

# Concurrent Tutorials and Applications

E. J. MASTASCUSA AND MAURICE F. ABURDENE

The use of tutorial programs running side by side with applications programs is an effective way to teach many facets of introductory instrumentation and measurements. Side-by-side tutorials were implemented for teaching the use of instrumentation in stand-alone and IEEE-488-controlled situations and the display and analysis of the data in various applications programs. The hardware and software needed are described. By this approach, tutorials can be written by faculty, researchers, and students for teaching applications for which no commercial tutorials are available. Side-by-side tutorials have been implemented in Windows on International Business Machines (IBM) or IBM clone platforms by using the Borland C/C++ compiler. Windows permits the authors of the lesson to include code and code fragments that can be copied and pasted from the lesson program into the compiler program while both run simultaneously under Windows. The combination of Windows, an authoring system, and a Windows-compatible compiler permits the development of lessons that incorporate important pedagogical principles. Students can be given varied repetitions while actively interacting with the lessons. Feedback can be immediate and effective.

Personal computers present interesting possibilities for teaching modern laboratory instrumentation. In this paper we describe how we are using a combination of personal computers, computer-controlled instruments, a Windows environment, several different languages, and an authoring system to enhance how students learn instrumentation. This combination produces a unique learning environment that allows students to proceed at an individual pace in an interactive environment and that frees instructors to focus on student problems in a way that enhances teaching effectiveness.

The student experiment stations that we use include a number of interesting features. Each station consists of a computer, an IEEE-488 card and software, and a set of instruments. Specifically, we have eight copies of the following in every laboratory space.

- International Business Machines (IBM) 386 personal computers (PCs) with 180-megabyte (MB) disk, 8-MB RAM,
- Software for the PC, including Borland C/C++ software,
- National Instruments IEEE-488 board (NI-488.2) and software (BASIC and C drivers),
- Fluke Hydra or HP3421A data acquisition units, and
- HP54601A digital storage oscilloscopes.

Figure 1 shows the system.

These are the standard instruments in three laboratories. Individual laboratories have specialized instruments. For example, the control systems laboratory has IEEE-488 digital-to-analog converters and power amplifiers to run pump motors and fans. Electronics and circuits laboratories have function generators and power supplies on the IEEE-488 bus. These instruments customize

each laboratory for a particular course or purpose, but the general learning environment is the same in all laboratories.

Each laboratory has the same general software. The most critical element of the software is the Windows environment that permits the use of authoring tools to produce multimedia lessons. The lessons run on the machine located directly beside the instrument. As the student progresses, it is possible to use multimedia lessons that interact with various kinds of software to control an instrument over the IEEE-488 bus. The software, the instruments, and the multimedia lessons all run at once to guide the student through the use of the instrument, the basics of the command set for an instrument, and the issuing of commands and the taking of measurements within a program.

## INTERACTIVE LESSONS

An authoring system can be used to develop lessons about instruments. The lessons have several characteristics that enhance the effectiveness of student learning. Lessons can be self-paced. Fast students can go through the lessons quickly. Slower students can proceed at an appropriate pace to eliminate problems that can occur when trying to walk students through an operation in a lockstep fashion.

Lessons can be at a depth appropriate to the student. For example, in discussing how to use a voltmeter, a pictorial representation of a voltmeter, like that in Figure 2, is shown on the screen.

In the computer lessons, this picture appears with a note that the student can click on any part of the pictorial representation for more information. The information is presented in a field that explains, for example, the function of the red input terminal of the voltmeter. Students are free to look at whatever information they need.

Students can look at information on the screen in a nonthreatening environment. Students may be intimidated from asking an instructor about something they fear the instructor believes is very basic. A well-written multimedia lesson permits students to examine basic material any time they feel the need to do so.

In addition to flexible access to material, students can determine the degree of detail they require on the topic. Lessons can be constructed to give students the option of pursuing something in more depth by including option buttons within a lesson. For example, the voltmeter lesson will display a short discussion of how the voltmeter displays its readings when the student clicks on the light-emitting diode display portion of the sketch. Within the box displayed there is a "More" button, which, when clicked, displays a short discussion of how many digits that meter might have.

## RUNNING LESSONS WITH CONCURRENT APPLICATIONS PROGRAMS

Multimedia lessons can also be written to teach students the basics of IEEE-488 commands. The National Instruments IEEE-488

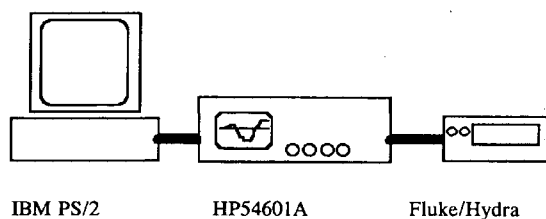


FIGURE 1 Experiment station.

board installed at Bucknell University has a Windows program, WIBIC, that permits the user to issue commands one at a time over the IEEE-488 bus. The program comes in a Windows version, so it is possible to run that program concurrently with a multimedia lesson in another window.

Since we know the instruments attached to the computer and also the approximate knowledge level of the student, we can write lessons that will walk the student through particular commands for the instrument. For example, the data acquisition unit in our set of instruments has the following set of commands:

```
“*RST\r\n”
```

```
“FUNC 0,DCV,AUTO\r\n”
```

```
“*TRG\r\n”
```

```
“LAST? 0\r\n”
```

This particular set of commands resets the data acquisition unit (DAU) (\*RST) and then indicates that the function is a direct current voltage measurement (DCV) on channel 0 and that the DAU should be automatically scaled. Then, the measurement is triggered (\*TRG), and the last value measured is readied for transmission to the computer (LAST). After that, the lesson guides the students through the commands to read the measured value.

There is an unexpected advantage to this approach. The programming system displays the characters received from the instrument, including the terminating characters in the string. Seeing the information transmitted back to the computer drives home to the students that it is a character string and not a number that is sent by the instrument. Later, when programming to receive measurement results, students find it much easier to accept the idea of a character string.

#### COMPUTER CONTROL IN C/C++: RUNNING LESSONS SIDE BY SIDE

Once acquainted with an instrument in a “single-command” mode using the National Instruments program, the student is ready to use a programming language to control the instrument. The system can be programmed in BASIC and in C/C++. BASIC is

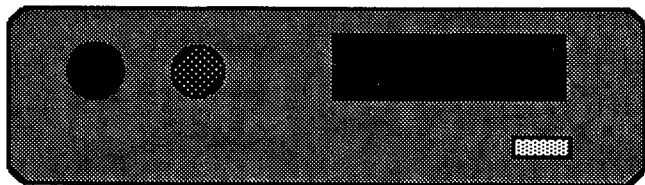


FIGURE 2 Voltmeter.

an interpreter and runs in its own DOS environment. The C/C++ compiler can run in a Windows environment and has the usual abilities to cut and paste from other Windows applications running simultaneously. In this environment, one can write multimedia lessons in which the fundamentals of instrument control are taught in an interactive manner.

Our instrument tutorials include the use of C/C++ to control instruments. In a Windows environment, the C/C++ compiler can run at the same time as the multimedia lesson. Students can size the lesson window and the compiler window so that the multimedia lesson and the compiler are running side by side on the monitor. The multimedia lesson can include program examples and program fragments for instrument control. Program examples and program fragments can be cut from the multimedia lesson by the student and pasted into the compiler editor window, compiled, linked, and run.

The ability to provide students with sample programs and fragments at a controlled rate within the lesson is one advantage of this method. Also, it is possible to have students check and learn what happens when they make mistakes. The lessons can even direct the students to do something that is not correct so that they learn from it, with the multimedia lessons pointing out why some particular approach is incorrect. Again, in a nonthreatening environment monitored by a multimedia lesson, there are no grade implications when a mistake is made. Since the tutorial anticipates mistakes, students are more apt to experiment or to be embarrassed less because they realize that mistakes are made often if the tutorial anticipates them.

The multimedia lessons can include a wide variety of functions for use with the instruments. The cut and paste capabilities permit students to develop their own programs and applications rapidly and with some assurance of success.

#### CONCLUSIONS

The environment discussed in this paper permits students to learn how to use instruments manually and under computer control. The method has proved to be efficient because it permits an instructor to focus on students with problems and to let students who are proceeding well to continue at their own pace. It is also an effective method when judged in terms of student learning.

One feature of this approach is the structured repetition that is built into the system. When students use an instrument manually, they need to attend to several things; for example, they must reset the instrument, set the type of measurement, take the measurement, and interpret the results. Later, in the single-command mode with IEEE-488, they repeat essentially the same sequence of operations. Finally, they repeat that sequence of operations again in a program. The repetition provides a reinforcement to their learning that aids retention and improves understanding. Interest is piqued, and many students return to the laboratory to reuse and rerun the lessons.

Also, this method is effective because it involves the students interactively in their learning. Students need to work with the instruments and to keep working through the lessons. Our experience has shown that students find this approach more interesting.

Finally, although this method has been presented for teaching how to use a data acquisition unit, it is applicable to other instruments. We are developing lessons for oscilloscopes, power supplies, and function generators. We have developed lessons to teach the C/C++ programming language that give the students the

background they need to program the instruments. The side-by-side approach has proven effective for teaching instrumentation and programming.

#### **ACKNOWLEDGMENTS**

Borland C++ is a trademark of Borland International, Inc.

IBM and PS/2 are registered trademarks of International Business Machines Corporation.

Hydra is a trademark of John Fluke Mfg. Co., Inc.

NI-488.2 is a trademark of National Instruments Corporation.

Development of the principal laboratory described in this paper was supported primarily by NSF. That support is gratefully acknowledged.