Development of a digital instrument as a motivational component in teaching embedded computers

Gracián Triviño[1], Felipe Fernández[2]

[1]Universidad Politécnica, Madrid, Spain, gtrivino@fi.upm.es
[2]Universidad Politécnica, Madrid, Spain, felipe.fernandez@es.bosch.com

Abstract

Nowadays it is frequent that, at the first levels of Computer Engineering studies, the students have acquired some practical experience developing software projects. However, they have little or none experience facing up the project of designing and developing a simple computer electronic module.

On the other hand, due to the strong development of the area and its continuous presence in the media, students are especially motivated towards robotics and, in general, towards systems that can interact with the physical environment.

According to these circumstances, this paper describes the strategy followed to introduce a new subject denominated "Digital Instrumentation and Data acquisition". This subject is an optional part of the curriculum area dedicated to Computers Architecture in the Faculty of Computer Engineering at the Polytechnic University of Madrid (Spain).

1. Introduction

Two years ago, we faced up the challenge of designing the layout of a new subject in the area of Computers Architecture. In teaching embedded computers, it is especially clear that the goals of teaching are not only a set of theoretical concepts. Together with them, you need to teach practical procedures, and to teach attitudes encouraging the students to develop a personal interest in studying the topics related with the subject.

Considering the characteristics of the current curriculum, the students acquire during the first courses some capabilities to develop software projects while they have little or none experience designing and developing computer electronic circuits. One elemental rule in teaching consists of using the available knowledge in the students' minds as the basis to build the new knowledge structures. Therefore, a first idea was to find a way of using this capability for software development as one of the basis of the new subject structure.

On the other hand, the new subject was going to be one more in the set of optional subjects that are available for the students choice. Then it was necessary to think about special motivations, the marketing strategy that would encourage the student to introduce our subject in his/her course configuration.

During last years, an increasing interest among students on topics related with robotics has been detected. More specifically, students are interested in computers that are able to interact with the physical environment. Therefore, a second idea was to use this interest as a motivation to convince the students to enrol in our subject.

We decided to name this undergraduate course "Digital Instrumentation and Data acquisition". Under this denomination, our intention was to cover a space detected in our curriculum of the Computers Architecture area. A classification of the didactical contents of the new subject is the following:

Theoretical contents

- Different types of sensors making emphasis in the applied physical principles.
- Differential amplifiers, instrumentation amplifiers, A-D converters.
- Instruments, instrumentation systems, standard instrumentation platforms (GPIB, VXI, PXI)
- Instrumentation languages, instrumentation software environment.

Practical contents

- Design and building of an electronic circuit for data acquisition based in microcontrollers.
- Design and building of electronics circuits to handle the signal provided by different sensors.
- Design and building of software programs handling this hardware to create digital instruments.

The complete subject program can be found in the subject Internet web page [1]. The remainder of this paper is aimed to describe different aspects of the resources developed to support the teaching of the practical contents.

2. Practical Project

As a part of their learning activities, the students of "Digital Instrumentation and Data Acquisition" must develop a *practical project* that consists of building a digital instrument.

Through the analysis, design and building of a digital instrument the student learns practical knowledge that complements the corresponding theoretical concepts and procedures.

When we faced up the development of some resources to support the students during the *practical project* development, we considered some special requirements having in account not only a teaching strategy but also an adequate marketing strategy:

- The student must have the possibility of using his/her own personal computer (PC) as an important component of the practical project.

The idea is to use the fact of that most of the activity of students developing software is currently performed using their personal computer.

- The whole hardware of the *practical project* must be compact and small enough to be portable.

This is not only, because that could make more comfortable the project development, but also, because the students' interest in this type of practical projects allows using the project itself as a marketing reclaim. The student will carry the circuit board with him/her having the possibility of talking about it and showing it to fellows.

- After the *practical project* is finished, the system must be able to be used for new projects. Consequently, all the resources used must remain available to the student. If the *practical project* hardware is of student's property, it will remain available not only for other projects at the university but also to develop his/her own home projects. To make this possible, it is desirable to maintain the whole project as low cost as possible.

3. Hardware resources

The main support of the *practical project* is a printed circuit board that we have called iFOTON. This circuit contains a microcontroller with few additional electronic devices and a free mounting area where it is possible to solder additional components. Figure 1 shows a iFOTON block diagram.

This circuit includes all that is necessary to make easy the development of an electronic digital instrument. Figure 2 shows a photo of the PCB with all the electronic components mounted. A brief description of iFOTON main features is the following:

Microcontroller

The PIC16F873/76 microcontroller from Microchip [2] has been chosen. This family of devices has an excellent cost-benefit rate and its use in the market has been growing up during the last years. This microcontroller has a set of interesting characteristics that we are only to enumerate: It has a RISC architecture, 4K of flash program memory, 193 bytes of RAM, 128 bytes of EEPROM. Moreover, the encapsulated peripheral devices are: A/D converter of 10 bits, 3 timers, PWM modules, communications modules (USART, I2C) and 3 parallel digital input output ports.

Serial port

It is the main mechanism of communication between iFOTON and the PC. The USART provided by the microcontroller has been used. A Maxim MAX232 serial signal adapter has been used in order to convert the TTL signal provided by this peripheral to the RS232 protocol logical level [3].

Programming port

An important advantage of the selected microcontroller is the so-called "In Circuit Programming" characteristic [4]. This allows, using only a 5 volts power supply and handling a limited set of connections (4), to read and write the microcontroller program memory without extracting it from the circuit socket.

The PC standard parallel port is used to handle these connections. This port fulfils the IEEE 1284 (SPP, EPP, ECP) standard signalling method for a bi-directional parallel peripheral interface for personal computers [5]. To make the connection the student must build the adequate cable following the instructions provided with the iFOTON documentation.

Input output port

The microcontroller provides three configurable input-output ports that have been situated near the free mounting area:

- All of them can be configured as digital inputs. They have associated internal programmable pull-ups and the possibility of associating encapsulated devices to interrupt processing and to perform pulse wide measurement.
- All of them can be selected as digital outputs. They have 25 mA sink/source current capability.
- Five pins of them can be configured as analogical

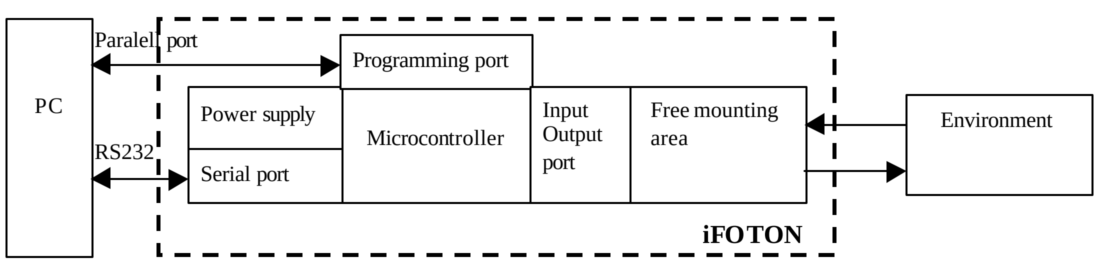

Figure 1. iFOTON Block diagram

inputs: These inputs share the multiplexed A/D converter.

Power supply

IFOTON uses the DC power supply that is generally provided with the PC loudspeakers. This device has a jack connector that provides 9 volts where the central connection is connected to ground. Then a simple 7805 voltage regulator and two capacitors are enough to complete the circuit power supply.

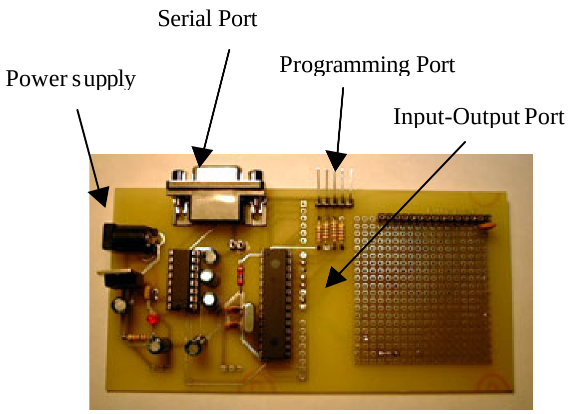

Figure 2. iFOTON PCB

4. *Software resources*

A set of software tools has been developed to support the design and building of *practical projects* with iFOTON: a Programmer, a Project configuration control, and an Interpreter of commands. The first two of them are executed in the student PC and the third one in the iFOTON microcontroller.

Code Programmer

This is a software tool that converts iFOTON in a microcontroller programmer [6]. This program uses the connection between the PC and the iFOTON Programming Port to allow the user reading and writing the microcontroller program memory. The object file to be programmed must be HEX formatted [7]. This format is generated for most of the assemblers and compilers available in the market.

Project Configuration Control

Together with the Code Programmer, a simple Project Configuration Controller is provided. This program allows creating a files structure for every *Practical Project*. It creates in the PC disk a directory with the project name containing a set of mandatory files: Requirements, Design, Diagrams and Software source. The Project Configuration Controller asks the user to associate these files with the corresponding software tool used to generate them.

Using iFOTON a fairly amount of different *practical projects* could be developed. The idea of this tool is to get a standardised set of documents for all these projects. On the one hand, this will make them easier to be analysed and evaluated by the teacher, and on the other hand, this will help to build a projects database easier to reuse.

Interpreter of commands

Taking into account the availability of the microcontroller programmer, a first approach in order to build a digital instrument could be to develop software to be loaded in the microcontroller memory. This allows building a stand-alone instrument. However, as mentioned above, the possibility of allowing the student to develop software to be executed in his/her personal computer is desirable. This avoids the need of knowing details about the microcontroller programming features.

In order to make this possible, an Interpreter of Commands, that has been called iF, has been designed and implemented. This software, loaded in the microcontroller memory, allows managing the iFOTON input-output ports using commands, which are sent and received via the serial port.

Header	"$"
Command code	Access to ports is performed using two characters: First: "L" means Read, "E" means Write Second: "D" means Digital, "A" means Analogical There are other special commands: "S" means Scanner "M" means Stepper Motor control Etc.
iFOTON address	This allows connecting several iFOTON systems to the same serial bus
Port address	Meaning the specific pin concerned
Data	Used with writing commands
Final code	";"

Table 1. Commands format

To simplify the design of the Interpreter, a set of restrictions have been introduced assigning specific functions to every iFOTON Input-Output Port. The port A, pines 1 to 5 in the iFOTON Input-Output Port, is used as analogical input. The port B, pines 6 to 13, is used as digital input. And, port C, pines 14 to 22, is used a digital output.

Therefore, by executing the iF interpreter in iFOTON and running a terminal emulator as the Microsoft HyperTerminal in the PC, we can use the keyboard to send commands and the screen to read the answers.

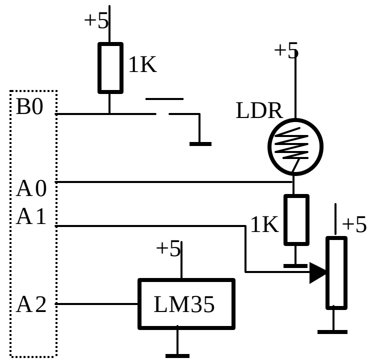

Figure 3. Sensor connections

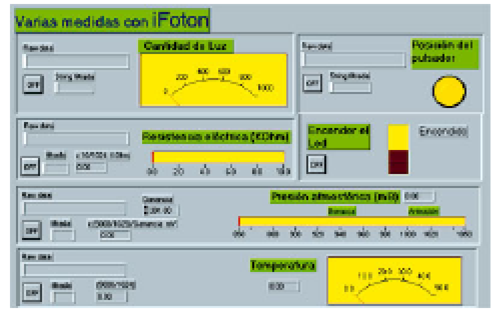

Figure 4. Instrument user interface

A command is a string of ASCII characters with the format described in table 1.

The following are some examples of using the iF commands:

"$LA00;" reads the analogical value of voltage in pin 0 of port A. The answer could be $0127; a value in the range 0 -1025 meaning a value in the range 0 - 5000 mV.

"$LD02;" reads the digital value in pin 2 of port B.

The answer could be $0; meaning a low digital TTL level input.

"$ED050;" writes the digital value 0 in the pin 5 of port C.

As far as commands that are more specific are considered useful, they are implemented expanding in this way the iF interpreter possibilities.

5. *Documentation*

Most of the documentation that is required to undertake the *practical projects* is provided to the students using the iFOTON web page in Internet [8].

The aim is to provide not only the necessary information to allow the students to solve their projects but also to help the advanced students to get detailed information and to explore new possibilities of use.

This set of documents includes iFOTON description, iFOTON building manual, iF commands description and some examples of *practical projects*. The hypertextual documentation includes links to the main company providers of the electronic components used. Logically this information must be in continuous evolution.

6. *Examples of practical projects*

6.1. Environment data acquisition

This *Practical Project* consists of representing in the PC screen the data obtained from the following sensors: a push button, the position of a potentiometer, the room temperature, the light intensity in the room, and the atmospheric pressure.

All the necessary electronic components can be allocated in the free soldering area. The chosen temperature sensor was the LM35 from National that can be connected directly to an analogical port. A LDR sensor was used to measure the light intensity. Figure 3 shows the simple circuits that must be mounted in the free soldering area to handle these sensors.

The MPX2200A sensor from Motorola was chosen for absolute pressure sensing. The provided signal by this sensor needs to be amplified. The students must analyse different instrumentation amplifier circuits [9] and build one of them to handle this sensor signal.

Once the hardware components have been mounted, all the measures can be obtained directly using iF commands.

The software part of this *Practical Project* is aimed at learning how to program with a specific software tool for the development of the so-called *virtual instruments*. The students must learn to design and build one of these instruments using the visual language LabView from National Instruments [10]. They can use an available free cost Student Version of this program [11].

Figure 4 shows an example of this *Practical Project* user interface that has been developed using LabView.

6.2. Design and building of a simple Digital Analyser

The *project* consists of building a simple Digital Analyser with only one input and limited buffer of memory and resolution. The design of this second example of *practical project* is based on using the special iF command "$S". The Digital Analyser probe is connected to pins C1 and C0 that must be tied together. The "$S" command uses the microcontroller capabilities for pulse wide measurement. It obtains in microseconds the time the digital signal remains high and the time the signal remains low. Using these values, the answer of this command is a text string that contains 40 values of time intervals starting with the first change of level. For example "$U:64554C: 473U:50244C: 101..." where U: means up level, 64554 are the microseconds that the signal remains high, C: means down level, and so on.

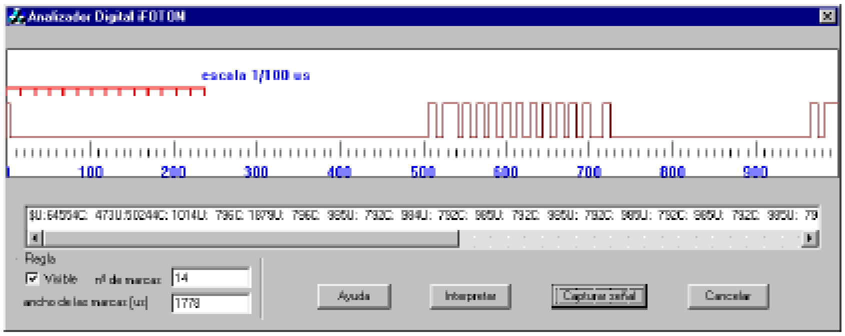

Figure 5. Digital Analyser user interface

Students will use these data as input to the PC software module.

For this *practical project*, students are required to develop the software using a generic programming tool as the C++ programming language. Using, for example, a tool as Microsoft Visual C++ software development environment, the design of a graphical user interface is not very difficult.

To test the Digital Analyser the student is asked to connect to the probe the signal provided by an Infrared Sensor (TFM5360 from TEMIC). When this sensor captures the signal emitted by a TV infrared remote control, it provides a TTL signal formed by a sequence of pulses.

Figure 5 shows an example of a Digital Analyser user interface. The screen shows the response obtained after receiving a signal from a TV PHILLIPS infrared remote control.

7. Conclusions

The following conclusions have been obtained after have experimented the new subject pedagogical structure for two courses.

"Digital Instrumentation and Data acquisition" is a subject especially orientated to students interested in having a double skill in hardware and software.

An important number of students of software would like to study and build hardware as well. They appreciate the possibility of building a physical system as opposite to developing systems exclusively made with software components. In fact, we think that the *Practical Project* is one of the causes for that, in this period of two years, the number of students that has chosen our subject has been duplicated.

iFOTON has demonstrated to be useful to support the students *Practical Projects* and also as a basis to develop some others prototypes of data acquisition and control systems. It is meaningful that we have received from Internet some demands for purchasing iFOTON.

The developed set of resources provides a powerful tool for teaching digital instrumentation. However, we can make an additional effort improving the Project Configuration Manager. If we make an additional development, it will be possible to create a library of hardware-software reusable modules. This library will be available in Internet.

References

[1] http://www.dtf.fi.upm.es/~gtrivino/iad.html

[2] PIC16F87X. 28/40-pin 8-Bit CMOS FLASH Microcontrollers. Document DS30292B Microchip (1999).

[3] http://www.dtf.fi.upm.es/~gtrivino/IFOTON /1798.pdf

[4] EEPROM Memory Programming Specification. Document DS39025E Microchip (2000).

[5] IEEE Standard Signalling Method for a Bi-directional Parallel Peripheral Interface for Personal Computers. IEEE 1284-2000.

[6] Spur, R. *A PC-Based Development Programmer for the PIC16C84*. Application Note AN589, Microchip, 1999

[7] Richey, R. (1998): Downloading HEX files to PIC16F87X PIC Microcontrollers. TB025. Microchip.

[8] http://www.dtf.fi.upm.es/~gtrivino/IFOTON

[9] Paton B.E. *Sensors, Transducers, and LabVIEW.* Prentice Hall, 1999

[10] Bishop R.H. Learning with LabView. Addison-Wesley, 1998

[11] Jamal R., Pichlick H. *LabVIEW applications and solutions*. Prentice Hall, 1999