Teaching IT Concepts Is Enhanced by Including Hardware in Experiential Learning

Dr. Richard G. Helps, Brigham Young University

Richard Helps has degrees in EE and a PhD in Instructional Technology. He is primarily focused on Cyber-Physical systems in IT with related interests in HCI. He is also interested in instructional design. He is a member of ACM-SIGITE, IEEE-CS and ASEE. He is an ABET accreditation commissioner.
Teaching IT Concepts is Enhanced by Including Hardware in Experiential Learning

Abstract

Information Technology (IT), like other computing disciplines, is a largely software-oriented discipline, however teaching aspects of computing are significantly enhanced with hardware support. For example some programs use microcontrollers and low-level languages like C or assembly language to teach basic computer architecture concepts. Others use the flexibility of programmable platforms to teach basic discrete mathematics concepts. Yet other programs in IT recognize that the computing world is moving to different platforms, such as mobile platforms and the “Internet of Things” and choose to incorporate these systems in the design of their student educational experience.

Using hardware in computing instruction enhances the learning experience for many students, especially those with learning profiles indicating a preference for active or experiential learning. Both Keirsey-profile (MBTI) ‘Artisans’ and ‘Rationals’ can benefit from this approach.

A significant problem with using hardware-oriented tools to enhance learning is that students (and many faculty) in IT typically don’t have a deep background in hardware design and debugging. Therefore hardware platforms must be selected which allow for development of learning experiences with ready-made and easy-to-use platforms. Platforms such as mobile phones epitomize this, as the hardware platform is complete and sealed, but includes a variety of sensors and interfaces, which offer versatile and flexible options for instructional development. Some single-board computers, such as the Raspberry Pi, offer extensive development support to minimize the need for hardware expertise, similarly a few microcontroller platforms, such as the Cerebot series and the Arduino series, offer extensive pre-built sensors and other interfaces, complete with software drivers and development systems, making them available to a software-oriented audience.

This paper describes the educational concepts that lead to improved learning through incorporating hardware platforms. These principles are illustrated with several example applications that validate the principles.

Introduction

IT, like most computing disciplines, has an emphasis on software and systems integration rather than hardware design and application. None of the IT fundamentals (Programming, Networking, Human-Computer Interaction, Information Management and Web systems) specifically address hardware issues\(^1\). Similarly the IT ABET accreditation criteria emphasize systems design and do not specifically mention hardware-oriented requirements. For example ABET criterion C for computing programs lists “An ability to design ... a computer-based system”\(^2\) Many IT and other computing professionals implement complex systems using existing hardware modules or computers with little thought as to the functioning or design of that hardware. Electronic design topics do not feature anywhere in almost all IT curricula. Thus, on the whole, IT designers expect to acquire their hardware modules as working units, rather than design and build them.
This is not an absolute statement; there is also a question of granularity. IT designers will happily integrate connection modules, such as plug-in cards for computer busses, or adapters from one transmission protocol to another, but they will not, in general, design or analyze circuits.

Despite this systems and software-oriented approach there are cogent reasons for using hardware in IT instructional design. Many of these reasons are related to the learning experience. The educational community has long promoted multi-sensory learning. The team of Myers and Briggs developed the MBTI tests for understanding preferences, and successors to the MBTI test are still widely used to today. Similarly David Kolb’s experiential learning theory promotes multimodal learning based on a cycle of concrete experience, reflective observation, abstract conceptualization and active experimentation. All of these experiential learning modes can be enhanced by engaging the students with interactions with hardware modules. Small low-cost computing platforms, such as the Arduino microcontroller and related devices, provide a way to physically encapsulate many of the learning concepts related to IT and students can program, control and interact with these systems in a very direct physical fashion, involving not only sight but touch, feel, and other senses too. The opportunity for students to interact with these systems provides a very immersive and interactive learning environment.

There are also IT industry trends that indicate that IT students should become far more aware of topic areas such as sensors, actuators, small (embedded) computers and mechatronic systems. In particular the rapid and recent growth of the cyber-physical systems and the “Internet of Things” (IoT) is opening new opportunities and obligations for next-generation IT professionals. The IoT has received considerable publicity recently with events and announcements appearing regularly in the technical and business press. Here are just a few examples of the growing importance of this field:

- There are international IoT conferences
- Major IT-oriented corporations are pursuing and promoting the IoT, such as Microsoft, Samsung, Google, Cisco and others
- The FTC is urging companies to adopt best practices for security and privacy in IoT design
- The ACM and the IEEE Computer society have multiple articles on the topic and the IEEE has a dedicated “Internet of Things” website

The IoT grew out of the world of embedded systems and cyber-physical systems, however where the original embedded systems were very often isolated and dedicated to a single task, IoT devices have communication with other computers and are thus very much in the domain of IT, requiring considerations of networking, security, human-computer interaction and systems design.

Internet of Things and related Cyber-Physical Systems (CPS) are qualitatively different from an IT design and administration perspective. Conventional desktop/laptop computers, servers in network operating centers and, more recently, smartphones are limited to a very few well-understood operating systems (desktops and laptops are dominated by Windows and Linux, smartphones are dominated by iOS and Android—which is descended from Linux) and there is a significant commonality in CPU hardware, dominated by the Intel and ARM architectures. CPS and IoT due to their diverse application domains have a much more diverse and heterogeneous
hardware and operating system vista. It cannot reasonably be said that any operating system or hardware platform dominates this environment. With the growth in this sector of the market it is very probable that IT professionals will need at least a working knowledge of IoT technologies and devices in the near future.

Despite these opportunities and trends the hardware that is introduced into IT classrooms and labs—as part of the learning experience—should be well-suited to the needs and experience of the students, the context of IT education and the constraints of instructional design. This article addresses two questions in applying these principles.

Questions

The two questions considered here are:

- What are the opportunities and difficulties related to incorporating computer-hardware systems as part of the IT university student learning experience?
- What is the framework that directs and constrains this teaching approach?

This article addresses these questions and then presents several examples of using hardware platforms in IT instructional design to validate the proposed framework.

Opportunities and Difficulties

The major benefit from using teaching environments including hardware is the interactivity between the student and the system. Consider the example of using a small microcomputer module, programmed to demonstrate analog to digital conversion as a concept. The students interact directly with the device; they flex the analog sensor with their fingers and immediately see the corresponding digital value appear at the output. They change the inputs back and forth and try variations within seconds, or analyze a particular setup more thoroughly and interactively. Their action provokes an immediate and instantaneous response. In Human-Computer Interaction (HCI) terminology this is truly a “direct manipulation” interface. If we add into the experience the ability to modify the device to represent different discrete algebraic concepts we create a learning platform that has the adaptability of software combined with the tactile interaction of hardware, and the learning can be enhanced.

However the realities of introducing this type of technology into learning must also be recognized. The key issues are summarized below:

- Interactive (multi-modal) and immersive learning modes are beneficial
- Professional IT practice is a mixture of HW & SW. (Systems approach)
- Simulations are incomplete representations of systems
- IT students don’t have a background in electronic design
  - IT students fear hardware usage
- Practical cost, reliability and maintenance add other constraints

The possible learning benefits have already been described; recognizing that some learners are more susceptible to certain types of teaching further enhances these. Technology instructional design discussions often refer to “hands-on” or experiential learning styles. These and related
learning styles are also described by some well-established psychological evaluation instruments. Using hardware in computing instruction enhances the learning experience for many students, especially those with learning profiles indicating a preference for active or experiential learning. Both Keirsey-profile (MBTI) ‘Artisans’ and ‘Rationals’ can benefit from this approach.

Another benefit of this approach is that the learning experience becomes more authentic. IT concepts can be illustrated by various simulators, however all simulators are necessarily an approximation of the real system. The professional world of IT practice will include both hardware and software. Software simulations frequently omit non-ideal system characteristics. For example consider the analog to digital (A/D) conversion system again. A simple model of analog to digital conversion might show uniform digital steps along a linearly increasing analog line. Real A/D systems do not linearly increase from zero to a maximum. There are frequently inaccuracies at both ends of the scale caused by imperfect but real sensors; there are also errors in measurement between the extremes. The digital conversion process adds other inaccuracies. Simulated systems are extremely valuable to experienced designers who are aware of the differences between the simulation and actual systems, however there is a risk in educational environments that learners will believe that the simulation is an exact representation of the real system and be misled. More authentic learning situations, when available, can help to address this problem. By using a physical device in a carefully designed lab environment these imperfections can be fruitful sources of discussion and learning.

The potential difficulties in using these devices in instructional design are also real. Since IT students do not typically have a circuit design background they are unfamiliar with and even fearful of the issues raised by connecting sensors and actuators to a computing platform. These concerns are addressed in the Common Concerns section later.

**Framework**

These considerations lead to the following framework bring proposed. Teaching with hardware in IT requires that the system meet the following guidelines

**Educational:**
- The system must practically illustrate some important concept of IT
- The system must provide an interactive and immersive experience for students
  - This means that it needs the capability to handle a diverse range of physical inputs and outputs

**Constraints/Requirements:**
- The system must use a combined computing and hardware interface
- The system must not require IT students to complete detailed electronic circuit design and implementation, for which they lack the appropriate background.
- The system must provide a standardized hardware system which is diverse enough to explore many teaching situations but standardized enough to be within the scope of non-electronics students

With these requirements and constraints we will consider alternative hardware platforms.
Selection of platform

There are a multiplicity of possible hardware platforms that could be considered which would fit within the proposed framework. However these can be broken down into a more manageable short list of general types. The key property that every considered type of platform requires is that it can support interfaces to allow students the direct experiential and multi-modal interaction with the hardware to support the direct-manipulation learning interface. This interface must include access to sensors, actuators and displays. Secondly other aspects of cost, flexibility and availability must be considered. Four types of platform will be compared. These are, laptop computers/tablets, smartphones/PDAs, single-board computers and microcontroller-based systems.

Laptop computers/tablets: Using these in teaching has one main advantage, that of ample computing power. The disadvantages are many. They are very expensive to purchase and maintain. If students are required to buy their own systems then compatibility and technological currency is hard to enforce. The biggest disadvantage of these platforms for hardware-integrated learning is that these systems are difficult or expensive to interface to sensors and actuators, which are needed to give the real-world interactions that are desired.

Smartphones/PDAs: these devices have the advantages of being portable and containing a range of sensors in a closed package. Development environments are freely available. They have disadvantages in that expanding the sensors and actuators beyond those that are built-in is difficult and expensive, even more so than for laptop computers. They have the same problems as laptops in terms of purchasing and maintaining them. While most students already own a smartphone there is little standardization, making instructional design very difficult. Smartphones also have a short generational life, provoking a need to redesign the labs for the newer hardware systems every couple of years.

Single-board computers (SBC): These are a promising alternative. There is a wide range of SBCs available, and prices range from very cheap (tens of dollars) to moderate (hundreds of dollars). Most of these SBCs come with standard operating systems (usually Linux), opening the doors to a wealth of software. Their big disadvantage for the intended purpose is the problem of obtaining or designing suitable interfaces to turn them into reliable interactive devices suitable for teaching. As noted above, IT students are singularly inexperienced in these areas and will be at a disadvantage in using or exploring these systems. Some of these difficulties can be mitigated as will be explained.

Microcontroller-based systems: These systems are the most prolific in terms of available alternatives. There is a seemingly endless array of sizes, costs and capabilities available, mostly at low prices. These systems also suffer from the electronic interface problem, however solutions are more easily available for this type of system.

The application of the teaching concepts will be shown through the following examples and the solution to the difficulties will be discussed.
Examples of Hardware to support IT Teaching

At Brigham Young University we have found several courses are significantly enhanced by including hardware in the course instructional materials. Here are a few examples.

An introductory networking class emphasizes the lower layers of the ISO-OSI 7-layer networking model. Part of this class requires teaching basic discrete algebra concepts, such as logic gates and state machines as well as analog to digital (A/D) and digital to analog (D/A) conversion. In this class analog input sensors are connected to a microcontroller and the results of the conversion are visible on a row of LEDs. Similarly switches represent digital input, which can be echoed on the LEDs and to the output as a measured analog voltage. The response of the hardware is instantaneous to the human observer and the systems operate asynchronously, IE the output appears as each switch is flipped or sensor is activated.

An introductory class (IT 101) uses hardware in a couple of instances to help students grasp concept of computing and involve them with the learning experience. One in particular is known as the “Back to the Future” exercise, based on the movie of the same name. Students are provided with a hardware setup with LEDs in the Y-shape of the time traveling “flux capacitor” from the movie as well as hardware and a graphic display that represents the (analog) brake pedal and accelerator from the DeLorean sports car. These are connected to a small microcontroller. By deliberate design the microcontroller is physically separated from the sensors and actuators by a short cable. This clearly and physically illustrates the difference between the computational aspects (microcontroller) and the real-world aspects (accelerator, brake, displays etc.).

Students are challenged to write a program to get the light patterns and sound effects of the “flux capacitor” to perform correctly. The speed of the ‘car’ as it accelerates is displayed on a speedometer on a browser screen. Using a browser screen enables a discussion of networking communications together with the experience of getting the sensors and actuators to respond appropriately. The details of this laboratory experiment and the student learning have been discussed in previously published reports\(^{18,19}\)

Another example is in a computer architecture class. Computer architecture is a staple of IT programs. To help students grasp the basics of discrete algebra the class has been structured around a series of design laboratories. Each laboratory experience has the students design a simple game, using a microcontroller and a set of inputs and outputs. Throughout the semester the games become more complex and the students develop greater skills in programming and in manipulating the environment. In each case they play the game by activating sensors and seeing various (physical) outputs.

The final example is a course in mechatronics targeted at IT students. Mechatronics studies inherently include mechanical systems, electronic circuitry and computing. Mechatronic courses are frequently offered by Mechanical Engineering departments for mechanically oriented students who wish to expand into the area of computing. These courses normally involve
interfacing computers with sensors and transducers. At Brigham Young University we felt it important to offer a class that would allow IT students to participate in the growing area of the IoT. Thus the mechatronics class, which is specifically targeted at IT students, links the world of mechanical devices, electronics and computers with the IT world of systems design, networking, security, HCI and many other IT topics. In this class a variety of experiments involve computing and real-world interactions at varying levels of complexity.

**Common Problems**

All these applications share some common problems in the context of an IT learning experience. The primary problem is the reality that IT students have little exposure to hardware design, electronic circuit design, analysis or debugging. It is not reasonable to expect IT students to develop an electronic interface between a sensor and a computer from scratch.

Our solution to this problem is use a standardized set of hardware and sensor/actuator models. There are several companies developing microcontroller-based systems for this type of application. We use Cerebot development microcontrollers from Digilent Inc. There are a few reasons for choosing this particular system over many competing ones.

- The primary reason is that Digilent provides a very wide range of standardized, pre-built interface modules, which they call Pmods. The benefit of this standardized approach is that connecting a module for a particular sensor or actuator to a computing platform is simply a matter of plugging it in. Since all input plugs are standardized it is hard to make an error. The modules come with supporting documentation and sample programs, so it is easy for the students to test them and then develop their own systems. Thus IT students can explore many real-world effects, and can be involved with real sensors and actuators, without the need to design and debug the interface electronics.

- They also provide a wide range of computing platforms.

- The platforms are all programmable using two or three levels of development environment, ranging from hobbyist to professional. The hobbyist interface is an implementation of the very popular Arduino programming IDE and the hardware will run most Arduino code (‘sketches’) without modification.

We use these platforms in all classes that are using hardware as teaching aids. This uniformity was not a system-wide, *a priori* decision but rather it evolved over time as the standardization and reliability of the system proved their worth in one class after another.

We also considered using the Arduino platform with the many third party interface modules that are available for that system. These interface modules are known as ‘shields’ in Arduino parlance. While there is a very wide selection of these interfaces modules they are less standardized and the supporting software is of very variable quality, however they undoubtedly provide a viable alternative platform.

Another platform we use is the ‘Raspberry Pi’ single-board computer. This is applied when an operating system is required. While there are many single board computers with similar or superior features to the Raspberry Pi, none have caught the popular imagination as this one has. A significant aspect of its popularity is its low price ($35).
The popularity of Raspberry Pi is important for our educational objectives. Because it is so popular there are many interface units readily available. This is analogous to the situation with the Arduino, the interface modules are prolific, but of varying quality and ease of use. While they are not as standardized as the Cerebot Pmod modules there is considerable standardization. No other SBC platform has the support and range of systems and modules that the Raspberry Pi has, hence this is the best match for our constraint of requiring a no-hardware-design system to enable IT students to interact with real-world systems.

The standardized systems we have selected also have reasonable costs and good reliability. The initial cost to provide systems for class sections of 60-plus students was less than two thousand dollars and the maintenance costs for parts of the order of hundreds of dollars per year—usually for the acquisition of new sensors and actuators to extend the systems further.

We arrived at the above systems after some trial and error. Earlier systems had more capability and more sophisticated options. They were also more expensive, more problematic to integrate into classes and less reliable overall. They required considerable effort from lab staff to keep them running. Even simple platforms using discrete chips to demonstrate Boolean algebra were less reliable than the microcontroller platforms, and the microcontrollers are far more flexible and reprogrammable. The systems using discrete parts obviously have a lower parts cost, but if the labor to design them, build them and maintain them is included the current systems are cheaper overall.

**Results**

We started with a single class. Due to strong student response and flexibility of the platform we are now using these units in multiple classes. Our analysis shows that the critical properties it offers to the learning experience are

- Immersive interactive learning
- Immediate of interactive control (direct manipulation)
- Tactile and experiential learning

The problems of this approach include the lack of experience of IT students with the HW

The key parameters to handling this problem are

1. Provide ready-packaged hardware that obviates the need to build circuits
2. Use a source of hardware that provides three key properties, IE a wide selection, standardized interfaces and software libraries for accessing the software

Educational impact evaluation at this stage is mostly qualitative. An example of the engagement aspect of the approach is shown by the game-design assignments in the computer architecture class. This is taught at an introductory level to students with little programming experience. By being able to write simple code and immediately see the results of a set of LEDs they by-pass the need to deal with the complex graphics environment in a modern GUI while, on the other hand the complete architecture of the system is available to them, not hidden under layers of “black-
Faculty in multiple classes report strong student engagement. Students are more willing to continue exploring concepts where they have complete control of the learning environment and frequently try variations on the original lab experiments. Faculty report multiple instances of apparent emotional satisfaction with students saying, “I wonder what it will do if I try this.” They can immediately explore options and then respond “Oh. Look what it does, I wonder why?” Furthermore, due to the flexible, open-ended and multi-modal nature of the learning systems faculty often propose more sophisticated variants to advanced students while finding it easy to demonstrate simple principles to struggling students. Faculty and student responses to using these systems have been very positive. The instructors report no negative effects from using the hardware and considerable satisfaction and good student learning. Students in class surveys and informal discussions report satisfaction in understanding the concepts with this approach.

On the whole we report satisfaction in the learning experiences and the student engagement. More formal learning evaluations, while desirable from an educational-theoretical standpoint, will be difficult as controlled studies suggest the necessity for some students using the previous non-hardware approach and both faculty and students are reluctant to do that.

Conclusions

The benefits of using interactive hardware platforms for teaching some IT concepts are strong. Trends in IT towards much more diverse computing platforms that are integrated into the real world indicate that IT students should pay more attention to hardware systems.

Hardware adds a significant dimension to the learning experience of IT students. Commonly available systems provide the necessary elements for introducing hardware into IT courses. The benefits to students are clear. The framework we have developed leading to the use of widely standardized systems has been demonstrated to be effective in multiple classes and will be continued and extended.

Bibliography

11. IEEE Internet of Things” http://iot.ieee.org, accessed Feb
15. Peter Rosati, “Students' Psychological Type and Success in Different Engineering Programs” Frontiers in Education, 1997