IEEE DATA ACQUISITION

REFERENCES:

National NI-488.2 User and Reference Manuals. (These manuals come with the National Instruments IEEE-488 interface board for the lab PC that we will use in this experiment.)

Various instrument manuals—for example, the Keithley *Model 196 DMM Instruction Manual*.

Various Basic and Visual Basic Manuals

NOTE: Manuals may be checked out from your instructor. They are not to be taken out of the Science Hall, and you are PERSONALLY RESPONSIBLE for letting your instructor know where the manuals are, and for returning them after the experiment.

I. Introduction

Very often, the actual taking of experimental data can be a tedious and time-consuming part of an otherwise interesting experiment. Thus one important function of the computer for scientists and engineers is to automate data acquisition, and it is the function of this experiment to introduce you to that process. There are a number of methods of interfacing computers to experiments; the one that you will study here uses the IEEE-488 standard, somtimes also referred to as gpib (for general purpose interface bus) or hpip (for Hewlett-Packard interface bus). It has the following advantages and disadvantages:

Advantages of the IEEE-488 interface:

1. It is easy to use. The electronics are already built into the instruments (such as voltmeters, frequency counters, oscilloscopes, etc.), so that one does not have to do much electronics before starting an experiment.
2. It uses a reasonably standard set of commands.
3. In most implementations, “high level” programming languages such as C, BASIC, and FORTRAN can to be used to program the experiment, so that little or no assembly level programming is needed.
4. Instruments that support the IEEE-488 interface are widely available.

Disadvantages of the IEEE-488 interface:

1. In practice, it is comparatively slow. Although in principle the IEEE-488 standard supports data transfer rates of up to 1 MBytes/sec, in practice rates are generally much slower, since few instruments are designed to support such fast

transfer rates, and those few are generally very expensive. The instruments in the physics department are not particularly fast; according to the manuals, they can transfer instrument readings at rates up to several hundred per second. We have rarely used them at rates faster than a few per second.

2. The total cable length should not exceed 20 meters, unless special and expensive "repeaters" are used.
3. Instruments equipped with IEEE-488 interfaces can be expensive.
4. For some experiments, standard instruments with an IEEE interface aren't available.

Thus, for experiments that use standard instruments and for which comparatively slow rates of data transfer are adequate, the IEEE-488 interface is a convenient and powerful way of automating an experiment.

II. Physical Setup

The IEEE-488 standard is implemented by a board that plugs into a slot inside the computer. That board serves as the CONTROLLER; that is, it manages the bus, sends commands to and receives data from the other devices on the bus, and so on. The board manufacturer (National Instruments) supplies a series of assembly language subroutines with the board, so that the user can access the IEEE functions by calling those subroutines from a language like C, BASIC or FORTRAN. There is also an Interactive Control utility that you can use to become familiar with these commands.

A standard IEEE-488 cable, to be described below, connects the computer to the instruments. Instruments can be connected in a "chain"—that is, one cable runs from the computer to the first instrument; a second cable runs from the first instrument to the second; and so on. Up to 15 instruments can be operated from one IEEE board. Each instrument is identified by an individual "address," a number from 0 to 30 that allows that instrument to be identified. This address is the *primary* address; in addition, instrument manufacturers have the option of defining *secondary* addresses to assist with specific instrument functions. The primary address is often set using a small DIP switch that is typically set on the back of or inside the computer. In newer instruments, the address can often be set using the front panel controls.

III. IEEE-488 cable and IEEE-488 standard commands

The IEEE-488 cable consists of 24 lines. These lines are used as follows:

1. There are eight ground lines.

2. There are three control (or ”handshake”) lines that are used to coordinate data transfer between the the controller and the instruments.
3. There are eight bi-directional data transfer lines, that are used to send commands back and forth between the instruments and the controller, and also to transfer data–meter readings, and the like.
4. Finally, there are five general interface management lines.

The last two sets of lines are used to send commands back and forth between the instruments and the controller. Each device must be designated as a TALKER, a LISTENER, or as IDLE. At any one time there can be only one talker, but there can be several listeners. And at different times, a given device can serve as both talker and listener. Thus the controller is a talker when it sends instructions to a device, and a listener when it receives data. For example, a typical data transfer might proceed as follows:

1. The controller board on the PC is identified as a TALKER.
2. The instrument (for example, a voltmeter) is identified as a LISTENER. (NOTE: Some inexpensive instruments are TALKERS only; they cannot be set to a LISTEN mode.)
3. The controller sends commands to the instrument; for example, it can tell a Digital Multimeter to set itself to an ohmmeter mode.
4. Next the controller tells itself to be the LISTENER and the instrument the TALKER.
5. The instrument sends information (for example, a voltage reading) to the controller. (NOTE: Information is sent in the form of a ”string”– that is, a series of ASCII characters. If the string represents a number—say a voltage— the program must convert this string to a real number.)

Let us see how this process works in more detail. We begin by describing the five interface management lines. Like most computer electronics, these lines transfer *digital* information by switching between a “low” state (typically 0 volts) and a “high” state (typically +5 volts). Thus one speaks of a line being high or low, asserted or not asserted, or some similar terminology. The names and functions of the five interface management lines are listed and sometimes *briefly* described below; see the manuals for more complete descriptions.

NOTE: It is often more effective to understand these commands by seeing how they are used than by puzzling too long over the definitions. The long lists of commands can be a little intimidating when you see them for the first time, but you will soon learn which ones are important for your application, and how to use them.

1. ATN (ATtentioN)

   When ATN is asserted, information on the bus is interpreted as commands; otherwise, as ”messages”–for example, meter readings.

2. IFC (InterFace Clear)

3. REN (Remote ENable)

   When REN is asserted, the instrument can be programmed by the controller–that is, from the computer in most cases.

4. SRQ (Service ReQuest)

5. EOI (End Or Identify)

In addition, there are a number of standard IEEE-488 commands that are sent on the data lines; for these commands to be sent, ATN must be asserted. The names and functions of these commands are listed (and sometimes *briefly* described below. See the manuals for more information.

A. Universal Commands (go to all devices on the bus)

   1. DCL (Device Clear)
      Sets all devices on the bus to their default power-up states; those states are defined by the device manufacturer.
   2. LLO (Local LockOut)
      Deactivates the front panel controls of all devices on the bus.
   3. PPU (Parallel Poll Unconfigure)
   4. SPE (Serial Poll Enable)
   5. SPD (Serial Poll Disable)

B. Addressed or Primary Commands (go to specific device addresses)

   1. GET (Group Execute Trigger)
      Triggers a group of devices simultaneously.
   2. SDC (Selected Device Clear)
      Similar to DCL, but acts only on a specifically addressed device.
   3. GTL (Go To Local)
      Returns control to the instrument’s front panel controls.
   4. PPC (Parallel Poll Configure)
   5. TCT (Take ConTrol)
      Used to specify another device as a controller; (used only if there exists a second device that can function of a controller).

C. Secondary Commands

1. PPE (Parallel Poll Enable)
2. PPD (Parallel Poll Disable)
3. MSA (My Secondary Address)
4. MLA (My Listen Address)
   defines the controller as listener
5. MTA (My Talk Address)
   defines the controller as talker
6. UNL (UnListen)
7. UNT (UnTalk)
   These last two commands place all devices in the IDLE state; for example, UNL disables all current listeners.

IV. DEVICE-SPECIFIC COMMANDS

In addition to the standard IEEE-488 commands listed above, each device manufacturer will usually define a set of commands that are specific to each instrument. For example, the Keithley manual for the Model 196 Digitial Multimeter that you will use in this experiment has a set of commands that relate to the operation of the meter. These commands are usually sent as "messages;" that is, they are sent with the ATN line unasserted. You will find list of these commands at the end of this section of the write-up.

Similarly, the NI-488.2 controller card and software used on the lab PC has numerous device-specific commands that implement the IEEE standard. They are discussed below; but see the manuals for a full discussion.

V. The Experiment

NOTE: You should write a detailed description of the actual procedure you follow in your laboratory notebook. This description should be written while you are in the lab doing the experiment . Do NOT rely on your memory or on rough notes taken on scrap paper!! Do NOT simply copy the suggested procedure in this writeup—write down what you actually do!!!

The experiment consists of three parts:

1. Play with the computer and the IEEE-488 apparatus and learn how they work, using the Interactive Control utility. Here, the only apparatus you will need is the computer, an IEEE-488 device (for example, the Model 196 DMM and a power supply.

2. Write a short Visual Basic program that takes control of the Model 196 DMM, sets the mode, and makes a measurement.
3. Finally, do the Newton's Law of Cooling experiment.

1. Interactive Control Untility

Begin by reading through Chapter 6 of the User Manual.

Then, make sure the IEEE cable is connected to the Keithly Model 196 DMM (digital multimeter) and to the computer, and turn on the power to the DMM.

Then, find the National Instrument "Measurement and Automation" icon, and double-click to open MAX, the Measurement and Automation Explorer. Take a few minutes to look over the information given on-screen, which should include the IEEE addresses of both the DMM and the IEEE card in the computer. Right-Click on the "instrument," choose "communicate with the instrument," and make sure everything is working.

Then, right-click on the GPIB0, and choose Interactive Control. A command shell will open. Go back to the MAX program and in the same way, open the NI Spy untility—this program will monitor the IEEE commands and let you know if there are any errors.

You can use the Interactive Control Utility in three modes.

Device Mode

This mode is the easiest to use, and the one you will probably want to use for you Visual Basic program later.

Proceed as follows:

1. At the Prompt, type ibdev

   This command opens and initializes a "device descriptor" that opens communication between the IEEE Controller and the DMM. You will be prompted for the board (or Controller) index (0), the primary and secondary addresses of the DMM (7 and 0), and other information that you will find in Chapter 6 of the User Manual. If you like, you can enter all of this information following the ibdev command, instead of waiting to be prompted.

2. Type ibclr, and watch the Keithly 196 as you do so. Note and explain the changes you see in your lab book.

   This command sends the IEEE SDC command (see above) to the DMM. It also sends a LLO (local lockout) command—notice that the function buttons on the DMM no longer respond.

3. Type ibloc, which returns command to the instrument by sending a GTL (go to local) command. Watch what happens to DMM panel lights, and note that DMM function buttons respond again.

4. Type ibclr again, and then use ibrd and ibwrt to send commands to the DMM, and read data from it. A description of these commands is given in the Reference Manual. See if you can strip away the alphabetic descriptors in the DMM output, change the meter function, and so on, using ibwrt.

   Again, watch the DMM panel lights, and be sure you understand why they are changing.

5. When you are finished, type ibonl 0 to take the DMM off-line.

Interface or Board-Level Mode

This mode is more complex, but gives considerably more detailed control over the IEEE Controller and the DMM. Proceed as follows:

Set up the board by typing the following commands (which you should look up in the Reference manual):

ibfind gpib0	Initializes IEEE interface; see Reference Manual.
ibpad 0	This command sets the index for the IEEE Controller to zero. It may not be necessary, but can't hurt anything.
ibrsc 1	requests system control for the IEEE board. See Manuals.
ibsic	Sends an IFC (interface clear) command.
ibsre 1	Sends REN (remote enable) command.

Then, use the following commands to communicate with the DMM:

ibcmd string	Sends an IEEE command. The string should be in quotes. For example, "\x20+\x47" sets the IEEE board (at address 0) to LISTEN and the DMM (at adddress 7) to TALK mode. The \x indicates a hexidecimal number. See Appendix A in the Reference Manual for other examples. See, for example, if you can set the Controller to listen, and the DMM to talk.
ibrd ibwrt	These commands should work as before, if the correct commands have been set using ibcmd.

As before, use ibonl 0 to take the interface off-line when you are finished. Feel free, of course, to try some of the other commands discussed in the manuals. Try iblines, for example.

Multi-Device Mode

This mode is particularly convenient if you are using multiple IEEE devices, but can be used even if you are using only one. Initialize by typing

set 488.2(0)

at the Interactive Control prompt, and see the manuals for commands to try.

2. Short Basic Program

Once you understand the use of these subroutines in the Interactive Control utility, the next step is to write a short BASIC program that uses them. We will use Microsoft Visual Basic. It has the disadvantage of being restricted to Windows (as oppposed to Linux or UNIX), but has the advantage of making it easy to write simple programs.

The first step, if you have not done much programming, is to learn something about BASIC programming in general, and Visual Basic in particular. Your instructor should have a few books or manuals you can use in getting started. Proceed as follows:

1. Start Visual Basic
2. Drag a "Command Button" to the Visual Basic "form" that will appear on screen.
3. Double-click the Command Button to open a code window—here, you can enter a short program that will execute when you click the command button while the program is running.

Don't be concerned if you haven't done a great deal of programming. In this experiment, you will only need to write a short program. There are both sample programs and faculty help available!

Once you can write a short program that does some simple arithmetic and reports the results, you can proceed to write a program that will control the IEEE interface and the DMM. There are sample programs that will help you get started.

Be sure to include comments in your program, explaining what each section does.

Your program should perform the following functions:

1. Set the digital multimeter (DMM) to REN and LLO.

2. Set the DMM to a non-default state.
3. Take an instrument reading.
4. Remember that the instrument reading is initially in the form of a string. Convert that string to a real number (using the VAL command, and perform some arithmetic operation on it (for example, add 5 to it).

Be sure that you save your work frequently.

When you have a working program, demonstrate it for your instructor. Be sure to print a copy to include in your lab notebook. At this point you are ready to do an actual experiment, and to see automated data acquisition in practice. The experiment will involve Newton's Law of Cooling, and is described in more detail below. It will take some hours to take the data, but you won't have to be there! Once you set up the apparatus and start the program, the computer will take the data.

Table 3-8. Device-Dependent Command Summary

<table>
<tr><th>Mode</th><th>Command</th><th colspan="8">Description</th><th>Paragraph</th></tr>
<tr><td>Execute</td><td>X</td><td colspan="8">Execute other device-dependent commands.</td><td>3.9.1</td></tr>
<tr><td rowspan="8">Function</td><td>F0</td><td colspan="8">DC volts</td><td rowspan="8">3.9.2</td></tr>
<tr><td>F1</td><td colspan="8">AC volts</td></tr>
<tr><td>F2</td><td colspan="8">Ohms</td></tr>
<tr><td>F3</td><td colspan="8">DC current</td></tr>
<tr><td>F4</td><td colspan="8">AC current</td></tr>
<tr><td>F5</td><td colspan="8">ACV dB</td></tr>
<tr><td>F6</td><td colspan="8">ACA dB</td></tr>
<tr><td>F7</td><td colspan="8">Offset compensated ohms</td></tr>
<tr><td rowspan="9">Range</td><td></td><td>DCV</td><td>ACV</td><td>DCA</td><td>ACA</td><td>Ohms</td><td>ACV dB</td><td>ACA dB</td><td>Offset Compensated Ohms</td><td rowspan="9">3.9.3</td></tr>
<tr><td>R0</td><td>Auto</td><td>Auto</td><td>Auto</td><td>Auto</td><td>Auto</td><td>Auto</td><td>Auto</td><td>Auto</td></tr>
<tr><td>R1</td><td>300mV</td><td>300mV</td><td>300 µA</td><td>300 µA</td><td>300 Ω</td><td>Auto</td><td>Auto</td><td>300 Ω</td></tr>
<tr><td>R2</td><td>3 V</td><td>3 V</td><td>3mA</td><td>3mA</td><td>3 kΩ</td><td>Auto</td><td>Auto</td><td>3k Ω</td></tr>
<tr><td>R3</td><td>30 V</td><td>30mV</td><td>30mA</td><td>30mA</td><td>30 kΩ</td><td>Auto</td><td>Auto</td><td>30 kΩ</td></tr>
<tr><td>R4</td><td>300 V</td><td>300 V</td><td>300mA</td><td>300mA</td><td>300 kΩ</td><td>Auto</td><td>Auto</td><td>30 kΩ</td></tr>
<tr><td>R5</td><td>300 V</td><td>300 V</td><td>3 A</td><td>3 A</td><td>3MΩ</td><td>Auto</td><td>Auto</td><td>30 kΩ</td></tr>
<tr><td>R6</td><td>300 V</td><td>300 V</td><td>3 A</td><td>3 A</td><td>30MΩ</td><td>Auto</td><td>Auto</td><td>30 kΩ</td></tr>
<tr><td>R7</td><td>300 V</td><td>300 V</td><td>3 A</td><td>3 A</td><td>300MΩ</td><td>Auto</td><td>Auto</td><td>30 kΩ</td></tr>
<tr><td rowspan="3">Zero</td><td>Z0</td><td colspan="8">Zero disabled</td><td rowspan="3">3.9.4</td></tr>
<tr><td>Z1</td><td colspan="8">Zero enabled</td></tr>
<tr><td>Z2</td><td colspan="8">Zero enabled using a zero value (V)</td></tr>
<tr><td rowspan="2">Filter</td><td>P0</td><td colspan="8">Filter disabled</td><td rowspan="2">3.9.5</td></tr>
<tr><td>Pn</td><td colspan="8">Filter on with a value of n (n=1 to 99)</td></tr>
<tr><td rowspan="7">Rate</td><td></td><td colspan="8">Resolution</td><td rowspan="7">3.9.6</td></tr>
<tr><td></td><td>DCV</td><td>ACV</td><td>DCA</td><td>ACA</td><td>OHMS</td><td>ACV dB</td><td>ACA dB</td><td>Offset Compensated Ohms</td></tr>
<tr><td>S0</td><td>3½d</td><td>3½d</td><td>3½d</td><td>3½d</td><td>3½d(R1-R4)
5½d(R5-R7)</td><td>5½d</td><td>5½d</td><td>5½d</td></tr>
<tr><td>S1</td><td>4½d</td><td>4½d</td><td>4½d</td><td>4½d</td><td>4½d(R1-R4)
5½d(R5-R7</td><td>5½d</td><td>5½d</td><td>5½d</td></tr>
<tr><td>S2</td><td>5½d</td><td>5½d</td><td>5½d</td><td>5½d</td><td>5½d</td><td>5½d</td><td>5½d</td><td>5½d</td></tr>
<tr><td>S3</td><td>6½d</td><td>5½d</td><td>5½d</td><td>5½d</td><td>6½d(R1-R6)
5½d(R7)</td><td>5½d</td><td>5½d</td><td>6½d</td></tr>
<tr><td></td><td colspan="8">Integration period: 3½d=318µsec, 4½d=2.59msec, 5½d and 6½d=Line cycle</td></tr>
<tr><td rowspan="8">Trigger Mode</td><td>T0</td><td colspan="8">Continuous on Talk</td><td rowspan="8">3.9.7</td></tr>
<tr><td>T1</td><td colspan="8">One-shot on Talk</td></tr>
<tr><td>T2</td><td colspan="8">Continuous on GET</td></tr>
<tr><td>T3</td><td colspan="8">One-shot on GET</td></tr>
<tr><td>T4</td><td colspan="8">Continuous on X</td></tr>
<tr><td>T5</td><td colspan="8">One-shot on X</td></tr>
<tr><td>T6</td><td colspan="8">Continuous on External Trigger</td></tr>
<tr><td>T7</td><td colspan="8">One-shot on External Trigger</td></tr>
</table>

Table 3-8. Device-Dependent Command Summary (Cont.)

Mode	Command	Description	Paragraph
Reading Mode	B0 B1	Readings from A/D converter Readings from data store	3.9.8
Data Store Size	I0 In	Continuous data store mode Data store of n (n=1 to 500)	3.9.9
Data Store Interval	Q0 Qn	One-shot into buffer n=interval in milliseconds (1msec to 999999msec)	3.9.9
Value	V±nn.nnnn or V±n.nnnnnnE+n	Calibration value, zero value	3.9.10
Calibration	C0 C1	Calibrate first point using value (V) Calibrate second point using value (V)	3.9.10
Default Conditions	L0 L1	Restore factory default conditions and save (L1) Save present machine states as default conditions	3.9.11
Data Format	G0 G1 G2 G3 G4 G5	Reading with prefixes. Reading without prefixes. Buffer readings with prefixes and buffer locations. Buffer readings without prefixes and with buffer locations. Buffer readings with prefixes and without buffer locations. Buffer readings without prefixes and without buffer locations.	3.9.12
SRQ	M0 M1 M2 M4 M8 M16 M32	Disable Reading overflow Data store full Data store half full Reading done Ready Error	3.9.13
EOI and Bus Hold-off	K0 K1 K2 K3	Enable EOI and bus hold-off on X Disable EOI, enable bus hold-off on X Enable EOI, disable bus hold-off on X Disable both EOI and bus hold-off on X	3.9.14
Terminator	Y0 Y1 Y2 Y3	CR LF LF CR CR LF	3.9.15
Status	U0 U1 U2 U3 U4 U5 U6 U7 U8	Send machine status word Send error conditions Send translator word Send buffer size Send average reading in buffer Send lowest reading in buffer Send highest reading in buffer Send current value Send input switch status (front/rear)	3.9.16
Multiplex	A0 A1	Auto/Cal multiplex disabled Auto/Cal multiplex enabled	3.9.17