

ABSTRACT

Software Coincidence of General Purpose Data Acquisition Cards with MATLAB Toolbox

DİNÇ, Hakan

Master Degree in Electrical and Electronics Engineering

Supervisor: Prof. Dr. Arif Nacaroğlu

May.2006, 50 pages

Data acquisition systems are generally used to acquire physical data by using their concerning software and hardware. There are many components of professional measuring systems such as data acquisition hardware, transducers, signal conditioners, computer and software. In some professional testing and measurement software packages are already installed on hardware components of data acquisition systems to simplify the applications. However, these applications are used for small size data acquisitions, for very low voltages and frequencies.

In this thesis it is aimed that managing software of data acquisition cards and Matlab Data Acquisition Toolbox on one platform (Data Acquisition Control Panel). The preparing step of hardware initialization is made by written interface program automatically. All requested DLL (Dynamic Link Library) for Matlab can be registered, modified or automatically created, if it is not present on directory. Creating analog inputs and their necessary steps are reduced and linked to special buttons on Data Acquisition Control Manager. All unnecessary log files automatically removes from hard drives when digital oscilloscope program ends. It is not necessary to be familiar Matlab, Data acquisition Toolbox and hardware interface programs to use digital oscilloscope program of Matlab.

Data Acquisition Control Manager is linked to softscope (Digital Oscilloscope Program) to complete easy measurements and testing.

Keywords: Data Acquisition, Test and Measurement, Matlab, Data Acquisition
Toolbox

ÖZET

Matlab Tabanlı Yazılımların Genel Amaçlı Veri Toplama Kartlarına Uygunluğu

DİNÇ, Hakan

Yüksek Lisans Tezi, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi : Prof. Dr. Arif Nacaroglu

May.2006, 50 sayfa

Veri toplama kartları genel olarak fiziksel verileri kendi yazılımlarını ve donanımsal bileşenlerini kullanarak veri toplarlar. Profesyonel veri toplama sistemleri yazılım, veri toplama kartı donanım ve yazılımı, insiyatör , bilgisayar gibi birçok bileşeni vardır. Bazı veri toplama kartları uygulamaları kolaylaştırmak için ihtiyaç duydukları yazılım paketlerini kendi üzerlerinde barındırırlar. Ancak bu tür kartlar genellikle küçük çapta veri toplama amacı ile kullanılır.

Bu tezde veri toplama kartına ve Matlab'a ait Data Acquisition Toolbox yazılımlarının tek bir platform üzerinden kontrolü amaçlanmıştır. Veri toplama kartı ile çalışmadan önce gerekli hazırlıklar bu ara yüz yazılımı tarafından otomatik olarak yapılmaktadır. Gerekli olan DLL dosyalarının kaydı, düzenlenmesi ve yok ise yeniden oluşturulması otomatik olarak tanımlanmaktadır. Data Acquisition Control Panel olarak adlandırılan yazılım, ölçümler için analog girişlerin ve gerekli adımların oluşturulmasını sağlar. İhtiyaç duyulmayan gereksiz Log dosyaları program bitiminde otomatik olarak temizlenir. Kullanıcının Matlab, Data Acquisition Toolbox ve veri toplama kartı ara yüz programı hakkında bilgi sahibi olmasına gerek kalmadan kolay bir şekilde sayısal osiloskop programı çalıştırılabilmektedir.

Data Acquisition Control Panel bir buton yardımı ile Softscope adlı sayısal osiloskop programına açılmaktadır.

Anahtar Kelimeler: Veri toplama, Test ve ölçüm, Data Acquisition Toolbox, Matlab

TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	i
ABSTRACT	ii
ÖZET.....	iii
TABLE OF CONTENTS/CONTENTS.....	iv
LIST OF FIGURES.....	vi
CHAPTER 1 INTRODUCTION.....	1
CHAPTER 2 DATA ACQUISITION SYSTEMS.....	4
2.1. Data Acquisition Hardware.....	4
2.1.1 Analog input Subsystems.....	4
2.1.2 Analog Output subsystems.....	14
2.1.3 Digital Input / Output Subsystems.....	14
2.1.4 Counter / Timer Subsystems.....	15
2.2. Sensors.....	15
2.3. Signal Conditioning.....	17
2.3.1. Amplification.....	18
2.3.2. Filtering.....	18
2.3.3. Electrical Isolation.....	18
2.3.4. Multiplexing.....	19
2.3.5. Excitation source.....	19
2.4. The Computer.....	19
2.5. Software.....	20
CHAPTER 3: ADVANTECH DATA ACQUISITION BOARDS.....	21
3.1 Device Testing.....	22
3.1.1 Testing analog input function.....	23

3.1.2	Testing digital input function.....	23
3.1.3	Testing digital output function.....	24
3.1.4	Testing counter function.....	25
3.2	Analog input connections.....	25
3.3	Calibrations.....	26
 CHAPTER 4 DATA ACQUISITION CONTROL PANEL.....		29
4.1	Data Acquisition Toolbox.....	29
4.2	Software Requirements of Data Acquisition Control Panel.....	32
4.3	Softscope.....	32
4.3.1	Displaying Channels.....	32
4.3.2	Creating Additional Displays.....	33
4.3.3	Configuring Display Properties.....	35
4.3.4	Scaling the Channel Data.....	36
4.3.5	Triggering the Oscilloscope.....	37
4.3.6	Making Measurements.....	39
4.3.7	Exporting Data.....	40
4.4.	The Use of data Acquisition Control Panel.....	42
 CHAPTER 5: CONCLUSIONS.....		45
 REFERENCES.....		46
 APPENDICES.....		47
A.1	Connectors and pin assignment.....	48
A.2	I/O Connector signal descriptions.....	49
A.3	Block Diagram of Advantech PCI 1711.....	50

LIST OF FIGURES

Figure 2.1 Hardware Scanning.....	5
Figure 2.2 Sine wave quantization.....	7
Figure 2.3 Quantization error for sine wave	8
Figure 2.4 Differential input.....	10
Figure 2.5 Single-Ended input	11
Figure 3.1 Advantech test window.....	22
Figure 3.2 Analog input test	23
Figure 3.3 Digital input test.....	24
Figure 3.4 Digital output test	24
Figure 3.5 Counter tests	25
Figure 3.6 Single ended connections.....	26
Figure 4.1 Two channel Softscope screen	32
Figure 4.2 Displayed channels.....	33
Figure 4.3 Scope editor screen	34
Figure 4.4 Scope channel editor	34

Figure 4.5 Separated channel displays	35
Figure 4.6 Scope option editor screen.....	36
Figure 4.7 Channel Edit	38
Figure 4.8 Measurement editor	39
Figure 4.9 Measurement properties screen	40
Figure 4.10 Exporting Channels data	41
Figure 4.11 Channel editor	41
Figure 4.12 Data acquisition control panel main screen.....	43
Figure 4.13 Device testing of installed adaptors	44
Figure 4.14 Device test of Advantech DAQ card	44

CHAPTER 1

INTRODUCTION

Classical data acquisition systems are consisting of hardware and software components. In recent year's data acquisition hardware are designed in form that is more compact. Most of data acquisition boards that are widely used in laboratories can be installed directly to main boards of any computer by using ISA, PCI or USB slots. ISA slots are no longer produced on main boards of personal computers. USB data acquisition boards are used for fast and small size applications.

Data acquisition systems, as the name implies, are products and/or processes used to collect information to document or analyze some phenomenon. In the simplest form, a technician logging the temperature of an oven on a piece of paper is performing data acquisition. As technology has progressed, this type of process has been simplified and made more accurate, versatile, and reliable through electronic equipment. Equipment ranges from simple recorders to sophisticated computer systems. Data acquisition products serve as a focal point in a system, tying together a wide variety of products, such as sensors that indicate temperature, flow, level, or pressure.

Data acquisition systems are used in Automotive industry as vehicle test equipment, medical controlling machines such cardiograph, automation of production process control, Test and measurements in laboratories, military guardians systems such as radars and locators etc.

Today, most scientists and engineers use personal computers (PCs) with PCI, PXI, Compact PCI, PCMCIA, USB, FireWire, parallel, or serial ports for data acquisition in laboratory research, test and measurement, and industrial automation. Many applications use plug-in devices to acquire data and transfer it directly to computer memory. Others use data acquisition (DAQ) hardware remote from the PC that is coupled via Ethernet, parallel, or serial port. Obtaining proper results from a PC-based DAQ system depends on each of the following system elements

The computer used for DAQ system can drastically affect the maximum speeds at which can continuously acquire data. As computers continuously improve, DAQ system can take advantage of the computer's enhanced capabilities, including improved real-time processing, the ability to use complex video graphics, and higher streaming-to-disk throughput. Today's technology boasts Pentium IV and PowerPC class processors coupled with the high-performance bus architectures. The PCI bus and USB port are standard equipment on most of today's desktop computers and yield up to 132 Mbytes/s theoretical data transfer capabilities. External and portable PC buses such as PCMCIA, USB, and FireWire offer a flexible alternative to desktop PC-based DAQ systems while achieving up to 40 Mbytes/s transfer rates. For remote or distributed DAQ applications, measurement nodes can be placed near sensors, signal sources, and use standard networking technology, such as Ethernet, serial, or wireless. When choosing a DAQ device and bus architecture, keep in mind the data transfer methods supported by chosen device and bus and the transfer rates. The data transfer capabilities of computer can significantly affect the performance of DAQ system. Twenty years ago, PCs were capable of transferring at rates around 5 MHz, whereas today's computers can transfer significantly faster. As PC speed continuously increases, DAQ system speed increases as a result.

Today's PCs are capable of programmed I/O and interrupt data transfers. Direct memory access (DMA) transfers increase the system throughput by using dedicated hardware to transfer data directly into system memory. Using this method, the processor is not burdened with moving data and is therefore free to engage in more complex processing tasks. With National Instruments driver software, NI-DAQ 7, which serves as the interface between the hardware and the computer, the DMA routines to transfer waveform data across the PC bus were optimized, thus providing the ability to transfer data as fast as possible. To reap the benefits of DMA or interrupt transfers, DAQ device must be capable of these transfer types. For example, while PCI and FireWire devices offer both DMA and interrupt-based transfers, PCMCIA and USB devices use interrupt-based transfers. Depending on how much processing is needed during data transfer, the rate at which the data is transferred from the DAQ device to PC memory may be affected by the data transfer mechanism.

The limiting factor for real-time storage of large amounts of data often is the hard drive. Hard drive access time and hard drive fragmentation can significantly reduce the maximum rate at which data can be acquired and streamed to disk. For systems that must acquire high-frequency signals, select a high-speed hard drive for PC and ensure that there is enough contiguous (unfragmented) free disk space to hold the data. In addition, dedicate a hard drive to the acquisition and run the operating system (OS) on a separate disk when streaming data to disk.

In the past, applications requiring real-time processing of high-frequency signals needed a high-speed, 32-bit processor with its accompanying coprocessor or a dedicated plug-in processor such as a digital signal processing (DSP) board. However, with today's processors, the same real-time analysis can be performed without a specialized DSP because they are capable of rates around 2.5 GHz.

Determine which operating system and computer platform will yield the greatest long-term return on investment while still meeting short-term goals. Factors that influence choice may include the experience and needs of both developers and end users, other uses for the PC (now and in the future), cost constraints, the availability of different computers with respect to implementation time frame, and software support on that particular operating system. Traditional platforms include Mac OS, which is known for its simple graphical user interface and Windows 2000 or XP which include native plug and play and power management. Furthermore, real-time operating systems provide reliability and robustness that may appeal to particular application.

In chapter 2 the data acquisition real time components are detailed. In the next 3 chapters the brief information about data acquisition toolbox and system of acquiring analog data from hardware will be examined. In the chapter, Data Acquisition Control Panel the written program will be expressed with the steps of acquiring operations. Users can easily see the advantages of eliminating some unnecessary operations from data acquisition phenomena when program is in real-time use.

CHAPTER 2

DATA ACQUISITION SYSTEMS

2.1. Data Acquisition Hardware

Data acquisition hardware is either internal or installed directly into an expansion slot (ISA, PCI, USB etc) inside computer, or external and connected to computer through an external cable. For example, VXI modules are installed in an external VXI chassis, and data is transferred between MATLAB and the module using an external link such as FireWire.

At the simplest level, data acquisition hardware is characterized by the subsystems it possesses. A subsystem is a component of data acquisition hardware that performs a specialized task. Common subsystems include;

- Analog input
- Analog output
- Digital input/output
- Counter/timer

Hardware devices that consist of multiple subsystems, such as the one depicted below, are called multifunction boards.

2.1.1. Analog Input Subsystems

Analog input subsystems convert real-world analog input signals from a sensor into bits that can be read by computer. Perhaps the most important of all the subsystems commonly available, they are typically multichannel devices offering 12 or 16 bits of resolution.

Many data acquisition hardware devices contain one or more subsystems that convert (digitize) real-world sensor signals into numbers computer can read. Such devices are

called analog input subsystems (AI subsystems, A/D converters, or ADCs). After the real-world signal is digitized, it can be analyzed, store it in system memory, or store it to a disk file. The function of the analog input subsystem is to sample and quantize the analog signal using one or more channels. It is considered a channel as a path through which the sensor signal travels. Typical analog input subsystems have 8 or 16 input channels available to users. After data is sampled and quantized, it must be transferred to system memory. Analog signals are continuous in time and in amplitude (within predefined limits). Sampling takes a “snapshot” of the signal at discrete times, while quantization divides the voltage (or current) value into discrete amplitudes. Sampling, quantization, channel configuration, and transferring data from hardware to system memory are discussed below.

Sampling

Sampling takes a snapshot of the sensor signal at discrete times. For most applications, the time interval between samples is kept constant (for example, sample every millisecond) unless externally clocked. For most digital converters, sampling is performed by a sample and hold (S/H).

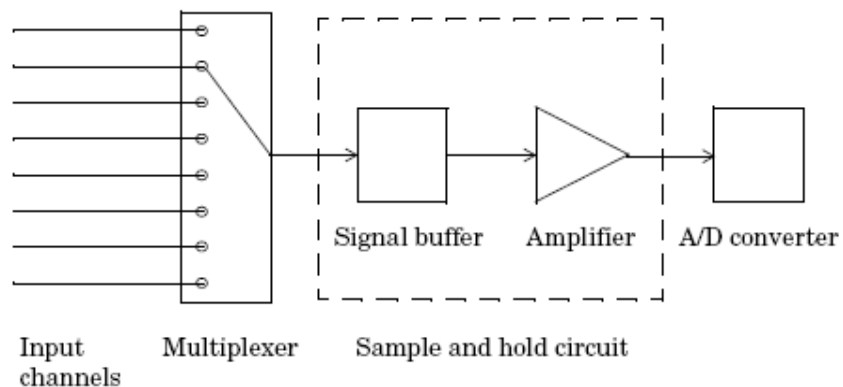


Figure 2.1 Hardware Scanning

Therefore, if multiple channels are used, those channels cannot be sampled simultaneously and a time gap exists between consecutive sampled channels. This time gap is called the channel skew. It can be considered the channel skew as the

time it takes the analog input subsystem to sample a single channel.

Additionally, the maximum sampling rate of hardware is rated at typically applies for one channel. Therefore, the maximum sampling rate per channel is given by the formula;

$$\text{Maximum sampling rate per channel} = \frac{\text{Max.Board Rate}}{\text{Number of Channels Scanned}} \quad (2.1)$$

Typically, achieving this maximum rate only under ideal conditions. In practice, the sampling rate depends on several characteristics of the analog input subsystem including the settling time and the gain, as well as the channel skew.

Configuration using scanning hardware is shown below. If channel skew can not be tolerated in application, it must be used hardware that allows simultaneous sampling of all channels. Simultaneous sample and hold hardware is discussed in the next section.

Simultaneous Sample and Hold Hardware

Simultaneous sample and hold (SS/H) hardware samples all input signals at the same time and holds the values until the A/D converter digitizes all the signals. For high-end systems, there can be a separate A/D converter for each input channel.

For example, suppose that need to simultaneously measure the acceleration of multiple accelerometers to determine the vibration of some device under test. To do this, SS/H hardware must be used because it does not have a channel skew. In general, it is needed to use SS/H hardware if sensor signal changes significantly in a time that is less than the channel skew, or if it is required to use a transfer function or perform a frequency domain correlation.

Quantization

As discussed in the previous section, sampling takes a snapshot of the input signal at an instant of time. When the snapshot is taken, the sampled analog signal must be converted from a voltage value to a binary number that the computer can read. The

conversion from infinitely precise amplitude to a binary number is called quantization. During quantization, the A/D converter uses a finite number of evenly spaced values to represent the analog signal. The number of different values is determined by the number of bits used for the conversion. Most modern converters use 12 or 16 bits. Typically, the converter selects the digital value that is closest to the actual sampled value. The figure 2.2 below shows a 1 Hz sine wave quantized by a 3 bit A/D converter.

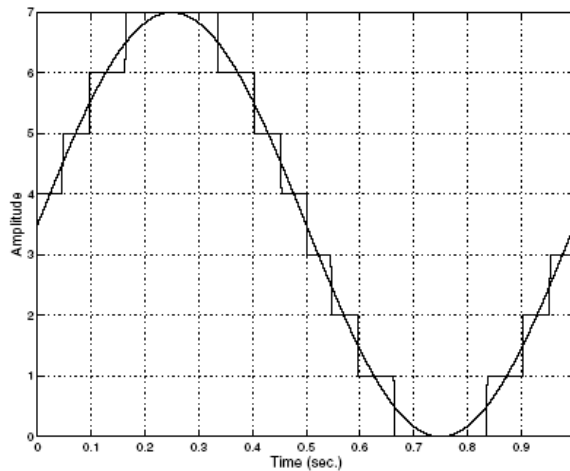


Figure 2.2 Sine wave quantization

Quantization Error

There is always some error associated with the quantization of a continuous signal. Ideally, the maximum quantization error is ± 0.5 least significant bits (LSBs), and over the full input range, the average quantization error is zero. The quantization error for the previous sine wave is calculated by subtracting the actual signal from the quantized signal.

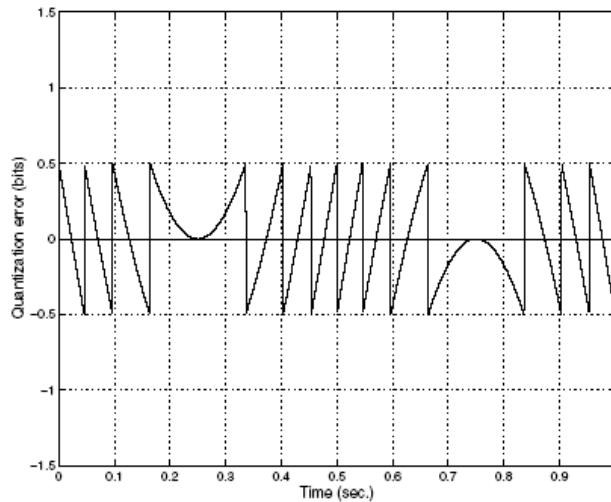


Figure 2.3 Quantization errors for sine wave

Input Range and Polarity

The input range of the analog input subsystem is the span of input values for which a conversion is valid. Input range can be changed by selecting a different gain value. For example, National Instruments' AT-MIO-16E-1 board has eight gain values ranging from 0.5 to 100. Many boards include a programmable gain amplifier that allows user to change the device gain through software. When an input signal exceeds the valid input range of the converter, an over range condition occurs. In this case, most devices saturate to the largest representable value, and the converted data is almost definitely incorrect. The gain setting affects the precision of measurement the higher (lower) the gain value, the lower (higher) the precision. Refer to "How Are Range, Gain, and Measurement Precision Related?"

An analog input subsystem can typically convert both unipolar signals and bipolar signals. A unipolar signal contains only positive values and zero, while a bipolar signal contains positive values, negative values, and zero.

In many cases, the signal polarity is a fixed characteristic of the sensor and input range must be configured to match this polarity. It is crucial to understand the range of signals expected from sensor so that user can configure the input range of the

analog input subsystem to maximize resolution and minimize the chance of an overrange condition.

Clocking of Acquired Samples

Samples are acquired from an analog input subsystem at a specific rate by a clock. Like any timing system, data acquisition clocks are characterized their resolution and accuracy. Timing resolution is defined as the smallest time interval that can be accurately measured. The timing accuracy is affected by clock jitter. Jitter arises when a clock produces slightly different values for a given time interval. For any data acquisition system, there are typically three clock sources: the onboard data acquisition clock, the computer clock, or an external clock. The Data Acquisition Toolbox supports all of these clock sources, depending on the requirements of hardware.

The Onboard Clock.

The onboard clock is typically a timer chip on the hardware board that is programmed to generate a pulse stream at the desired rate. The onboard clock generally has high accuracy and low jitter compared to the computer clock. Onboard clock should always be used when the sampling rate is high and when a fixed time interval between samples are required. The onboard clock is also called as internal clock.

The computer (PC) clock is used for boards that do not possess an onboard clock. The computer clock is less accurate and has more jitter than the onboard clock, and is generally limited to sampling rates below 500 Hz. The computer clock can be called as the software clock.

An external clock is often used when the sampling rate is low and not constant. For example, an external clock source is often used in automotive applications where samples are acquired as a function of crank angle.

Channel Configuration

Input channels can be configured in one of these two ways:

- Differential
- Single-ended

Choice of input channel configuration might depend on whether the input signal is floating or grounded. A floating signal uses an isolated ground reference and is not connected to the building ground. As a result, the input signal and hardware device are not connected to a common reference, which can cause the input signal to exceed the valid range of the hardware device. To circumvent this problem, the signal must be connected to the onboard ground of the device. Examples of floating signal sources include ungrounded thermocouples and battery devices.

A grounded signal is connected to the building ground. As a result, the input signal and hardware device are connected to a common reference. Examples of grounded signal sources include nonisolated instrument outputs and devices that are connected to the building power system.

When hardware is configured for differential input, there are two signal wires associated with each input signal — one for the input signal and one for the reference (return) signal. The measurement is the difference in voltage between the two wires, which helps reduce noise and any voltage that is common to both wires.

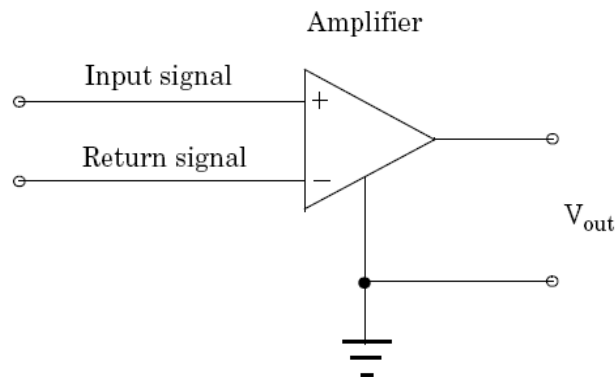


Figure 2.4 Differential input

The input signal is connected to the positive amplifier socket (labeled +) and the return signal is connected to the negative amplifier socket (labeled -). The amplifier has a third connector that allows these signals to be referenced to ground. It is recommended that using differential inputs under any of these conditions:

- The input signal is low level (less than 1 volt).
- The leads connecting the signal are greater than 10 feet.
- The input signal requires a separate ground-reference point or return signal.
- The signal leads travel through a noisy environment.

Single-Ended Inputs

When hardware is configured for single-ended input, there is one signal wire associated with each input signal, and each input signal is connected to the same ground. Single-ended measurements are more susceptible to noise than differential measurements because of differences in the signal paths.

As shown below, the input signal is connected to the positive amplifier socket (labeled +) and the ground is connected to the negative amplifier socket (labeled -).

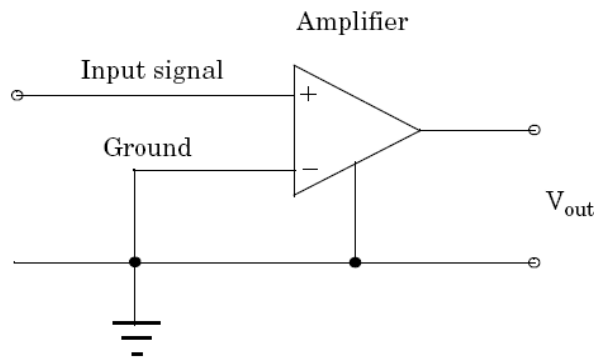


Figure 2.5 Single-Ended input

- The input signal is high level (greater than 1 volt).
- The leads connecting the signal are less than 10 feet.
- The input signal can share a common reference point with other signals. It should

be used differential input connectors for any input signal that does not meet the preceding condition.

There is two different type of single-ended connection;

- Referenced single-ended (RSE) connection

The RSE configuration is used for floating signal sources. In this case, the hardware device itself provides he reference ground for the input signal.

- Nonreferenced single-ended (NRSE) connection

The NRSE input configuration is used for grounded signal sources. In this case, the input signal provides its own reference ground and the hardware device should not supply one.

Transferring Data from Hardware to System Memory

The transfer of acquired data from the hardware to system memory follows these steps:

- 1-Acquired data is stored in the hardware's first-in first-out (FIFO) buffer.
- 2-Data is transferred from the FIFO buffer to system memory using interrupts or DMA.

These steps happen automatically. Typically, all that's required from user is some initial configuration of the hardware device when it is installed.

The FIFO buffer is used to temporarily store acquired data. The data is temporarily stored until it can be transferred to system memory. The process of transferring data into and out of an analog input FIFO buffer is given below:

- 1-The FIFO buffer stores newly acquired samples at a constant sampling rate.
- 2-Before the FIFO buffer is filled, the software starts removing the samples.

For example, an interrupt is generated when the FIFO is half full, and signals the software to extract the samples as quickly as possible.

3-Because servicing interrupts or programming the DMA controller can take up to a few milliseconds, additional data is stored in the FIFO for future retrieval. For a larger FIFO buffer, longer latencies can be tolerated.

4-The samples are transferred to system memory via the system bus (for example, PCI bus or AT bus). After the samples are transferred, the software is free to perform other tasks until the next interrupt occurs. For example, the data can be processed or saved to a disk file. As long as the average rates of storing and extracting data are equal, acquired data will not be missed and application should run smoothly.

Interrupts

The slowest but most common method to move acquired data to system memory is for the board to generate an interrupt request (IRQ) signal. This signal can be generated when one sample is acquired or when multiple samples are acquired. The process of transferring data to system memory via interrupts is given below:

1-When data is ready for transfer, the CPU stops whatever it is doing and runs a special interrupt handler routine that saves the current machine registers, and then sets them to access the board.

2-The data is extracted from the board and placed into system memory.

3-The saved machine registers are restored, and the CPU returns to the original interrupted process.

The actual data move is quick, but there is a lot of overhead time spent saving, setting up, and restoring the register information. Therefore, depending on specific system, transferring data by interrupts might not be a good choice when the sampling rate is greater than around 5 kHz.

DMA

Direct memory access (DMA) is a system whereby samples are automatically stored in system memory while the processor does something else. The process of

transferring data via DMA is given below:

- 1- When data is ready for transfer; the board directs the system DMA controller to put it into in system memory as soon as possible.
- 2- As soon as the CPU is able (which is usually very quickly), it stops interacting with the data acquisition hardware and the DMA controller moves the data directly into memory.
- 3- The DMA controller gets ready for the next sample by pointing to the next open memory location.
- 4- The previous steps are repeated indefinitely, with data going to each open memory location in a continuously circulating buffer. No interaction between the CPU and the board is needed. Computer supports several different DMA channels. Depending on users application, it can be used one or more of these channels, For example, simultaneous input and output with a sound card requires one DMA channel for the input and another DMA channel for the output.

2.1.2 Analog Output Subsystems

Analog output subsystems convert digital data stored on computer to a real-world analog signal. These subsystems perform the inverse conversion of analog input subsystems. Typical acquisition boards offer two output channels with 12 bits of resolution, with special hardware available to support multiple channel analog output operations. Analog output subsystems are also referred to as AO subsystems, D/A converters, or DACs.

2.1.3 Digital Input/Output Subsystems

Digital input/output (DIO) subsystems are designed to input and output digital values (logic levels) to and from hardware. These values are typically handled either as single bits or *lines*, or as a *port*, which typically consists of eight lines. While most popular data acquisition cards include some digital I/O capability, it is usually limited to simple operations, and special dedicated hardware is often necessary for performing advanced digital I/O operations.

2.1.4 Counter/Timer Subsystems

Counter/timer (C/T) subsystems are used for event counting, frequency and period measurement, and pulse train generation.

2.2 Sensors

A sensor converts the physical phenomena of interest into a signal that is input into data acquisition hardware. There are two main types of sensors based on the output they produce: digital sensors and analog sensors. Digital sensors produce an output signal that is a digital representation of the input signal, and has discrete values of magnitude measured at discrete times. A digital sensor must output logic levels that are compatible with the digital receiver. Some standard logic levels include transistor-transistor logic (TTL) and emitter-coupled logic (ECL). Examples of digital sensors include switches and position encoders. Analog sensors produce an output signal that is directly proportional to the input signal, and is continuous in both magnitude and in time. Most physical variables such as temperature, pressure, and acceleration are continuous in nature and are readily measured with an analog sensor. For example, the temperature of an automobile cooling system and the acceleration produced by a child on a swing all vary continuously. The sensor depends on the phenomenon that is measuring.

When choosing the best analog sensor to use, the characteristics of the physical variable must be matched measuring with the characteristics of the sensor. The two most important sensor characteristics are current signals and voltage signals. Current signals are expressed below.

Current Signals

Current is often used to transmit signals in noisy environments because it is much less affected by environmental noise. The full scale range of the current signal is often either 4-20 mA or 0-20 mA. A 4-20 mA signal has the advantage that even at minimum signal value, there should be a detectable current flowing. The absence of this indicates a wiring problem.

Before conversion by the analog input subsystem, the current signals are usually turned into voltage signals by a current-sensing resistor. The resistor should be of high precision, perhaps 0.03% or 0.01% depending on the resolution of hardware. Additionally, the voltage signal should match the signal to an input range of the analog input hardware. For 4-20 mA signals, a 50 ohm resistor will give a voltage of 1 V for a 20 mA signal by Ohm's law.

The most commonly interfaced signal is a voltage signal. For example, thermocouples, strain gauges, and accelerometers all produce voltage signals. There are three major aspects of a voltage signal that need to be considered.

If the signal is smaller than a few millivolts, it might be required to amplify it. If it is larger than the maximum range of analog input hardware (typically ± 10 V), it will have to divide the signal down using a resistor network.

Whenever data acquired, the highest frequency must be decided that want to be measured.

The highest frequency component of the signal determines how often user should sample the input. If user have more than one input, but only one analog input subsystem, then the overall sampling rate goes up in proportion to the number of inputs. Higher frequencies might be present as noise, which user can remove by filtering the signal before it is digitized. If user samples the input signal at least twice as fast as the highest frequency component, then that signal will be uniquely characterized. However, this rate might not mimic the waveform very closely. For a rapidly varying signal, user might need a sampling rate of roughly 10 to 20 times the highest frequency to get an accurate picture of the waveform. For slowly varying signals, user need only consider the minimum time for a significant change in the signal.

Duration

How long do users want to sample the signal for? If user is storing data to memory or to a disk file, then the duration determines the storage resources required. The format of the stored data also affects the amount of storage space required. For example,

data stored in ASCII format takes more space than data stored in binary format.

Sensor Bandwidth

In a real-world data acquisition experiment, the physical phenomenon user is measuring has expected limits. For example, the temperature of automobile's cooling system varies continuously between its low limit and high limit. The temperature limits, as well as how rapidly the temperature varies between the limits, depends on several factors including driving habits, the weather, and the condition of the cooling system. The expected limits might be readily approximated, but there are an infinite number of possible temperatures that user can measure at a given time.

The bandwidth is given by the range of frequencies present in the signal being measured. User can also think of bandwidth as being related to the rate of change of the signal. A slowly varying signal has a low bandwidth, while a rapidly varying signal has a high bandwidth. To properly measure the physical phenomena of interest, the sensor bandwidth must be compatible with the measurement bandwidth. User might want to use sensors with the widest possible bandwidth when making any physical measurement. This is the one way to ensure that the basic measurement system is capable of responding linearly over the full range of interest. However, the wider the bandwidth of the sensor, the more users must be concerned with eliminating sensor response to unwanted frequency components.

2.3 Signal Conditioning

Sensor signals are often incompatible with data acquisition hardware. To overcome this incompatibility, the sensor signal must be conditioned. The type of signal conditioning required depends on the sensor user are using. For example, a signal might have small amplitude and require amplification, or it might contain unwanted frequency components and require filtering.

Common ways to condition signals include

- Amplification
- Filtering
- Electrical isolation
- Multiplexing
- Excitation source

2.3.1 Amplification

Low-level signals — less than around 100 milivolts — usually need to be amplified. High-level signals might also require amplification depending on the input range of the analog input subsystem. For example, the output signal from a thermocouple is small and must be amplified before it is digitized. Signal amplification allows user to reduce noise and to make use of the full range of hardware thereby increasing the resolution of the measurement.

2.3.2 Filtering

Filtering removes unwanted noise from the signal of interest. A noise filter issued on slowly varying signals such as temperature to attenuate higher frequency signals that can reduce the accuracy of measurement. Rapidly varying signals such as vibration often require a different type of filter known as an antialiasing filter. An antialiasing filter removes undesirable higher frequencies that might lead to erroneous measurements.

2.3.3 Electrical Isolation

If the signal of interest contains high-voltage transients that could damage the computer, then the sensor signals should be electrically isolated from the computer for safety purposes.

User can also use electrical isolation to make sure that the readings from the data acquisition hardware are not affected by differences in ground potentials. For example, when the hardware device and the sensor signal are each referenced to ground, problems occur if there is a potential difference between the two grounds.

This difference can lead to a ground loop, which might lead to erroneous measurements. Using electrically isolated signal conditioning modules eliminates the ground loop and ensures that the signals are accurately represented.

2.3.4 Multiplexing

A common technique for measuring several signals with a single measuring device is multiplexing.

Signal conditioning devices for analog signals often provide multiplexing for use with slowly changing signals such as temperature. This is in addition to any built-in multiplexing on the DAQ board. The A/D converter samples one channel, switches to the next channel and samples it, switches to the next channel, and so on. Because the same A/D converter is sampling many channels, the effective sampling rate of each individual channel is inversely proportional to the number of channels sampled.

User must take care when using multiplexers so that the switched signal has sufficient time to settle.

2.3.5 Excitation Source

Some sensors require an excitation source to operate. For example, strain gauges, and resistive temperature devices (RTDs) require external voltage or current excitation. Signal conditioning modules for these sensors usually provide the necessary excitation. RTD measurements are usually made with a current source that converts the variation in resistance to a measurable voltage.

2.4. The Computer

The computer provides a processor, a system clock, a bus to transfer data, and memory and disk space to store data. The processor controls how fast data is accepted by the converter. The system clock provides time information about the acquired data. Knowing that user recorded a sensor reading is generally not enough. Users also need to know when that measurement occurred.

Data is transferred from the hardware to system memory via dynamic memory access (DMA) or interrupts. DMA is hardware controlled and therefore extremely fast. Interrupts might be slow because of the latency time between when a board requests interrupt servicing and when the computer responds.

The maximum acquisition rate is also determined by the computer's bus architecture.

2.5. Software

Regardless of the hardware user are using, user must send information to the hardware and receive information from the hardware. User sends configuration information to the hardware such as the sampling rate, and receives information from the hardware such as data, status messages, and error messages. User might also need to supply the hardware with information so that user can integrate it with other hardware and with computer resources. This information exchange is accomplished with software. There are two kinds of software:

- Driver software
- Application software

CHAPTER 3

ADVANTECH DATA ACQUISITION BOARDS

In this study, the Advantech multifunctional data acquisition board is used to acquire data. There are many data acquisition hardware manufacturers in the electronic market. Standard multifunctional cards have analog inputs/outputs, digital inputs outputs. The features of Advantech multifunctional data acquisition board is give below.

16-channel Single-Ended A/D Input

12-bit A/D conversion with up to 100 kHz sampling rate

16-channel Digital Input

16-channel Digital Output

Programmable Counter/Timer

Automatic Channel/Gain Scanning

The Advantech PCI-1711 is a Plug-and-Play device, which fully complies with the PCI Specification Rev 2.1. During card installation, all bus-related configurations such as base I/O address and interrupts are conveniently taken care of by the Plug-and-Play function. Users have virtually no need to set any jumpers or DIP switches.

The PCI-1711 features an automatic channel/gain scanning circuit. This circuit design controls multiplexer switching during sampling. Users can set different gain values for each channel according to their needs for the corresponding range of input voltage. The gain value settings thus selected is stored in the SRAM. This flexible design enables multi-channel and high-speed sampling for high-performance data acquisition (up to 100 KS/s.)

The PCI-1711/1731 provides an on-board FIFO memory buffer, storing up 1 to 1K A/D samples. Users can either enable or disable the interrupt request feature of the FIFO buffer. While the interrupt request for FIFO is enabled, users are allowed to specify whether an interrupt request will be sent with each sampling action or only

when the FIFO buffer is half saturated. This useful feature enables a continuous high-speed data transfer with a more predictable performance on operating systems.

The PCI-1711/1731 provides 16 digital input channels and 16 digital output channels. Users are left with great flexibility to design and customize their applications according to their specific needs.

The PCI-1711/1731 is equipped with a programmable counter, which can serve as a pacer trigger for A/D conversions. The counter chip is an 82C54 or its equivalent, which incorporates three 16-bit counters on a 10 MHz clock. One of the three counters is used as an event counter for input channels or pulse generation. The other two are cascaded into a 32-bit timer for pacer triggering.

3.1 Device Testing

Following through the Setup and Configuration procedures to the last step described in the previous section, user can now proceed to test the device by clicking the Test Button on the I/O Device Installation

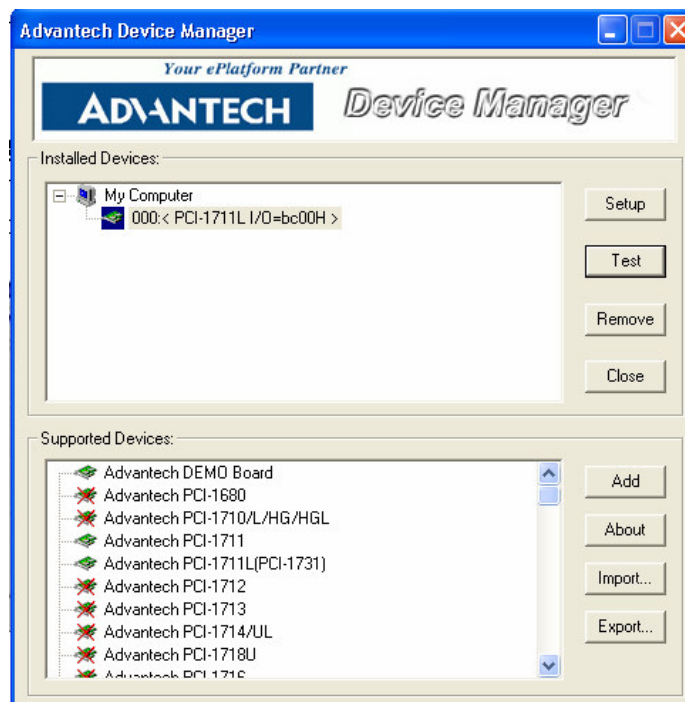


Figure 3.1 Advantech test window

On the Device Test dialog box, users are free to test various functions of PCI-1711/1731 on the Analog input, Analog output, Digital input, Digital output or Counter tabs.

3.1.1 Testing Analog Input Function

Click the Analog Input tab to bring it up to front of the screen. Select the input range for each channel in the Input range drop-down boxes. Configure the sampling rate on the scroll bar. Switch the channels by using the up/down arrow.

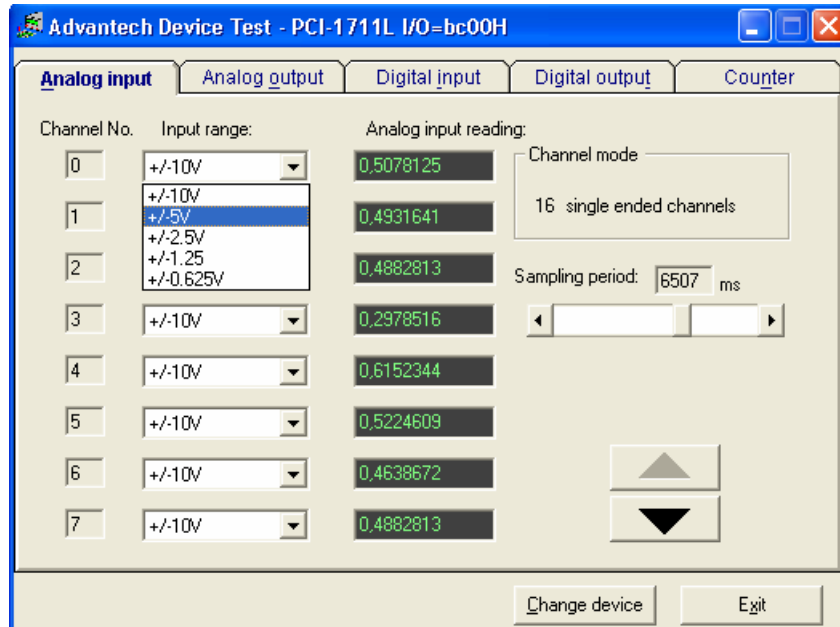


Figure 3.2 Analog input test

3.1.2 Testing Digital Input Function

Click the Digital Input tab to show forth the Digital Input test panel as seen below. Through the color of the lamps, users can easily discern whether the status of each digital input channel is either high or low. Digital Input tab on the Device Test dialog box is shown below in Figure 3.3.

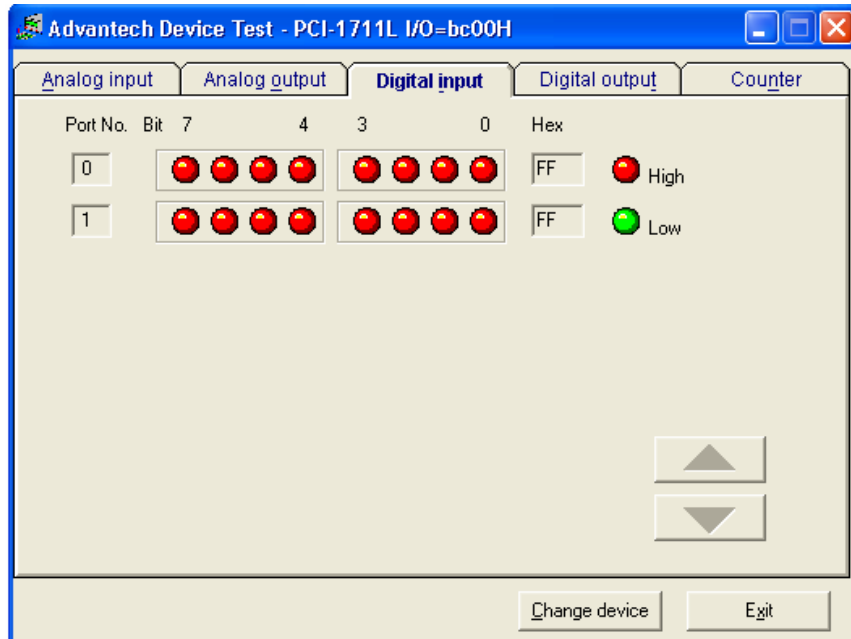


Figure 3.3 Digital input tests

3.1.3 Testing Digital Output Function

Click the Digital Output tab to bring up the Digital Output test panel such as seen on the next page. By pressing the buttons on each tab, users can easily set each digital output channel as high or low for the corresponding port.

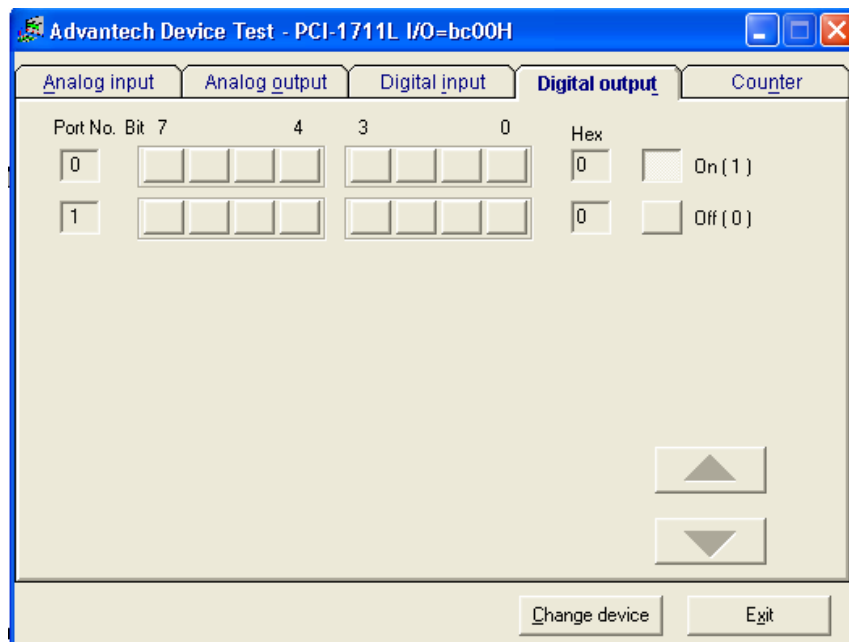


Figure 3.4 Digital output test

3.1.4 Testing Counter Function

Click the Counter Tab to bring its test panel forth. The counter channel (Channel 0) offers the users two options: Event counting and Pulse out. If user select Event counting, user need first to connect clock source to pin CNT0_CLK, and the counter will start counting after the pin CNT0_GATE is triggered. If user selects Pulse Out, the clock source will be output to pin CNT0_OUT. User can configure the Pulse Frequency by the scroll bar right below it.

Only after card device is properly set up, configured and tested, can the device installation procedure be counted as complete. After the device installation procedure is completed, user can now safely proceed to the next chapter, Signal Connections.

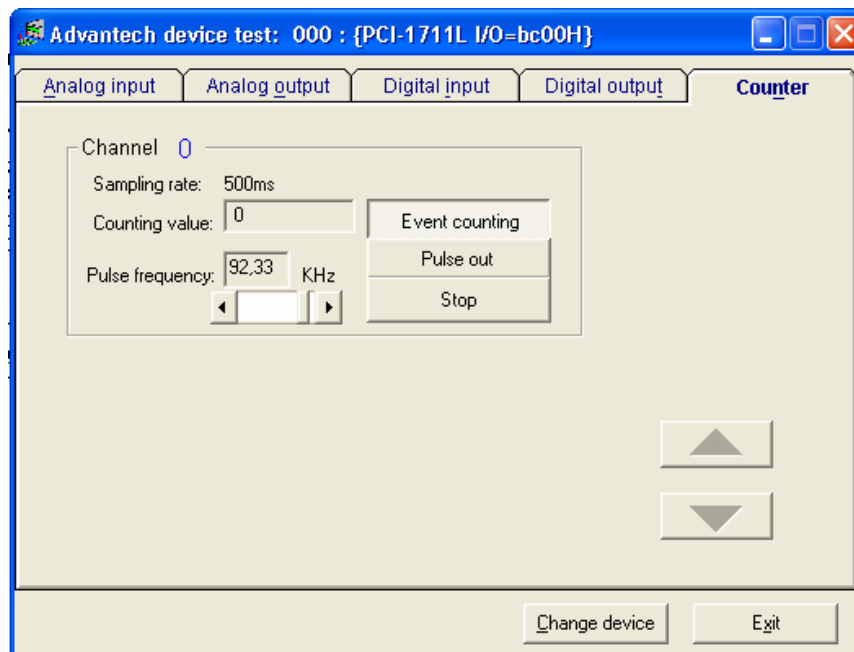


Figure 3.5 Counter test

3.2 Analog Input Connections

The PCI-1711/1731 supports 16 single-ended analog inputs. Each individual input channel is software-selected.

The single-ended input configuration has only one signal wire for each channel, and the measured voltage (V_m) is the voltage of the wire as referenced against the common ground.

A signal source without a local ground is also called a “floating source”. It is fairly simple to connect a single-ended channel to a floating signal source. In this mode, the PCI-1711/1731 provides a reference ground for external floating signal sources.

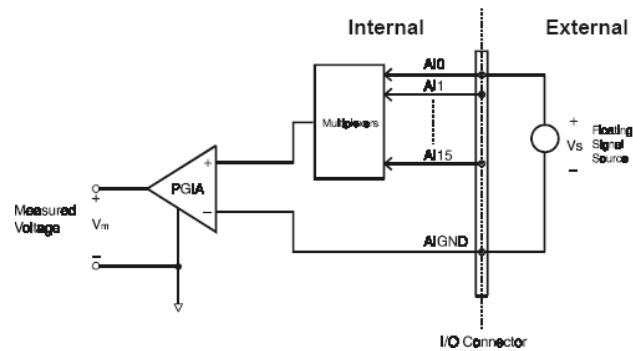


Figure 3.6 Single Ended Connections

3.3 Calibration

This chapter offers user a brief guide to the calibration programs. Regular calibration checks are important to maintain accuracy in data acquisition and control applications. Three calibration programs are included on the companion CD-ROM:

ADCAL.EXE A/D calibration program

DACAL.EXE D/A calibration program (Only for PCI-1711)

SELFCAL.EXE A/D self-calibration program (Only for PCI-1711)

Those calibration programs are designed only for DOS environment. User can access these programs by the default installation path:

C:\Program Files\Advantech\Adsapi\Utility\PCI1711

There are four variable resistors (VRs) on the PCI-1711 card and two variable resistors (VRs) on the PCI-1731 card. These variable resistors are to facilitate accurate adjustments for all A/D and D/A channels.

Please refer to the following two figures for the VR positions.

Fig. 5-1 PCI-1711 VR assignment

VR3 VR4 VR1 VR2

VR Function

VR1 A/D bipolar offset adjustment

VR2 A/D full scale (gain) adjustment

VR3 D/A channel 0 full scale adjustment

VR4 D/A channel 1 full scale adjustment

Fig. 5-2 PCI-1731 VR assignment

The following list shows the function of each VR:

Regular and accurate calibration procedures ensure the maximum possible accuracy. The A/D calibration program ADCAL.EXE leads user through the whole A/D offset and gain adjustment procedure. The basic steps are outlined below:

1. Connect a DC voltage source of +9.995 V to AI0.
2. Connect AGND to AI1, AI2, AI3, AI4 and AI5.
3. Run the ADCAL.EXE program.
4. Adjust VR2 until the output codes from the card's AI0 are focused on FFE (at least 70%), and adjust VR1 until the output codes from the card's AI1, AI2, AI3, AI4 and AI5 are focused on 7FF (at least 70 %).
5. Press the SPACE key to finish A/D calibration.

Self A/D Calibration

We know, in most cases, it is difficult to find a good enough DC voltage source for A/D calibration. We provide a self-adjusted A/D calibration program "SELCAL.EXE" to help solve this problem. The steps of self-calibration are outlined as below:

1. Connect DA0_OUT to AI0.
2. Connect AGND to AI1, AI2, AI3, AI4 and AI5.
3. Run the SELFCAL.EXE program.
4. First calibrate the D/A channel. Adjust VR3 until the DA0_OUT Output voltage approaches +10V. Then press the SPACE key.
5. Next we will do the A/D calibration. Now the DA0_OUT output voltage will be +9.995V, then adjust VR2 until the output codes from the card's AI0 focused on FFE (at least 70%) and adjust VR1 until the output codes from the card's AI1, AI2, AI3, AI4 and AI5 focused on 7FF (at least 70%).
6. Press the SPACE key to finish calibration procedures.

CHAPTER 4

DATA ACQUISITION CONTROL PANEL

4.1 Data Acquisition Toolbox

The Data Acquisition Toolbox provides a complete set of tools for analog input, analog output, and digital I/O from a variety of PC-compatible data acquisition hardware. The toolbox lets user configure external hardware devices, read data into MATLAB for analysis, and send out data. Based on MATLAB, the toolbox enables user to easily customize acquisitions, access the built-in features of hardware devices, and incorporate the analysis and visualization features of MATLAB and related toolboxes into user design. Together, MATLAB and the Data Acquisition Toolbox offer a single, integrated environment to support the entire data acquisition and analysis process. User can easily analyze or visualize data, save it for post-processing, and make iterative updates to test setup based on analysis results.

The Data Acquisition Toolbox uses the MATLAB object technology to expose the configurable features of each manufacturer's hardware devices. As a result, user can access and control individual board features by using MATLAB get and set commands. Adaptor Kit The Data Acquisition adaptor kit enables user to build custom interfaces to non-supported boards. The kit is included with the Data Acquisition Toolbox for interfacing to unsupported boards. It includes step-by-step documentation and diagnostic tools. Working with the Data Acquisition Toolbox The Data Acquisition Toolbox is a set of M-file and MEX-file functions that user can execute from the command line or through MATLAB programs. Based on the MATLAB object technology, the toolbox provides functions for creating device objects that are directly associated with hardware. These objects include a collection of base properties that apply to all supported hardware, such as sample rate, trigger settings, and channel properties. They also contain device-specific properties that expose the features and capabilities unique to hardware. Using Device Objects Device objects provide a gateway to the hardware's functionality and enable user to control the behavior of acquisition. For example, an analog input object created in

MATLAB contains all the parameters necessary for executing any supported analog input task. Analog Input The analog input functions enable user to create an analog input object, add channels to the object, read data into the workspace, and preview the most recently acquired data. Analog Output Analog output functions let user send data out to hardware. These functions enable user to create the object, add channels, queue data sets to be output, and send out sample data. Digital I/O Specific functions enable user to communicate with digital I/O hardware. User can create device objects, add lines, send data to the hardware, and read data into the workspace. Channels and Lines. The Data Acquisition Toolbox channels and lines are mapped to hardware's channels and lines. The toolbox supports an unlimited number of channels/lines, enabling user to use as many as hardware requires. Controlling Acquisition. The Data Acquisition Toolbox supports a wide range of functions for controlling acquisition. For example, user can set event information, evaluate the acquisition status, define triggers and callbacks, preview data while the device is running, and perform analysis on the fly. The toolbox also supports several hardware-specific properties that can be easily displayed and customized to meet specifications. Managing Data Functions are provided for previewing and extracting data for analysis. The toolbox streams data into MATLAB in double-precision floating format. This lets user work with the data just as user would with any other matrix in MATLAB. User can also stream in data using a native hardware format. Logging Data Functions are available for logging data to disk, memory, or both while the analog input object is running. The Data Acquisition Toolbox enables user to log data, events, and errors. In addition, functions are provided to extract data from toolbox-generated log files. SoftScope is a graphical user interface for selecting and configuring data acquisition sources and then acquiring, viewing, and analyzing data using a familiar, oscilloscope like interface. SoftScope lets user quickly verify hardware operation and perform live data. The toolbox supports three device objects: analog input, analog output, and digital I/O. The Data Acquisition Toolbox automatically performs A/D and D/A data conversions for receiving or sending data. While the Data Acquisition Toolbox is collecting data, MATLAB can analyze and visualize data. Two channels of live audio data displayed and analyzed using the SoftScope software oscilloscope interface. SoftScope's measurement capabilities find the peak voltage of data streaming in through channel 1. Analysis

The Data Acquisition Toolbox is a collection of M-file functions and MEX-file dynamic link libraries (DLLs) built on the MATLAB technical computing environment. The toolbox provides user with these main features: •A framework for bringing live, measured data into MATLAB using PC-compatible, plug-in data acquisition hardware

- Support for analog input (AI), analog output (AO), and digital I/O (DIO) subsystems including simultaneous analog I/O conversions
- Support for these popular hardware vendors/devices:
 - Advantech boards that use the Advantech Device Manager
 - Agilent Technologies E1432A/33A/34A VXI modules
 - Keithley boards that use DriverLINX drivers.
 - Measurement Computing Corporation (ComputerBoards) boards
 - National Instruments boards that use NI-DAQ software (except SCXI)
 - Parallel ports LPT1–LPT3
 - Windows sound cards

Additionally, user can use the Data Acquisition Toolbox Adaptor Kit to interface unsupported hardware devices to the toolbox.

- Event-driven acquisitions

A list of the toolbox functions is available to user by typing `help daq`. It can be viewed the code for any function by typing `type function_name`.

User can change the way any toolbox function works by copying and renaming the M-file, then modifying copy. User can also extend the toolbox by adding own M-files, or by using it in combination with other products such as the Signal Processing Toolbox or the Instrument Control Toolbox. The MathWorks provides several related products that are especially relevant to the kinds of tasks user can perform with the Data Acquisition Toolbox.

4.2 Software Requirements of Data Acquisition Control Panel

Data Acquisition Control Panel program can only work under the software platforms windows XP / NT. Windows XP and NT can recognize PCI slot data acquisition boards as play and plug. The interfaces DLL' (Dynamic Link Library) will automatically installed through their path of directories by installation wizard. Version of Data acquisition toolbox of Matlab must be greater than version 2.5.

4.3. Softscope

The data acquisition Oscilloscope is an interactive graphical user interface (GUI) for streaming data into a display.

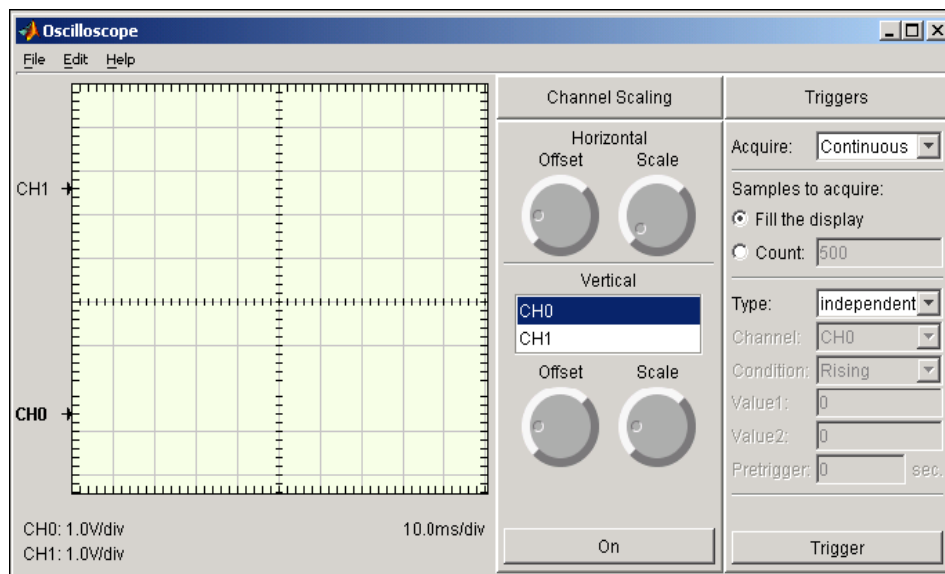


Figure 4.1 Two-channel Softscope screen

4.3.1 Displaying Channels

Click the **Trigger** button to begin streaming data into the display. The data from each channel defines a unique trace (line). To quickly scale the data, right-click the display and select **Autoscale** from the menu.

The display area contains this information:

- Labels and markers for each trace. For this example, the traces are labeled CH0 and CH1.
- Labels for the vertical units for each trace, and a label for the horizontal units for the display.

When the acquisition is not running, user can display data tips by moving the mouse cursor over the trace. The data tip is indicated by a red circle, and displays the value of the trace at the selected point. If user presses the Control key while the cursor is over the trace, the difference between the first data tip and the last data tip is displayed. Click the **Trigger** button to begin streaming data into the display. Display data tips by placing the mouse cursor over the trace.

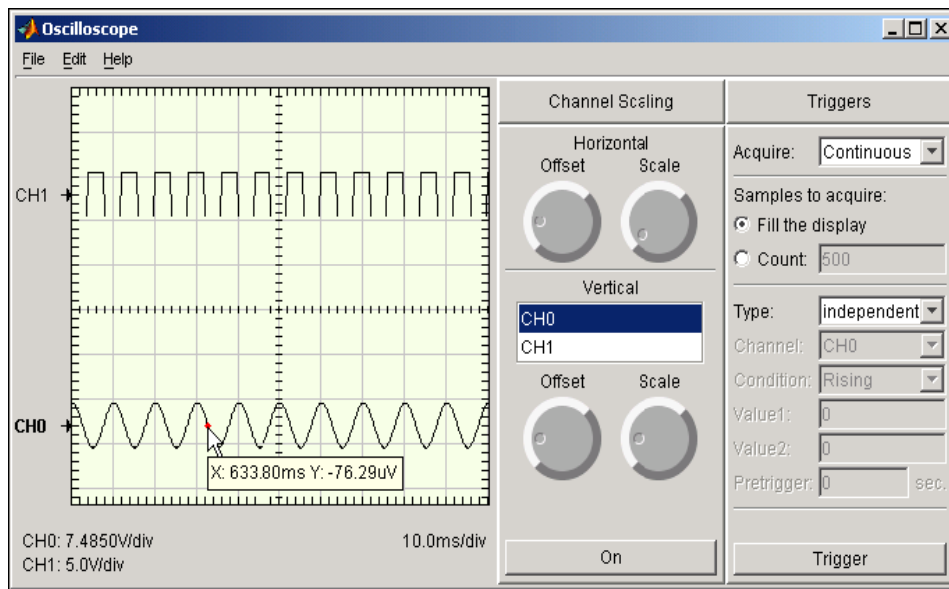


Figure 4.2 Displayed channels

4.3.2 Creating Additional Displays

To add additional displays to the Oscilloscope, use the Scope pane of the Scope Editor GUI. To open this GUI, select Scope from the Edit menu. As shown below, the new display is named display2.

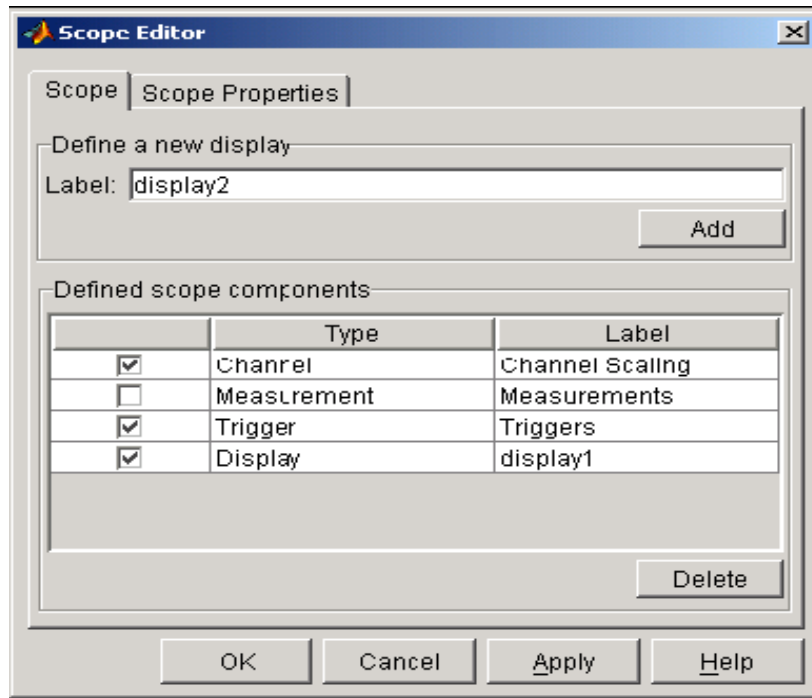


Figure 4.3 Scope editor screen

To show a trace in a particular display, use the Channel Display pane of the Channel Editor GUI. To open this GUI, select Channel from the Edit menu. As shown below, CH0 is associated with the new display.

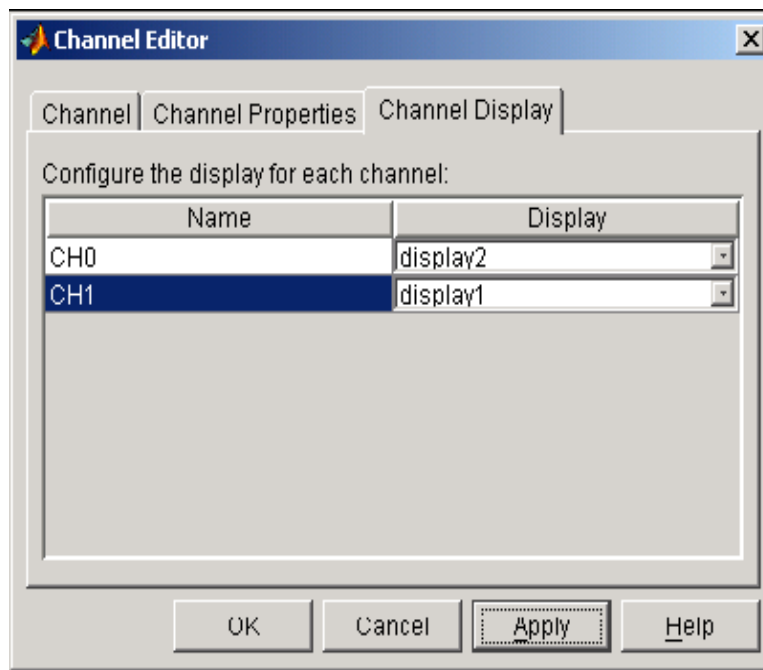


Figure 4.4 Scope channel editor

The Oscilloscope is now configured so that the CH0 trace is shown in the bottom display, and the CH1 trace is shown in the Figure 4.5.

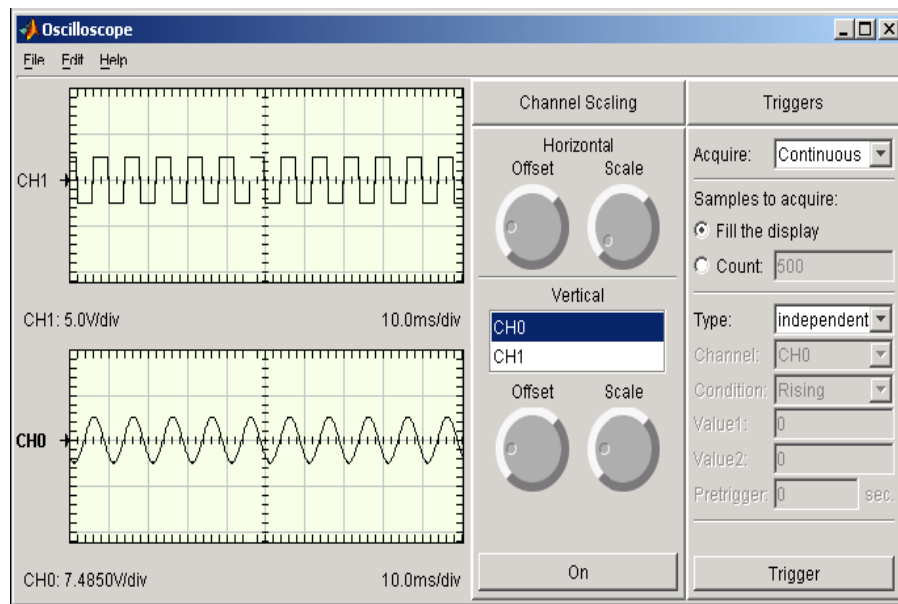


Figure 4.5 Separated channel displays

4.3.3 Configuring Display Properties

User can change the display characteristics of the Oscilloscope by configuring display properties. Users access the display properties these two ways:

- Property Inspector — Place the mouse cursor in the display of interest, right-click, and select Edit Properties from the menu.
- Scope Editor GUI — Select Scope from the Edit menu, and then choose the Scope Properties pane.

For this example, use the Scope Editor GUI to change the color of both displays to white. The steps are

1. Select both displays from the Select the scope components list.
2. Open the color picker for the Color property.
3. Select White from the color picker pop-up menu.

The Scope Properties pane and color picker are shown below. For descriptions of all display properties, click the Help button.

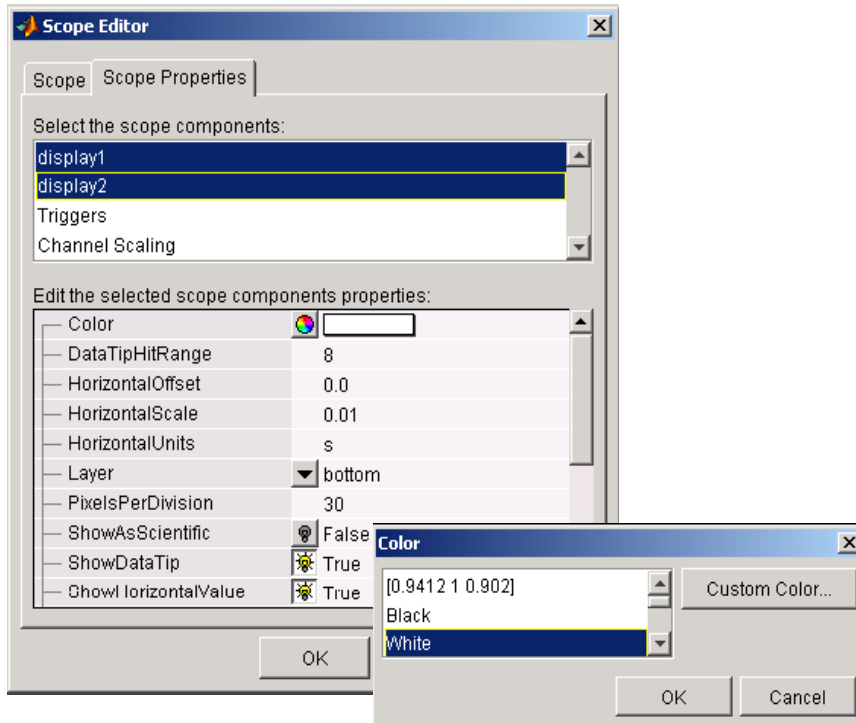


Figure 4.6 Scope optional editor screen

4.3.4 Scaling the Channel Data

User can scale the defined channels using the Channel Scaling panel. In particular, user can modify

- The horizontal scaling and offset for all display components.
- The vertical scaling and offset for one or more channels.

To simultaneously modify the vertical scaling for multiple channels, select the desired channel names in the list box. Additionally, using the On/Off button, user can add or remove the selected traces from the Oscilloscope. As shown below, the horizontal scale is changed to approximately 5 ms/div, and the vertical scale is modified to maximize the trace amplitudes. Note that the horizontal and vertical

scaling information is shown at the bottom of each display component.

To specify a precise horizontal scale or offset, users modify the associated display properties. To specify a precise vertical scale or offset, users modify the associated channel properties. User can access these properties using the Scope Editor and the Channel Editor, respectively. User open these editors with the Edit menu or a right-click menu. Note that all displays use the same horizontal offset and scale.

4.3.5 Triggering the Oscilloscope

To display acquired data in the Oscilloscope, user must click the Trigger button. User control how the data acquisition is initiated by specifying the acquisition type and the trigger type in the Trigger panel.

Acquisition Types

The Oscilloscope supports three acquisition types, which can be selected from the **Acquire** menu:

- **One Shot** — acquire the specified number of samples once.
- **Continuous** — continuously acquire the specified number of samples.
- **Sequence** — continuously acquire the specified number of samples, and use the dependent trigger type each time. For each acquisition type, user can either fill the display with data or user can acquire a specific number of samples. Additionally, the specified trigger type (see below) determines how the acquisition is initiated.

The Oscilloscope supports two trigger types, which can be selected from the **Type** menu:

- **Dependent** — Data acquisition depends on the data. Users define this dependency by specifying the hardware channel, trigger condition, trigger condition value, and whether pretrigger data is acquired. Note that user can specify a dependent trigger for only one channel at a time, and this channel initiates data acquisition for all other

channels defined for the Oscilloscope.

- **Independent** — Data acquisition starts immediately after user press the **Trigger** button, and is independent of the data. Note that the **Sequence** acquisition does not support this trigger type.

The Oscilloscope shown below is configured for a one-shot acquisition of 1000 samples for CH0 and CH1. The acquisition is dependent on the data, and is initiated when a rising signal level of -3.3 volts is detected on CH0. Additionally, the first 0.02 second of data is defined as pretrigger data.

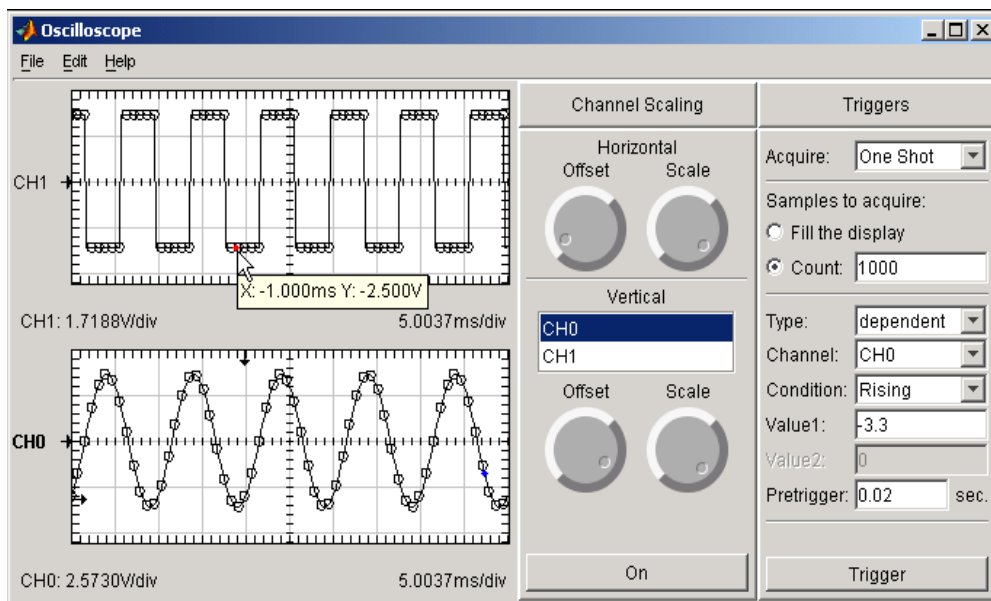


Figure 4.7 Channel Edit

When user use a dependent trigger type, the display associated with the selected channel contains these two indicators:

- The trigger level on the vertical axis.
- The location of the start of the trigger on the horizontal axis. The start of the trigger corresponds to the first acquired sample at time zero. As shown by the data tips for CH1, data to the left of the indicator is defined as pretrigger data and has negative time values. Note that user can change the indicator locations graphically by placing the mouse cursor over the indicator and sliding it to the desired location.

4.3.6 Making Measurements

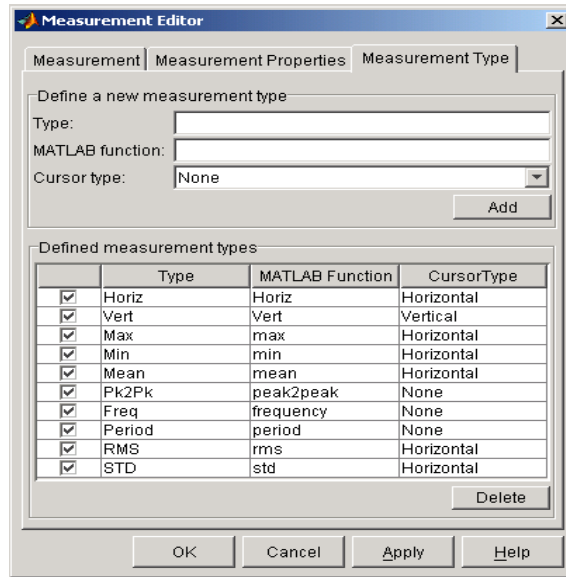


Figure4.8 Measurement editor

User can make measurements on the acquired data with the Measurements panel. The Oscilloscope provides many predefined measurement types such as horizontal and vertical cursors, and basic math calculations such as the mean and standard deviation. Additionally, user can define new measurement types that suit specific needs. As shown below, user can list the predefined measurement types and create a new measurement type with the Measurement Type pane of the Measurement Editor GUI

Measurements that user define for the Oscilloscope are displayed in the Measurements panel. By default, this panel is not included as part of the Oscilloscope. To create the panel, users define one or more initial measurements.

There are two ways to do this:

- Right-click in the Channel Scaling panel and select Add Measurement from the menu.
- Use the Measurement Editor GUI, which user open by selecting the Edit->Measurement menu item. Alternatively, user can create an empty Measurements panel by selecting the

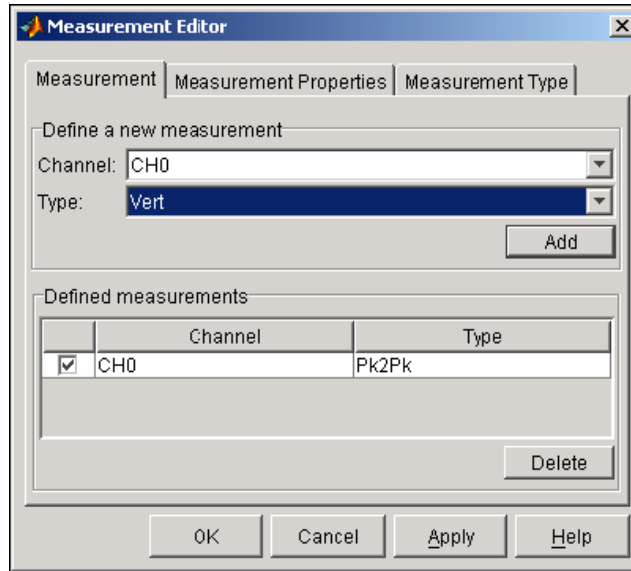


Figure 4.9 Measurement properties screen

The Measurement pane shown in Figure 4.9 is configured to add a vertical cursor measurement for CH0 to the Oscilloscope. Note that the peak-to-peak measurement is already defined for CH0.

After user click the OK or Apply button of the Measurement Editor, the Measurements panel is automatically added to the Oscilloscope. User can then click the Add Measurement button to define additional measurements.

4.3.7 Exporting Data

User can export this information to the MATLAB workspace, a figure, or a MAT-file:

- Channel data — Data associated with a hardware channel, a math channel, or a reference channel.
- Measurements — Data associated with a defined measurement. Note that some measurements such as the horizontal and the vertical cursor have no data to save.

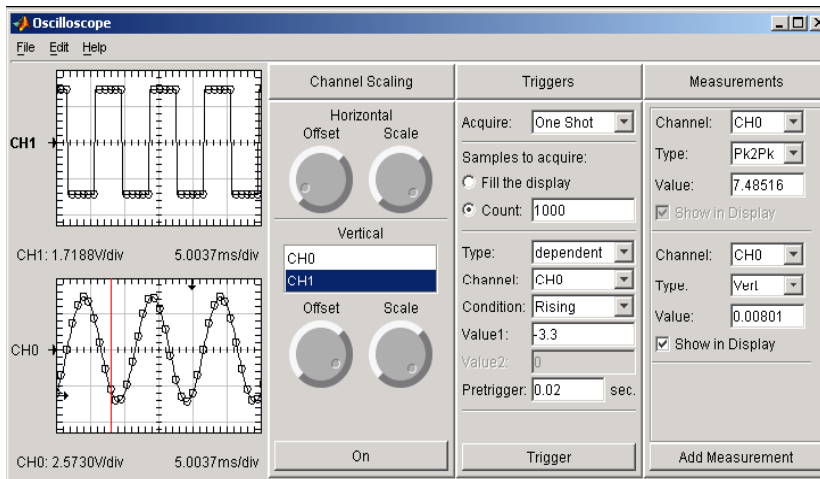


Figure 4.10 Exporting Channels data

Channel data can be exported with the Channel Exporter GUI, which user open by selecting the **File>Export->Channels** menu item. The GUI shown below is configured to export 1000 samples for both hardware channels to the workspace as a structure, which contains horizontal and vertical scaling information. The variable name for the CH0 data is c0 and the variable name for the CH1 data is c1.

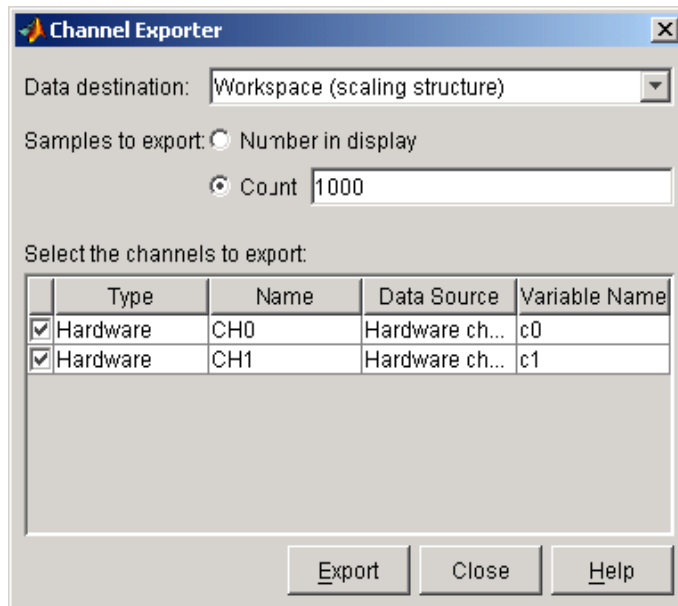


Figure 4.11 Channel editor

The saved structure is shown below, where t0 is the time of the first stored sample. Note that the time is negative because pretrigger data was acquired.

```
c0
c0 =
horizontalScale: 0.0050
horizontalOffset: 0
verticalScale: 2.5730
verticalOffset: 0
data: [1000x1 double]
t0: -0.0200
samplerate: 5000
```

4.4 The Use of Data Acquisition Control Panel

Data acquisition Control Panel is a interface program that helps user to initialize their hardwares. Data acquisition control panel interfaces between Data Acquisition Toolbox of Matlab, Advantech data acquisition hardware and softscope (digital oscilloscope program for Matlab).

In the process of data acquisition, the first step is to set up hardwares and required cablings. Analog inputs must be configured by written commands on Matlab to be able to acquire data from hardware.Interface (Adaptors) and DLL (Dynamic Link Libraries) between operating systems of computer and hardware are communicated.

Data acquisition Control Panel can be started from desktop of computer or typing DACP (Data Acquisition Control Panel) on Matlab editor.

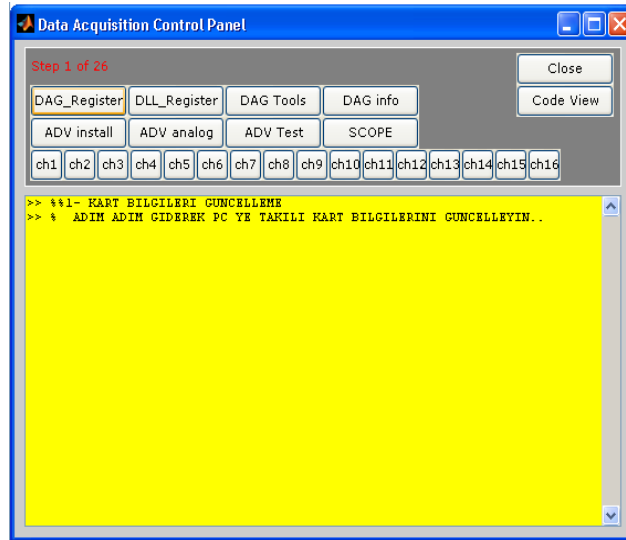


Figure 4.12 Data acquisition control panel main screen

User can pressed enter to initialize the Data Acquisition Control Panel. Control Panel will be appeared on the screen with some information windows and command buttons. There is many possibilities to increase the number of buttons to increase the operations numbers. Here in this thesis only required steps are created with callback functions.

Dag Register and DLL(Dynamic Link Library) Register buttons check the interface programs (Adaptors) in specified directories and their contents. Users can reach hardware components to have specified information such as installed adaptor list, data acquisition toolbox version, Matlab versions, number of channel configured, analog input properties. Advantech data acquisition hardware can also be managed from Data Acquisition Control Panel. ADSDA Devices window will be opened when ADV Test button is depressed.

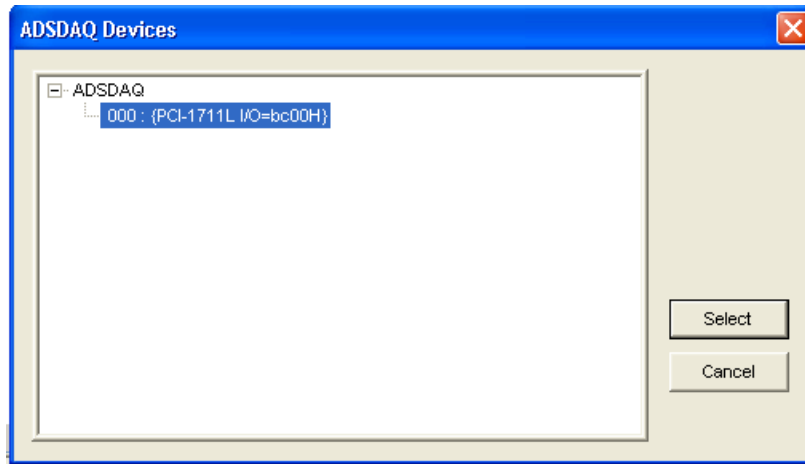


Figure 4.13 Device testing of installed adaptors

In this screen the number of installed data acquisition boards will be listed. One of that listed boards can be selected to test.

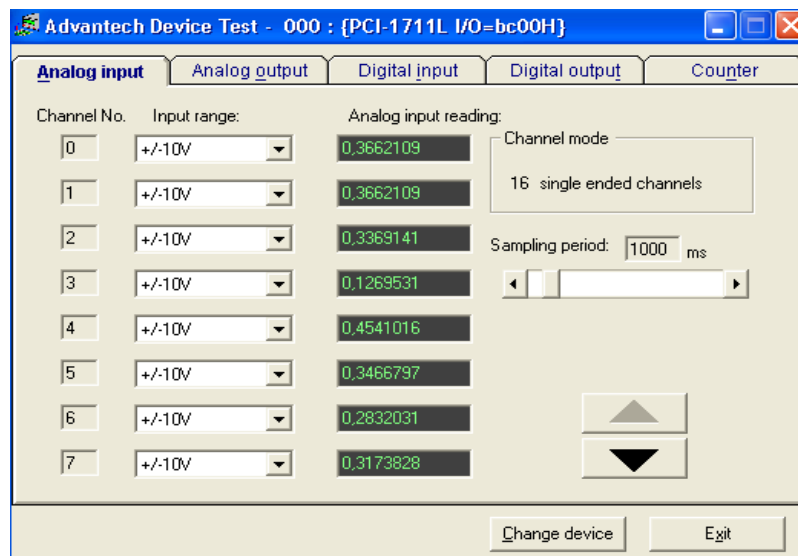


Figure 4.14 Device test of Advantech DAQ card

Analog inputs can be created by pushing the ADV analog button. Listed channel buttons from Ch1 to Ch16 creates channel on hardware. After operation, all analog inputs will be cleared automatically. Users can start measurement by softscope by clicking the Scope button.

Operation and control panel can be ended by clicking close button.

CHAPTER 5

CONCLUSIONS

Quality of measurements and easy-install features of Data Acquisition Toolbox is gathered by Data Acquisition Control Manager. Written program Data Acquisition Control Panel skips all unnecessary steps of typing commands on Matlab Editor in order to start up the acquisition.

Users of data acquisition systems must have advanced level of knowledge about data acquisition toolbox to acquire data. The buttons on Data Acquisition Control Panel, operate registration, getting some information from the data acquisition hardware and Matlab, and automatically create all required analog inputs and clear all after program ends.

The program of Data Acquisition Control Panel is open source codes on Matlab, so the program can be changed as users request. As a result the Data Acquisition Toolbox of Matlab has a opportunities to improve the quality of measurements and testing the devices without having knowledge about Matlab Data Acquisition Toolbox and Hardware.

Install wizard of Data Acquisition Control Panel automatically install required DLL and interfaces in their directories. All required software for Matlab and Advantech are also installed and registered automatically.

Quality of data acquisition process and its simplicity are quite important to save time in laboratories. In electronics and automation industries, there are several platforms, software and hardware. Flexible software can be modified as users request for high efficient measurement.

REFERENCES

- [1] **Howard A.** ‘‘Data Acquisition Techniques Using PC’s’’ Elsevier, pp 6-13, 2002

- [2] **T. Özkul** ‘‘ Data Acquisition and Process Control Using Personal Computers,’’ Marcel Dekker, pp 3-6, 1996

- [3] **K.James,**’’ PC Interfacing and Data acquisition’’, Newnes, pp 34-38, 2000

- [4] Data Acquisition Toolbox Adaptor Kit User’s Guide, The Mathworks Inc., 2002

- [5] Matlab 7.0 user’s guide. The Mathwoks Inc.2005

- [6] Advantech PCI-1711 Multifunctional.Board User’s Guide.Advantech Ltd.Co.2003 pp 17-44

APPENDICES

Appendix A

I/O Connectors and pin assignment

AI0	68	34	AI1
AI2	67	33	AI3
AI4	66	32	AI5
AI6	65	31	AI7
AI8	64	30	AI9
AI10	63	29	AI11
AI12	62	28	AI13
AI14	61	27	AI15
AIGND	60	26	AIGND
DA0_REF	59	25	DA1_REF
DA0_OUT	58	24	DA1_OUT
AOGND	57	23	AOGND
DI0	56	22	DI1
DI2	55	21	DI3
DI4	54	20	DI5
DI6	53	19	DI7
DI8	52	18	DI9
DI10	51	17	DI11
DI12	50	16	DI13
DI14	49	15	DI15
DGND	48	14	DGND
DO0	47	13	DO1
DO2	46	12	DO3
DO4	45	11	DO5
DO6	44	10	DO7
DO8	43	9	DO9
DO10	42	8	DO11
DO12	41	7	DO13
DO14	40	6	DO15
DGND	39	5	DGND
CNT0_CLK	38	4	PACER_OUT
CNT0_OUT	37	3	TRG_GATE
CNT0_GATE	36	2	EXT_TRG
+12V	35	1	+5V

Appendix B

I/O Connector signal descriptions

Signal Name	Reference	Direction	Description
AI<0..15>	AIGND	Input	Analog Input Channels 0 through 15.
AIGND	-	-	Analog Input Ground.
DA0_REF DA1_REF	AOGND	Input	Analog Output Channel 0/1 External Reference.
DA0_OUT DA1_OUT	AOGND	Output	Analog Output Channels 0/1.
AOGND	-	-	Analog Output Ground. The analog output voltages are referenced to these nodes.
DI<0..15>	DGND	Input	Digital Input channels.
DO<0..15>	DGND	Output	Digital Output channels.
DGND	-	-	Digital Ground. This pin supplies the reference for the digital channels at the I/O connector as well as the +5VDC supply.
CNT0_CLK	DGND	Input	Counter 0 Clock Input. The clock input of counter 0 can be either external (up to 10 MHz) or internal (1 MHz), as set by software.
CNT0_OUT	DGND	Output	Counter 0 Output.
CNT0_GATE	DGND	Input	Counter 0 Gate Control.
PACER_OUT	DGND	Output	Pacer Clock Output. This pin pulses once for each pacer clock when turned on. If A/D conversion is in the pacer trigger mode, users can use this signal as a synchronous signal for other applications. A low-to-high edge triggers A/D conversion to start.
TRG_GATE	DGND	Input	A/D External Trigger Gate. When TRG_GATE is connected to +5 V, it will enable the external trigger signal to input. When TRG_GATE is connected to DGND, it will disable the external trigger signal to input.
EXT_TRG	DGND	Input	A/D External Trigger. This pin is external trigger signal input for the A/D conversion. A low-to-high edge triggers A/D conversion to start.
+12V	DGND	Output	+12 VDC Source.
+5V	DGND	Output	+5 VDC Source.

Appendix C

Block Diagram of Advantech PCI 1711

