions described in this thesis, WMS also supports many functions associated with process control.

The hardware and software requirements for an automated acquisition system for welding process parameters, welding equipment checkout, and welding process modeling were determined in 1992. From these recommendations, NASA purchased the necessary hardware and software.

The new welding acquisition system is designed to collect welding parameter data and perform analysis to determine the voltage versus current arc-length relationship for VPPA welding. Once the results of this analysis are obtained, they can then be used to develop a RAIL function to control welding startup and shutdown without torch crashing.

1.1 Overview

Data Acquisition is the process of collecting information for analysis from the real world. This information is collected, stored and analyzed using a computer. The use of a computer automates the data acquisition process, enabling the collection of more information in less time. For slow processes, the data can be analyzed as it is being acquired (in real time) or otherwise, it can analyzed in a postmortem fashion. Creating a cost effective, high performance, automated data acquisition/analysis system for these problems is a complex task. It demands complete knowledge of the system requirements along with the market survey for available alternatives which enables us to decide on the optimum price/performance balance. Fortunately, recent advances in personal computer hardware and software has made it possible to develop a low-cost, high performance data acquisition system.

A portion of this research focuses on outlining techniques that can be used to create a low-cost real time data acquisition system utilizing the existing technology. Specifically, this thesis concentrates on the design and development of personal computer based (PC) Plug-in Data Acquisition System (DAQ) to facilitate acquisition and testing of process signals from a Plasma Welding Robot that is currently under development by NASA Marshall Space Flight center in Huntsville, AL. The research includes development of various user-friendly Graphical User Interfaces (GUI) in Windows environment using the object-oriented/ event driven features of a high level visual programming language. This application interacts with the hardware for real-time data monitoring and for subsequent data analysis. This research highlights the need for matching the parameters of the real world system and choosing the hardware/software environment while allowing for seamless interaction with human operators.

1.2 Data Acquisition

Most real-world data are not in the form that can be directly recorded by the computer. The data to be measured can be of various types, i.e. mechanical moments, electrical changes, temperature variations, etc.. These changes are measured using sensors or transducers which convert the physical quantity into an electrical quantity. These enable the data to be conditioned by electronic instrumentation, which operates on analog signals or waverers. These signals are continuos and monotonic. They vary continuously over a specified range (e.g. variations between +5V and -5V). These signals have to be digitized in order to be sent to the computer and hence be recorded. This is achieved using analog-to-digital converters (ADC). The resulting digital signal is usually an array of digital values of known range and separated by fixed time intervals (sampling interval). Additional channels can be added to the ADC via a multiplexer. This is an economical approach where all signals do not need to be simultaneously monitored. Because of the cost constraints, research was focussed towards developing a PC-based process monitoring system. Data Acquisition using personal computers and plug-in boards is widely used, but there are some applications where it is not appropriate. For example, external systems are preferred for systems requiring a large number of input channels, as it is not very practical to connect large number of wires to the PC. Also external systems have an advantage that their power supplies are built specifically for data acquisition, making it more desirable in instances requiring accuracy greater than 16-bits or extremely fast measurements. Important system parameters such as sampling speed, accuracy, resolution, amount of data, multi-tasking capabilities and the required data processing and display are used to make the decision on selecting the data acquisition system. Software is also an important factor for the design of the data acquisition system. Inefficient software can waste the usefulness of most efficient hardware (and vice-versa). Data acquisition software controls the collection, analysis and the display of data. With appropriate software, computers can process the acquired data, and produce outputs in the form of tables or plots. Some softwares are also used for automated industrial process controls. We will limit our discussion to Data acquisition and Data monitoring using personal computers. We will also discuss some of the commercially available software packages which can collect, analyze and display data graphically, using little or no programming. This software allows the user to concentrate on the application instead of worrying about the mechanics of acquiring data or how to plot a set of Cartesian coordinates.

1.3 Data Acquisition Techniques Using Personal Computers

Because PCs are so popular and have evolved into a powerful, cost-effective computing platforms, an increasing number of PCs are being used in laboratories and factories in data acquisition and control applications. The evolution of A/D converter technology and signal conditioning accessories and has helped plug-in data acquisition boards become the most popular instrumentation choice. IEEE 488 and VXI instruments are generally used for more sophisticated measurements. RS-232 is quiet popular for simple instruments and for controlling remote data acquisition subsystems. One can mix and match the types of instruments in a given subsystem to take full advantage of the options available to us. Various software tools available, make each type of instrumentation equally easy to use and integrate into the system. Careful selection of data acquisition components is necessary to meet the cost and performance requirements of each individual application.

The performance-to-cost ratio of the PC combined with the improved memory usage, better GUI, and the multitasking abilities of modern PC windowing environments, make PCs a very attractive combination for the data acquisition applications. The fastest growing method of acquiring data into the PC is the use of plug-in data acquisition boards. These boards continue to improve in both speed and accuracy. In fact boards with 16-bit resolution and high performance instrumentation are common. High accuracy signal conditioning systems enhance the capabilities of the plug-in data acquisition boards.

1.4 Window-based DAQ, Challenges and Prospects

Windows environment present some challenges for the developer trying to squeeze maximum performance from DAQ boards. To succeed, the developer must use some special techniques to optimize the speed of hardware interrupts and DMA. The non-preemptive multitasking of windows and the DMA management of interrupts hamper windows' real-time performance. However with some extra efforts, Windows can deliver excellent performance using todays' data acquisition hardware. The main challenge facing developers of real-time, hardware-interfacing Windows application is guaranteeing performance in non-preemptive, or corporative multi-tasking environment of

Windows. This limitations can be overcome by using hardware interrupts to gain real-time response from the CPU. Hence for Windows, the preemptive nature of hardware interrupts can be used to bypass the nonpremptive Windows environment.

1.5 Data Transfer Between DAQ and PC

Signals are digitized by the DAQ board's A/D converter and the resulting values stored in an on-board FIFO memory buffer. The buffer temporarily stores the data until the user moves it into the computer memory. Typically the user transfers data from the FIFO buffer to the computer memory using the programmed I/O interrupt or the DMA technique.

With the programmed I/O technique, the CPU repeatedly polls the DAQ-board registers until data is available in the FIFO buffer. The CPU then transfers the data in the computer memory. Engineers generally use the programmed I/O technique for less demanding DAQ applications that do not require precise hardware timings or very high transfer rates.

With the interrupt technique, the plug-in data acquisition board asserts a hardware interrupt whenever the ADC performs a conversion and data is available in the FIFO buffer. The CPU responds to the hardware interrupt by branching to the pre-programmed interrupt service routine (ISR) that includes instructions for the CPU to transfer data to the computer memory. Alternatively, the boards can also be configured to assert interrupts only when FIFO buffer is half full, which makes more efficient use of the CPU time. In either case, throughput will be limited by the CPU response time to the hardware interrupts.

The DMA technique uses the DMA controller, to move data from DAQ board to memory and vice-versa. A DMA controller manages several channels, each of which can be programmed to perform a sequence of transfers. Using the combination of the DMA and the interrupt techniques, plug-in DAQ boards can acquire sampled analog data at the rates of upto several million samples directly into PC memory. However, such data acquisition system requires efficient management of system resources.

2.0 THE WELDING ROBOT

Welding is a process of fusing metals together. Before describing the details of robot welding, let us look at how a human-being performs the welding process. Initially, the workpiece is cleaned of all contaminants, such as rust, dirt, oil, grease and paint. The welder then strikes an electric arc - perhaps by quickly tapping an electrode on the work piece. Once the arc is lit, the welder moves his torch in small circles until the heat creates a small pool of molten metal. When the welder gets adequate fusion, he moves the torch slowly along the seam between the parts to be welded, melting their adjoining surfaces, and feeding the welding rod into the pool of molten metal, just ahead of the arc.

The welder must painstakingly control his welding speed, the speed at which he fills the welding filler, and the welding current. He moves the welding rod and torch smoothly forward, making sure the hot end of the welding rod and the hot solidified rod are unexposed to the contaminating air. With foot controls, the welder adjusts the current to get proper fusion and penetration in the weld. He judges how much current to apply by the size of the molten metal puddle.

Manual welding is a discipline that requires skill and experience. The power generation, chemical, petroleum, and aerospace industries cannot take chances with human discrepancies and would greatly benefit from an automated welding process.

2.1 Automated Welding

For automated welding, there are four process variables that determine the quality of the weld. These are:

1. Welding Current,

2. Arc Voltage

3. Travel Speed

4. Filler-feeding Speed

Welding Current determines the quality of weld to a large extent. When we reduce the current, we reduce both the penetration and the width of the weld. For a given workpiece geometry, we usually want the current to be held constant. When we need more than one current level during a weld, the current should slope up or down to a new level rather than jumping to a new level. If current is constant, we can measure the distance from electrode to the workpiece (arc gap) as a voltage potential called the arc voltage. In some welding processes, this is used as feedback to control the arc gap. For a given current and arc voltage, travel speed determines the amount of energy delivered per unit length of the weld. Increasing the speed while keeping the current constant reduces both penetration and width of the weld. A final consideration is the control of how quickly the filler wire is filled into the molten pool. In automated welding, as with manual welding, the control of these four variables determines the quality of the weld.

The programmable power source (PT) in the welding circuit is used as a regulator and is controlled by the microcomputer. Output of the three phase rectifier is stepped down to rectified voltage of approximately 50V. The control circuit consists of sampling devices 'C' and 'V' , microcomputer, output device 'O' including amplifiers and driving components. The electric potential levels in the digital circuit are standard TTL output. We are interested in measuring the arc current and arc voltage in the above circuit. The current can be measured at 'C' using the current sensor, operating on the principle of Hall-effect.

Arc Voltage is measured at point 'V'. The variations in arc gap during the process of welding causes variations in the arc current. As a result, the current through R1 and R2 change. This changes potential drop across the resistor R2. In short, arc voltage variations are directly proportional to the variations across the resistor R2 which is used as voltage feedback for the control circuit.

2.2 Welding with Robots

For workpieces with simple geometry, the operator can set weld parameters to fixed preset values so that the electrode moves along the seam keeping the weld parameters constant. With more complicated geometries demand more positioning precision. A programmable weld profile, one that describes complicated changes in weld current, arc voltage, travel speed and wire-feed rate at all times during the weld is needed. The robot arm is used to obtain positioning precision with extra control for moving the electrode. To obtain a programmable weld profile, we need a programmable power supply. The robot arm moves the torch over the workpiece while the controller makes sure that it follows shape of the part, and power supply controls the weld parameters.

To program the weld profile, set-up test is performed by the "weld engineer". He programs a different set of process variables for every time interval during the welding process. He can program more than hundred points to hold different weld parameters. In this phase, he teaches the robot arm the shape of the workpiece. Once programmed, the power supply keeps welding parameters at their correct values for appropriate portion of welding sequence. The robot arm produces continuous path motion, travelling over the length of the seam.

A human varies the arc current to get the suitable molten weld puddle size. Since a system cannot measure the puddle size, it measures the arc voltage, which at constant current is a function of the z-axis distance between the electrode and the workpiece. The further the electrode travels from the workpiece, greater the arc voltage, wider the arc gap, and less heat penetrates the weld. The potential voltage drop across the arc is the fed back to the power source, which then controls the z-axis motor on the end of the robot arm. The system can also be programmed for specialized welding techniques such as weaving and oscillations.

2.3 The Measurement of Welding Parameters

2.3.1 Arc Current

Hall effect current transducers are used to measure the arc current. It measures AC, DC and IMPULSE.

> Principle: The magnetic field created by the current to be measured (primary circuit) is compensated by the field created by secondary winding, which is equal in ampere-turns and opposite in polarity to the primary which includes a Hall effect device, associated with compensating electronic circuit. This transducer operates on the principle of compensation of magnetic field or zero magnetic flux method (feedback system).

The Magnetic field is constantly controlled at zero, the amount of nulling current required to hold zero flux is the measure of primary current flowing, multiplied by the ratio of secondary winding. The fundamental relation is

$$[N_p.I_p] = [N_s.I_s].$$

This nulling current can be further expressed as voltage by dropping it across resistor. Additional circuitry is necessary to get the desired output.

2.3.2 Position Measurement

Optical Encoders are used to measure the position. Two main forms of encoders are: the incremental encoder, and the absolute encoder. Incremental encoders can only indicate displacement as a number of increments since the movement started; where as absolute encoders indicate the absolute position. Incremental encoders produce output pulses which are counted by an up/down counter, so the value of the count is an indication of how far the disk or strip has moved since the count began. Often two sensing elements are used, positioned on the transducer such that their outputs are 90 degrees out of phase; enabling simple logic circuits to determine the direction, and hence whether counter should move up or down.

The code wheel in the optical encoder rotates along with motor shaft movements interrupting the light beam from the LED to the photodiodes. These photodiodes form a part of integrated circuit consisting of the signal processing circuitry necessary to produce digital output. Comparators receive these signals and produce final output for channel A and channel B. Both the sensors are placed exactly one-half slot-width apart so that, as disk rotates, the light shining through the slots produces detector signals that are 90 degrees out of phase with each other. For clockwise rotation of the disk, the sensor V1 is always activated first and V2 for anticlockwise rotation.

2.3.3 Position-Counting Hardware

The signals from the above encoders are too obscure for most simple counters. Therefore, special hardware has to be provided to translate the pulses into up and down count pulses. It is then possible to derive position simply by counting pulses directly from the encoder. One could design a sequential machine to decode this complex signals and get the counter output of count-up (CU) and count-down (CD).

For this research, two-channel incremental optical encoder (HEDS-5500) was selected. This encoder has attributes of high reliability, high resolution, and easy assembly. It has following features:

Two Channel Quadrature output with optional index pulse.
Quick and easy assembly
Resolution 512 counts/rev
Incremental Encoder
TTL compatible
Low cost
Small Size

This encoder operates on the principle described above. It has resolution of 512 counts per revolution. The encoder position control information is then fed to the general purpose motion control IC (HCTL-1100).

2.3.4 HCTL-1100 General Purpose Motion Control IC

The HCTL-1100 is high performance, general purpose motion control IC. It performs all time-intensive functions of digital motion control freeing the host processor from all these computations. It has the following features:

Position and Velocity control
Low Power CMOS
Programmable Digital Filter and Commutator
8-bit Parallel, and PWM motor command Ports
TTL compatiblity
SYNC Pin for coordinating Multiple motion control ICs
100KHz to 2MHz operation
Encoder input port.

The HCTL-1100 is a general purpose motion controller which provides position and velocity control for DC, DC brushless stepper motors. It receives input commands from the host processor and position eedback from an incremental encoder with quadrature output. An 8-bit bidirectional multiplexed address/data bus interfaces the HCTL-1100 to the Host processor. The encoder feedback is decoded into quadrature counts and a 24-bit counter keeps track of the position.

3.0 SELECTION OF THE DRIVER SOFTWARE FOR THE DAQ SYSTEM

After researching several hardware/software options for designing the data acquisition system (DAQ) for the Plasma Welding Robot application, a 33-MHz 486 computer with 16-MBytes Random Access Memory (RAM) and 245 MBytes Hard disk space was chosen as the DAQ platform. The Data Acquisition Board (CIOAD16Jr-AT) in conjunction with Simultaneous Sample and Hold card (CIOSSH16) was selected as our data acquisition Hardware and the Device Driver (DRIVERLINX/VB) as our device driver. Visual Basic was selected to design the front-end Graphical User Interface (GUI) and and as the development software.

3.1 Data Acquisition Board (CIO-AD16Jr-AT)

High speed multi-function analog and digital I/O board 100% compatible replacement for the MetraByte's popular DAS-16G can be installed in any IBM PC/XT/AT/PS30 or compatible. Converts the personal computer into a high speed data acquisition system. The specifications for DAQ board are as below:

3.1.1 Specifications

Analog Inputs: The analog input section of the CIO-AD16 Jr. has been designed for flexibility and accuracy in a number of configurations and ranges. The analog signals are brought on board by a standard 37 pin D connector directly to two multiplexers. The two multiplexers may be configured as 16 channels of single ended input or 8 channels of differential input. Differential inputs can reject noise caused by ground loops but require a 3 wire hookup as opposed to 2. Signals are amplified by a programmable gain amplifier prior to conversion by the A/D converter. An integral sample & hold captures the signal which is then converted by the A/D converter. The 12-bit A/D converter provides a resolution of 1/4095 parts of full scale. The speed of data gathering is dependent on the method of triggering and data transfer.

Simultaneous Sample & Hold: Simultaneous Sample & Hold is an option which allows 16 analog input channels to be triggered simultaneously. This option is important to applications where channel to channel skew is not acceptable, such as audio digitization. The CIO-AD16Jr. can trigger an external CIO-SSH16 via the unused D/A REF1 input. The CIO-SSH16 reduces channel to channel skew from a minimum of 8 uS (CIO-AD16Jr., 3uS CIO-AD16Jr-AT) to zero with less than 50 nS aperture uncertainty.

Counter Timer: The 8254 counter/timer chip has 3 counters 16 bits each. Much of the 8254 is used by the CIO-AD16Jr. as a pacer clock to synchronize A/D conversions. One full counter, counter 0, is available for counting, pulse generation or frequency measurement. The output of counter 2 is available to provide external synchronization to the A/D converter or as a programmable rate source. If an application needs additional counters, a 10 counter CIO-CTR10 is available.

Triggering: A Trigger is the event that begins an acquisition/transfer cycle. There are three ways to trigger a CIO-AD16Jr.: software, internal or external. There are also three ways to transfer data from the CIO-AD16: program, interrupt service routing or DMA. An internal trigger is useful for synchronizing samples to a known time base; on the board XTAL and 8254 are programmable divider. Using an external trigger allows synchronizing samples to an external event.

I/O & Control Register Map: The CIO-AD16Jr. and MetraByte DAS-16 are 100% software compatible with one another because the I/O registers are located at the same address. The first 16 registers of the CIO-AD16Jr-AT also share these register functions.

Compatibility Mode-Jr-AT only: A switch on the CIO-AD16Jr-At will force the board into 100% CIO-AD16 compatibility mode. In compatibility mode, the Jr-AT has only 16 registers and those 16 are identical to a MetraByte DAS-16G. When the switch is in the Jr.-AT mode, the Jr.-AT is still 100% compatible with a DAS-16G and an additional 8254 counter is present at Base + 16-19. the upper four bits of Base + 11 open up and become the high-speed A/D transfer control bits.

Table 3.1 Specification Table

A/D Specifications	12 Bit
Channels	16 SE or Differential
A/D Type	Successive App. ADS 7800
Conversion Time	3 uS
A/D Convert & Transfer Speed	130KHz CIO-AD16Jr.
Accuracy	0.01% +/- 1 LSB
Integral Linearity	+/- 1 LSB

3.2 Simultaneous Sample and Hold (CIO-SSH16)

In many applications, it is crucial to determine outputs of various signals at exactly the same time. Any time skew between the sampling of these variables can lead to incorrect portrayal of how a given application really functions. Most data acquisition boards are not designed to sample multiple channels at the same time. They sample only one channel, switch the multiplexer to another channel and take another sample.

For example, assume that it is important to know the state of all three input signals at the same instant. The samples taken with CIOAD16/SSH16 and by regular data acquisition board provide the information desired. The samples taken by standard multiplexed data acquisition board can misrepresent the condition by considerable amount. For example, if the application needs to know the initial state of the of each variable, DAS16/SSH16 provides the information required. The simultaneous sample and hold accessory and acts as the front end signal amplification and capture for the CIO-AD16 series of analog input boards.

There are two major functions on board. Sixteen differential amplifiers have individual selectable gains providing very flexible amplification of individual signals. After amplification, each channel has a sample and hold which is controlled by the analog input board. The CIOAD16 eliminates the channel to channel input skew associated with multiplexed A/D inputs. A fast A/D sampling at 100,000 samples per second will exhibit a minimum channel to channel skew of 160 microseconds. Since skew is additive from channel to channel, the total scan skew is 160 microseconds. 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. Even low speed applications may require simultaneous sampling for all channels. The specifications for CIOSSH16 is described next.

3.2.1 Specifications

Sample & Hold Trigger: The signal S&H Trigger, from the CIO-AD16 controls the mode of the LF398 TRACK/HOLD chips on the SSH16. When T/H is high, all the LF398 chips track the analog signal on the SSH16 inputs. 200 nanoseconds after the T/H signal goes low, all the LF398s have settled and are holding a constant output voltage. The LF398s continues to hold a stable output voltage until the T/H signal once more enters TRACK mode.

The SSH16 is designed to be used as an amplification and sample & hold accessory for the CIO-AD16. The CIO-AD16 signal, S&HOUT (pin 26), synchronizes the LF398 sample & hold chips to the CIO-AD16. The control signal is hardware generated so any CIO-AD16 software works with the CIO-SSH16.

Input Gain Switch: The input gain switch controls the amplification level of the INA110 differential input amplifier. Each switch controls one gain weight and weights are additive. Shown to the right is a gain of 500+100 = 600. When all switches are to the right (off) the amplifier is at unity gain. In addition, the CIO-AD16 gains are multiplicative with the CIO-SSH16.

Superior to Programmable Gain: Programmable gain allows the selection of a unique gain/range for every channel and to control the gain under software. A complete amplifier on every channel provides the same flexibility and superior performance.

Calibration: The A/D board should be calibrated for the range it will be operated in. Each channel of the CIO-SSH16 can be calibrated for the range wanted. An A/D board with programmable gain can be calibrated for only one range at a time.

Speed: Amplifying the input to the A/D should not limit the maximum A/D sample rate. The CIO-SSH16 will sample at the maximum A/D throughput even at maximum gain of 800. Checking the specifications of any programmable gain A/D board shows that the throughput limitations at high gains are often severe.

Flexibility: Each channel should support a different gain even at DMA speeds. The CIO-SSH16 supports a different gain on every channel regardless of data transfer method. Many PGA A/D boards must remain in a single range during DMA operations. Those that are that flexible are also expensive, costing more than a CIO-AD16 and CIO-SSH16 combination, and do not include simultaneous sample & hold.

Table 3.2 Specifications Table for Simultaneous Sample and Hold Card

CHANNELS	16
Differential Amplifier	INA110
Gain Weights	1.10.100.200.500
Sample & Hold Chip	LF398
Aperture Time (Max.)	250 nSec
Acquisition Time (Max.)	10 uSec
Droop Rate	+/- 100 uVolts / mSec
Noise RMS 10 KHz - 100 KHz	0.01% of reading +/- 1 bit
Accuracy	0.01% of reading +/- 1 bit
Power Consumption (Max.	5.9 Watts

Software: Any software which supports the CIO-AD16 (or MetraByte DAS16) automatically supports the simultaneous sample & hold features of the CIO-AD16 and CIO-SSH16. Sample and hold triggering is done by hardware and does not require any new software, or modifications to existing software. A CIO-SSH16 installed as an analog signal conditioning accessory to your CIO-AD16 indicates your intention that
all channels be triggered simultaneously. No further set-up or programming is required. The CIO-AD16 and CIO-SSH16 combination is the perfect upgrade to any outdated DAS-16 applications that would benefit from simultaneous sample & hold or superior gain and range on every channel. No reprogramming of your application code, whether written in ASSEMBLY, C or BASIC is required.

3.3 The Device Driver - DRIVERLINX/VB

DriverLINX is the MS Windows 3.0 Driver for DAS-16 and DAS-8. Drivers are used to allow communications to occur with the data acquisition boards at a high level. This allows programs to be written generically without having to write directly to the board registers. For our application, we used MS Windows 3.x Dynamic Link Libraries (DLL) interface. Microsoft Windows DLL standard allows the use of any languages that supports the DLL construct. We use DriverLINX/VB for our application. DriverLINX/VB is a hardware-independent Visual Basic Custom Control designed to support hardware manufacturers' high speed analog and digital data acquisition using Visual Basic and Microsoft Windows 3.x. 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 the user flexibility to set up and run the data acquisition program, and automatically transfers collected data to another application through the standard Dynamic Data Exchange (DDE).

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

- Language Independent dynamic link libraries (DLLs).
- Features multitasking and multiuser capabilities.
- Provides developers of custom Windows 3.x applications a sophisticated, high-level interface to PC data-acquisition hardware by insulating developers from the complexity of interfacing with boards.

DriverLINX can save developers significant efforts in porting their data collection, instrumentation, monitoring, and control applications into the popular Windows 3.x environment.

This driver contains all the intelligence necessary 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. Multiple copies of one application or multiple applications can access DriverLINX without interfering with ongoing tasks. Special version of DriverLINX (DriverLINX/VB) is available to program in Visual Basic.

DriverLINX provides more than 100 services for creating foreground and background tasks to perform analog input and output, digital input and output, time and frequency measurement, event counting, pulse output, and period measurement. These services implement the most common data acquisition protocols without sacrificing the high speed data acquisition capabilities of the hardware.

DriverLINX overcomes the limitations of many data acquisition drivers, which provide only a low-level hardware interface that requires the application to synthesize protocols from sequence of calls to the driver. Other drivers provide high level interface but impose software, rather than hardware limitations on data collection.

As DriverLINX is a DLL, multiple applications using DriverLINX do not increase its memory requirements. Its non-swappable memory requirements are typically 10-12KB.

3.3.1 Data Driven Interface

DriverLINX/VB is implemented as two Visual Basic custom controls. Every data acquisition task is properly specified using the properties of "service request" and the "Logical Device Descriptor" custom controls. Like all other Visual Basic control, the programmer either sets the properties at design time or the application assigns values to the properties at run time. Service Requests identify the requested driver, the logical device to execute the service, the device subsystems needed (analog input/output, digital input/output, counter-timer, or other), the requested operation mode (polled, interrupt, DMA, or other), and the requested operation (start, status/read, stop, etc .). Logical Device Descriptors provide the application hardware and vendor independence by supplying a standard database of hardware features and capabilities. By employing this information, applications can easily and flexibly adapt to different hardware environments.

Service Requests implement the concept of abstract events for timing, starting and stopping the data acquisition tasks. By simply selecting from a wide choice of events, an application can easily specify complex data acquisition tasks. For analog and digital I/O tasks, events include software commands, terminal counts, analog or digital input monitoring, and many variations of counter-timer functions. Trigger events include analog thresholds, digital input patterns, time of the day or delay, memory flags, and internal or external interrupts. Event triggering allows flexible, data driven pre-, post-, and midpoint triggering of data collection process. By selecting various starting and stopping events, the application can implement variety of data acquisition protocols.

3.3.2 Logical Device Descriptor Control

A Logical Device Descriptor is a custom control whose properties describe the hardware characteristics of the data acquisition devices managed by DriverLINX/VB. A Logical Device in DriverLINX/VB is an abstraction for describing the data acquisition hardware within the computer. Each logical device is divided into six subsystems: the device itself, the analog input, the analog output, digital input, digital output and counter/timers. Usually, each logical device will directly correspond to a physical device. As logical device is an abstraction, however, multiple Logical devices may refer to the same physical device or one Logical Device may refer to a hardware on multiple physical device.

An application or the user may find it more convenient to have several Logical Devices to refer to one physical device. For instance, an application supporting a hardware device that is capable of performing simultaneous analog input and output may find it more convenient to treat the analog input and the analog output sections as separate Logical Devices or it may be easier to consider a physical device as separate Logical devices when the device is configured with different characteristics, such as single or differential analog input, different interrupt or DMA channels, etc. Modular data acquisition devices may also be described by different Logical Devices depending on the selection of modules.

4.0 DEVELOPMENT OF THE GRAPHICAL USER INTERFACE (GUI)

The Data Acquisition System software for the Plasma Welding Robot was coded in Visual Basic. After a brief explanation on Visual Basic, we will explain with examples the design of Graphical User Interface.

4.1 Visual Basic, Programming Concept

Development of this software incorporates Window programming techniques and operates in Microsoft's Windows 3.x environment. Programs in Visual Basic use a different approach than traditional DOS programs. DOS programs are written sequentially, i.e. one event follows the other. In DOS programs, controls proceeds down the list of statements, more or less in the order which the programmer has 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 an 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 but, the user is. The user selects controls among all the options present, and it is up to the program to respond to them in an appropriate manner. The code is specifically designed to respond to a particular event called-on by the user. The program is typically a collection of code sections whose execution order is determined by the user. 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 of the others.

4.2 Visual BASIC Environment

In Visual Basic, the primary objects are forms and controls. A form defines a window on-screen. Forms (and their controls) are what user actually see when running the application. A control is an object which is placed on the form. Each control is represented by an icon in the tool-box. Each of these objects have set of properties associated with them, some of which can be set during the design time or can be modified during run-time. They also have associated set of event procedures. These events are recognized by writing code for the appropriate event procedure. Objects can take on a life of their own. For example, one can pass objects as parameters to various procedures. Forms can be saved as disk files and can be reused in other applications. This saved file includes a description of the form, the controls on the form, changes in their property list, and associated event procedures. This feature of Visual Basic is brings it close to Object Oriented Programming technique. This object now exists independently of any particular application.

4.2.1 Brief Introduction

On entering the Visual Basic environment, we see the opening screen. Here is a brief introduction on the major components of the Visual Basic Screen.

Title Bar: Title bar is present in most Window's application, contains the name of the application.

Menu bar: This is the gateway to Visual Basic's drop-down menu system. The menu bar contains options to manipulate files, edit and debug programs, view particular windows and get context-sensitive help.

Form Window: This is the main window used to design the interface. Text boxes, command buttons and all other visual objects are placed on the form to design an application. When application is run, the form is what user sees. An application can have multiple forms.

Tool bar: The tool bar icons provide instant access to most frequently used commands.

Tool box: It contains set of graphic objects called 'Custom Controls' which can be selected and placed on the form.

Graphic Server: This icon represent some Windows Graphic Routines to support the Graph control. It is seen only when the graph control is loaded.

Project Window: This window lists the files, forms and modules that compromise the current application.

Properties Window: This window shows the attributes (Properties) associated with a selected form or control. By modifying values in this window, the attributes to the control can be modified.

Window environment is extremely friendly to the user, but programming in 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 the first step, creating buttons and menus - this is where Visual Basic really simplifies the code development process. Before, it was tedious process to design the appearance of the windows (i.e. 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 simplified. Visual Basic allows us to simply draw the windows we want, as well as the buttons, boxes and labels one wants. Adding or removing buttons or boxes works 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.

4.3 The User Interface

Some user friendly Visual Basic application interfaces are designed for Data Acquisition System. The main application consists to three modules linked together as one utility. The main module with the icon 'MAIN' and the title 'MONITOR' branches either to the acquisition module or the replay module, as selected by the user. This is the first interface of the Data Acquisition system. The animated button tool is used to display the sequence of welding events, switched periodically to give the impression of robot movements. This animation signifies that the welding is in progress. The picture tool is used to display the NASA Logo. There is a real-time clock displayed in the text window which shows the system time. The Command buttons are use to trigger any other user defined event. The 'Start Data Acquisition' window pops up and is used for the real-time data acquisition and the 'Replay Recorded Data' pops up, which is used to study and further analyze the data. The 'System Info' uses the Windows API to get the information regarding the current system in use, the available disk space etc .... .

Monitor 2, deals with the Real-time Data acquisition of the welding process parameters. For acquiring the data in real-time, this module is linked with the device driver 'DRIVERLINX/VB' to drive the CIOAD16-Jr. This is done by using the 'driverlnxvb.vbx' file as an interface between the code and the

Visual Basic environment. This library 'CIOAD16.DLL' in this case, has been customized to be used as a Visual Basic Custom Control. The DLL file is the actual device driver for the data acquisition hardware. To add the DRIVERLINX/VB custom control to the Visual Basic Tool box in the design mode, one must start Visual Basic in the design mode and add "driverlnxvb.vbx" in the Add File dialog box. Two new icons are added to the Tool bar, and name of the control file is added to the Project window. These can now be used as any standard Visual Basic Controls. Changes in the configuration (CIOAD16.INI file) is then made to match with the hardware. Then the code is written to execute the required mode to acquisition as discussed in Chapter 3.

Now, on the form, the Graph tool plots the line graph of the arc current and arc voltage acquired by the data acquisition card. Other Process parameters displayed are:

- Absolute torch position in the Cartesian coordinate system.
- The transformer core temperature, to check over heating.
- The wire feed speed.
- The torch travel speed.
- The Plasma flow and pressure.
- The Pneumatic flow and pressure.

The parameters button when clicked, displays the weld parameter window. This window consists of the various computed parameters of the arc voltage and current waveforms. Following parameters are displayed on the Weld Parameter Window.

- Average Positive Voltage
- Average Negative Voltage
- Average Ripple
- Instantaneous Voltage and Current values
- Reverse Current Hits
- The Forward time and the Reverse time.

These values are computed using the 'C' code to speed up the computations. The data transfer between the two takes place by accessing common files.

To initiate the acquisition, the 'Start' button is clicked. A message is prompted, asking the name of the file to store the data in. All files are stored with the extension *.daq, unique only to the acquired data files. The data is stored in the standard spread sheet format (i.e. Microsoft Excel), so that the user can use the powerful analysis features provided by Excel for accurate and detailed data analysis. The data is stored with the 'Date' and the 'Time of Acquisition' for accurate chronological records. The exit button exits the acquisition module and brings the user back to the main screen `MONITOR'.

The replay button on the main screen, brings up the module to replay the collected data for subsequent analysis of the arc parameters. The user can select the file to replay. The load menu loads the data from the file and display them. The first graph displays all points of both the arc-voltage and the arc-current combined. The second and the third graph plots selected range of the current and voltage combined.

The range to be plotted selected in the parameter window called on by the 'Parameters' item from the menu bar. The range can be selected from 50 points to 2000 points. The 'View' item gives user choice of selecting either the separate mode or the combined mode of plotting the graph. The Combined mode plots both graphs together for easier comparison. The 'Grid' item plots grid in the combined mode. The grid width is also menu driven.

There is a page scroll facility provided which scrolls the page by the distance equal to the number of points selected in the parameter window. The 'Page Back' and the 'Page Ahead' uses this scroll feature. Auto scroll scrolls both graphs forward automatically every two seconds. Another interesting feature included in this module is the ability to select the exact location of graph to analyze. Under graph 1, there is the scroll bar provided along the length of the graph. By matching the scroll bar position to the exact location to be analyzed, both the graphs will start plotting from that point. The exit button brings one back to the main screen.

The gauges have to be configured to match with the input data. The Full-dial gauge is used in the two dial gauges shown. The background of the gauge is a bitmap, individually drawn. Small code is written in the event procedure of the gauge to restrict its movements and give it the shape as seen. The 'Auto' button activates the gauges in real-time mode to display the data in real time. The scroll bar is provided to test the gauges.

This user interface was tested successfully using the real time signals. This complete module acted as the virtual instrument for the data acquisition system. The next chapter summarizes this research discusses the possible future enhancements for this project.

5.0 SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.1 Summary

Instrumentation technology is constantly changing and it is important to study the new alternatives available before designing a data acquisition system. As of today, the design of DAQ can be categorized into four distinct types of instruments: - IEEE 488-controlled instruments, plug-in data acquisition boards, RS-232 controlled instruments and the VXI Instruments. In many applications, any of these instruments can be used, but there are differences and trade-offs.

Personal Computers (PCs) have evolved into a powerful, cost-effective computing platform and are being widely used for data acquisition and control application. The evolution of A/D converter technology and signal conditioning accessories has helped the plug-in DAQ boards become a popular and cost-effective instrumentation choice. IEEE 488 and VXI systems are more expensive and generally used for sophisticated measurements.

For the 'Design of Data Acquisition System for the Plasma Welding Robot', Plug-in boards were selected, they are economical and perform adequately. Transducers were connected through the signal conditioning modules to the Simultaneous Sample and Hold card to acquire data of all channels simultaneously after which the data was sent to the DAQ. The acquired data is then transferred directly to the computer memory. Call drivers via DLL (DriverLINX) for Windows 3.x. as the device driver were used, which enables us to interact with the data acquisition hardware at high level and also give access to various features of Microsoft Windows.

Windows provide multi-tasking and an improved Graphical User Interface (GUI). Visual Basic for Windows was used to develop the GUI. It eliminates the requirement to learn new high-level language like 'C' or Pascal. Designing the windows application in Visual Basic is very convenient. Programmer configures the window by moving elements from tool bar to desired location on the form. Code is written in each object's event procedure to link various objects together to form an application. DriverLINX/VB appears as tool on the tool bar and is used as any other Visual Basic object. This custom control has to be tailored to according to the hardware used. Visual Basic has its limitations and takes much longer to execute the computation intensive code than compiled code for other high level languages. Hence it is not advisable to use Visual Basic for intensive mathematical Computations. To speed up the process, hardware drivers and intensive mathematical computations are done in 'C' or other high-level language and compiled into a DLLs. This DLLs can then be called from Visual Basic. As a result, the user combines the advantages of both the languages:

Visual Basic to design the GUI where speed is not critical, while high-speed language to perform compositionally intensive tasks.

Windows also provide the capability to share data between applications via clipboard, Dynamic Data Exchange (DDE) and Object Linking and Embedding (OLE) which have made windows very popular. For example, data can be acquired in real-time and sent directly to Excel and use the powerful analysis features of Excel.

5.2 Conclusions

It is apparant from this research that the following conclusions hold true:

1. PC based Data Acquisition System offers a number of benefits: Data can be gathered at multiple locations simultaneously, letting user monitor more points with less labour.
2. Data from all test points can be gathered at single location (the PC) for analysis and comparison.
3. Multiple computers can be networked for data and resource sharing.
4. Access to variety of commercially available software packages.
5. Option to use windows environment which provides high speed and multitasking feature.
6. PC based Data Acquisition using the plug-in data acquisition boards is the fastest growing instrumentation alternative as it offers very good price-to-performance ratio.
7. Plug-in data acquisition system offers ease of installation and flexibility. This system can be mounted on rack and moved to different sites without much complications.
8. Selection of data acquisition board demands careful study of the application requirements and the instrumentation alternatives available in the market.
9. Windows environment for data acquisition provide high quality of Graphical User Interface and multitasking. The user has option to use powerful Microsoft Window's features like Dynamic Data Exchange and Object Linking and Embedding or use other window's applications like Lotus 1-2-3 or Excel for subsequent data analysis.
10. The development of Graphical User Interface introduces a new pardigm called `Virtual Instruments (VI)' in the Instrumentation World.
11. With PCs and general purpose DAQ hardware, user can create specialized VIs for individual application - those for which a dedicated instrument may be too expensive or may not exist.. Thus VI instruments become user-defined rather than vendor-defined.

12. Modular design of VI instruments can be used for various applications and can never be outdated.
13. The user has Flexibility to alter the Data Acquisition System for future applications.
14. Commercially available software packages can be used for most applications, but some application require customized programming to design application-specific interface. Visual Basic simplifies Window's programming and eliminate requirement to learn new high-level language.
15. A PC-based Plug-in data acquisition system is not recommended when application requires a large number of input parameters or when they require very high resolution.

5.3 Recommendations

A portion of this research was directed to explore the possibility of enhancing the data acquisition system by in incorporating new technologies like the sound system and the impact of the new operating system on the current technology.

5.3.1 Sound System and Voice Interface

The Microsoft Windows Sound System is designed to be an audio productivity resource for any application using Windows version 3.1. It enhances the work with the power of sound. It enables us to add voice messages and recorded annotations to presentations, electronic memos spreadsheets and other documentations. The use of Voice Pilot increases ones efficiency by allowing voice commands to control the computer. In this research, we have investigated the possibility of using Windows 3.1 sound capabilities and object linking and embedding (OLE) technology to record and to acquire and analyze the acquired sound data.

The Windows Sound System Package comes equipped with following items:

- The Windows Sound System Board.
- Headphones and microphone.
- Windows Sound System disks.

To analyze the slow data, Windows sound system proof reader has been used. The proof reader reads the stored data loud from the spread sheets. The acquired data was filtered into Excel format. The proofreader can read the data as it is being acquired or can can read the stored data. Once the proof reader is installed it appears as `proof' menu on the excel menu bar. Select range of cells from which the data is to be read. Select Resume form the proof menu. The proof reader reads the data till end of the selection. It can read data by row or by column. For real time data acquisition, we have developed a system using Visual Basic which stores the data directly in Excel using the DDE characteristics of Visual Basic. As the data is being acquired, selection of the proof read button on the screen will invoke the Excel and executes a macro which will select the `On Entry' mode of the proof reader. Result: the data is read out loud as it is being acquired. This feature proved very useful to the Welding Operator as he could keep track of some important parameters while attending to other tasks.

Additional research was done on using the voice commands to manouver within the application. This requirement was suggested by the Plasma Welding Operator to view parameters of interest while work is in progress.

5.3.2 Future Operating System

Windows NT 's multiplatform capabilities seems to provide promising future for PC based data acquisition because of its true multitasking nature. In Windows NT, there is a Hardware Arbitration Layer (HAL) between the actual windows and the hardware which manages the I/O hardware interfacing. This means that the user must install DAQ drivers as a part of system-level device drivers. This drivers will accept programmable interrupts and DMA controllers through HAL calls. It is still too early to make appropriate predictions about DAQ hardware with NT as interrupts are handled very differently in NT. Moroever, NT has some drawbacks which are supposed to be fixed in its next release. It would be interesting to explore possibilities to use NT and IBM-OS2 for data acquisition because of their true multitasling natures and high processing as they are 32-bit operating system supporting multi-thread execution unlike the traditional DOS Operating system.

NASA National Aeronautics and Space Agency	Report Document Page	
1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Automated Data Acquisition Technology Development Automated Modeling and Control Development		5. Report Due MAY 1995
		6. Performing Organization Code University of Alabama in Huntsville
7. Author(s) Peter L. Romine		8. Performing Organization Report No.
		10. Work Unit No.
9. Performing Organization Name and Address University of Alabama in Huntsville Huntsville, Alabama 35899		11. Contract or Grant No. NAS8-38609 DO# 82
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546-001 Marshall Space Flight Center, AL 35812		13. Type of report and Period covered Final Technical June 7, 1993 - June 6, 1994
		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

This is the Final Technical Report for Delivery Order #82, on the Automated Data Acquisition System (ADAS) for Automated Modeling and Control Development.

Covers the period of performance June 7, 1993 - June 6, 1994.

17. Key Words (Suggested by Author(s))		18. Distribution Statement	
19. Security Class. (of this report)	20. Security Class. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	27	

NASA FORM 1626 OCT 86