XVII IMEKO World Congress
Metrology in the 3rd Millennium
June 22–27, 2003, Dubrovnik, Croatia

INTEGRATED LEARNING SYSTEM FOR DATA ACQUISITION

Radu Dobrescu, Radu Varbanescu, Ana Lucia Varbanescu

POLITEHNICA University of Bucharest, Faculty of Automation and Computers, Bucharest, Romania

Abstract – The paper presents an enhanced learning and training system in data acquisition. This new method is the result of an interactive process for enhancing the learning activities for the students. The system was built to meet a new concept: "integrated learning", an approach that allows students to exercise their abilities in real applications. These applications are entirely specified, having well-defined constructive structures and well-known finish lines. The hardware platforms are built in layers, the most suitable solution for the multi-field approach. Working like this, the students have the opportunity to work on each level so that their solutions meet the requirements for each part of the application, and also take into account the compatibility restrictions.

Keywords: data acquisition, integrated learning

1. INTRODUCTION

The paper defines a learning system that is used by the students from the terminal years of study in our University, in the Faculty of Automation and Computers, Department of Data Processing and Transmission. This system combines the classical issues to be learnt about measurement with the modern methods based on computer aided processing.

While designing this system, there were two main goals: to develop the right laboratory hardware platforms and to implement the working methods that allow the students to go through all the stages of the learning process.

2. HARDWARE PLATFORMS

The lab platforms are built in layers, as hierarchical systems for data acquisition (fig.1).
The structure we have designed has three layers:

Layer 1 is the input level (the low level), containing the acquisition devices – sensors and transducers – linked to some conditioning devices [1]. The outputs of this layer are inputs for the second layer, and they can be accessible to the students for analysis.

The measuring devices are selected and placed depending on the application to be solved. The application also dictates the performances and features required for the sensors and transducers.

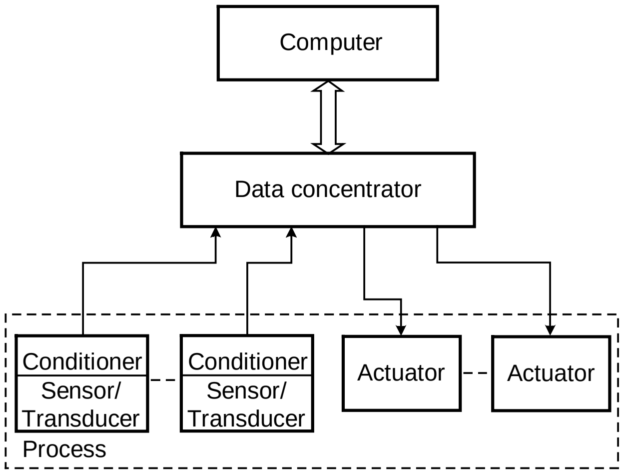

Fig. 1. Platform structure

Remember that the main processing unit is a digital one. Thus, the interface between the measuring devices and this main unit is not always simple – the signal levels may not be compatible. In this case, the Conditioner modules are required. These interfacing conditioners are, in fact, signal amplifiers, isolation modules or filters.

Other problems might appear if the system integrates both the acquisition and the control modules. If so, the corresponding actuators should be placed in the process; these elements are also selected based on the performance criteria required by the application.

In this system, all the parts can be tested in two different ways: the first approach allows the classical individual testing, in which a component is verified using multimeters or oscilloscopes. But besides this approach, the integrated learning systems allow the user to test the functional behaviour of any component, and to check its parameters in real conditions. So, without losing the independence of any part, the system allows a complex functional testing for both parts and modules.

Layer 2 is a module for data acquisition and primary data processing (the middle level) [2]. Its main component is a microcontroller. This module can also work as a control microsystem, being able to return several control signals to the process [3]. The third layer designs the algorithm that supervises the acquisition and control operations, and its decisions are based on the measured values outputted by the first layer.

The control microsystem is analyzed from the constructive point of view. Being coordinated and controlled by a MCU (microcontroller unit), it is equipped with resources that allow data acquisition, basic data processing, results displaying and data distribution. The application range is quite large, and its only constraints are the available resources of the MCU. A microcontroller has all the resources embedded in only one chip, thus allowing the control microsystem board to be easily built and then used in developing a large variety of applications. As the memory modules, the serial and parallel interfaces, the LCD display controller are inside the microcontroller chip, only the interfacing modules (the conditioners) for the connection with Layer1 are added on the board. Figure 2 presents this structure as a block diagram.

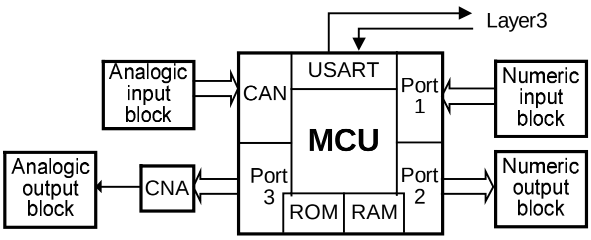

Fig. 2. The data concentrator

When developing a MCU-based application, there are 4 steps to be followed:

1. Choose the needed MCU
2. Write the control program using specific assembler language
3. Simulate and debug the program
4. Write the code into the program memory of the MCU (ROM module)

When finishing step 4, the MCU system will work as a standalone controller. Meanwhile, all the settings, tests and debugging sessions are made using the Layer3 computer (to be discussed below).

The link between Layer2 and Layer1 is made by the conditioner interfaces. The controller has some built-in interfacing modules – digital and analogical I/O ports – but there are many situations when external interfaces are also required. Typically, the digital signals are directly connected to the ports, while the analogical signals are routed through analogical multiplexers connected to the appropriate inputs of the microcontroller.

The link with the higher level (Layer3) is made via serial connection, as the microcontrollers have at least one built-in serial interface used for communication with the programming console (usually a computer).

So, the bottom line is: use a smart MCU as the main component of the Data Concentrator, and this will really simplify the rest of the interfacing problems, simplifying the Layer2 architecture.

Layer 3 (basically, a personal computer) is the complex processing unit (the high level), used for the management of the entire system. The strategy for running the application is developed here. Complex acquired data processing is also implemented at this layer, being augmented by the graphical and/or alphanumeric user interfaces [4]. It is the same computer that designs and implements the control program for Layer2, being able also to test and debug it. After the program is considered functional, it will be downloaded to the microcontroller system, enabling the latter to control those two lower layers and to give preprocessed data as feedback.

Please note that – although the levels are linked together – the inputs and outputs of each one can be displayed and read by the users/students. This – of course – is a didactic tool only, but it allows easy debugging and understanding of the embedded system.

3. THE LEARNING METHOD

Based on the presented hardware structure, we developed a learning method called “integrated learning”: the user has to perform data acquisition and management on each of the layers, going through all the steps starting from the measured data at the output of the sensor/transducer and ending with the graphical interface for the entire application.

This approach is possible because the students already have the necessary basic hardware and software knowledge for operating at each one of the three layers.

The lab activity for each platform follows the same pattern: it starts from a set of performances that should be reached by the application to be built. Fully functional examples are available for the students to study and, based on these examples, they should build their own solutions, meeting the functional and performance requirements.

These solutions require specific operations for each layer of the hardware structure. On the first layer, the acquisition devices should be well chosen, integrated in the application and calibrated for the application needs. On the second layer, the focus moves on programming skills, as the students should write some acquisition routines using the low level language of the microcontroller. Still, there are some hardware related activities, as they have to identify the correct communication links between the layers one and two, in order to be able to use the correct I/O ports. On the third layer they should implement the user interface. The students will use a high-level programming language to generate a software application that will allow the user of the entire system to manage and control it fast and easy.

Each application will follow the same pattern, illustrated in the following sequence:

1. Define and analyze the application
2. Choose the Layer1 components
3. Test the Layer1 functionality
4. Configure the control microsystem
5. Develop the control program
6. Is the control program working?
   NO: go to 5
   YES: continue
7. Design the user interface for configuring the application
8. Do the performances meet the requirements?
   NO: go to 2

YES: continue

9. Write the control application into the program memory of the MCU

Let us continue by adding some details for each one of these nine steps, thus proving that the solution is efficient and very student-friendly.

Step 1. Define and analyze the application

Goal: The students will learn about the application analysis pattern, and how to respond to a customer request by using a clear algorithm. After this step is completed, the requirements and the constraints coming from the process (the system to be controlled, the application the customer requests) are established.

Means:

- define the structure of the process
- define the functional parameters
- set the general level of performance required
- design the acquisition system structure
- define the system inputs and outputs
- set the accuracy level for the acquisition system.

Step 2. Choose the Layer1 components

Goal: The students are taught how to choose these devices from the various offers. They would learn how to read and interpret the parameters that influence some constraints and what decisions can be made in typical situations. By presenting real-life applications, they will be force to read the catalogue specifications and to learn how to use them.

Means:

- choose the measuring devices (transducers and sensors), respecting the constraints from Step 1
- place these devices as required by the application
- connect the devices – with or without external interfaces
- choose and place the actuators

Step 3. Test the Layer1 functionality

Goal: After choosing all the components of the first layer – the hardware part of the application, the students have to find a way to test their options. They will build data sets for testing and learn how to interpret the results according to the defined constraints.

Means:

- build data sets and their expected results
- test the equipment behaviour using the test data sets
- make the necessary adjustments for better performances

Step 4. Configure the control microsystem

Goal: The students will learn how to choose the components for Layer2. First of all, they will have to compare some microcontrollers and to decide which one is the best. And why. The criteria for this analysis should be the resources provided by the MCU as well as the available interfacing capabilities. Finally, the controller should allow multiple programming/debugging sessions, being part of a dynamic design process. As a direct result of this step, the students shall obtain a draft design for the Layer2 board, allowing them to build the data concentrator.

Means:

- obtain a list of available microcontrollers, analyze each one of them to meet the imposed criteria and choose the best one of them
- design a resource map of the MCU
- establish the memory and I/O requirements
- define the communication protocols with Layer1
- establish the auxiliary hardware needed – the interfaces that may be required between Layer1 and Layer2.
- draw a block diagram with all the components that are to be interconnected
- design the Layer2 board

Step 5. Develop the control program

Goal: The students learn how to program a MCU system. The general theoretical rules and the available programming languages are presented. The students are required to remember the secrets of assembly language programming, as they will use this low-level language for programming Layer2. They will further develop their skills to understand processor architecture and to program it so that all its features are used.

Means:

- establish the control algorithm
- write, test and debug the acquisition routines for all the measuring devices
- write, test and debug the control routines for the actuators
- design the entire algorithm
- develop the source file for the application, by embedding the algorithm structure with the functional routines
- test the entire program

Step 6. Is the control program working?

Goal: In this point, the students will make the first tests to see if the control application is really working as planned. If not, they have to go back to step 5 and review all the decisions made there. This is a part of software engineering they will have to use. It is – again – part of the real-life design process for commercial applications.

Means:

- design test data sets and their required results
- write some test routines for the MCU board
- test the regular cases as well as the special ones
- if there are any errors, do not just avoid them, but go back to step 5 and correct the software.
- test the program against the worst case scenarios
- do not forget that the user behaviour is hard to predict, so learn how to program in a defensive manner.

Step 7. Design the user interface for configuring the application

Goal: The customer is not always a specialist in computers. So, the students will have to think about the user interface. They will have to build a software bridge between

Layer2 and Layer3, providing the customer with high-level means for configuring the application.

Means:

- design a user interface for the Layer3
- establish the communication protocol between Layer2 and Layer3
- design and implement the software bridge (its goal will be to convert high-level data into low-level configuring scripts, that will result in application routines to be downloaded in the MCU's program memory)
- test the application generated by this software converter

Step 8. Do the performances meet the requirements?

Goal: The students have to learn how to make a real performance analysis of their newborn system. They have the milestones imposed from Step 1, and they have to test the system against them. If the performances are not as planned, the design process must be resumed from Step 2. Although this may appear as very difficult to the students, they have to learn that all these errors are part of the design process and have to be dealt with during this process.

Means:

- turn on the application
- try the best case scenario
- try the worst case scenario
- compare the results (they are already proven to be correct) from the performance point of view – time of response, resource management and so on
- find a way to measure all these for the real case and compare them with the imposed limits
- if the results exceed the previous defined limits, go back to Step 2.

Step 9. Write the control application into the program memory of the MCU

Goal: The students have to become familiar with downloading the code into the program memory of the MCU – either EEPROM or Flash. For them, this is the happy end. It is the final step of the application, when it becomes completely independent. The students have to reach this point so that can actually say "It works! It actually works!". The system is functioning as a standalone application. They will have only to test and document the results.

Means:

- present the options for downloading the code
- explain the differences between ROM, EEPROM ad Flash for the particular MCU they used
- program the chip
- start the standalone application

We have proposed a 9-step integrated learning method. The students come at the beginning of a semester with their basic knowledge in data acquisition, both hardware and software and, besides actually practicing, they learn to combine all these. By the end of the 3-moths semester, they have built a system from scratch.

This conclusion ends the theoretical part. A short example of applying this method is presented in section 6. It will be a practical approach for all these theoretical steps, in order to prove that they can be used in a real laboratory.

4. WHAT IS NEW ?

4.1. *Hardware-software interaction*

The students are used to learn about hardware and software in completely different courses. This approach forces them to combine them and to study their interaction.

The most important issue in this hardware-software interaction is the timing. All the actions shall respect the minimum timing limits imposed by the application. So, if the acquisition is too fast, the data processing must be optimized, avoiding data loss.

The idea to be emphasized is that the acquisition problem includes both the software and the hardware. Solving the problem means working on the layers, and adjusting both the software programs and hardware devices.

4.2. *Step-by-step testing*

Such an approach supplies the student with an excellent testing system. In each step, one can implement various tests, both regular and exceptional cases. The important thing is that, as the system is growing, the tests are becoming more complex. Still, this complexity is achieved not by generating more and more test from scratch, but by combining the lower level tests in different scenarios.

4.3. *User interface issues*

The students are not used to take into account the user interface. Here, they are forced to deal with it. This is a double challenge: first, they have to design a user-friendly interface, using a high-level language and, second, they have to link this interface to the low-level control program. Basically, what they get after all this hard work is a fully functional acquisition system, with a user-friendly interface and effective configuration options.

4.4. *Time-to-market*

This integrated approach has one overwhelming advantage: it takes less to be built. Given a complex acquisition system to be built, there are two basic roads to be followed: either design every module (every layer) in a specialized department and then assemble the results or design the entire system in this incremental approach. Even if the latter option is not very suitable in real life, it is an excellent opportunity for the students to train in all the fields involved here. They will also develop project management skills. They actually meet the big picture, not just a small part of it.

5. LAB STRUCTURE

The lab we are discussing here has enough platforms so that the students can work on each one of them in small teams. Some of the applications we designed are : a small railway with electrical trains, a mobile robot that moves around in a preconditioned space, an aquatic ecosystem.

The laboratory is always opened for students. Still, there is a time limit for each one of the problems they have to solve. They are given all the details and functional examples, so they have to build their own solution for it. Finally, they should present the solutions in front of their colleagues, thus improving the ability to publish their work.

6. APPLYING THE METHOD

As an example of a real implementation of this theoretical method, we have chosen to present an application for controlling a small railway network with electrical trains.

So, the process setup is as follows: a fixed railway network, with a dynamic topology implemented by railway switches and at least two trains traveling by. The user must have the possibility to configure the switches positions, thus choosing his/her preferred topology. The system should implement the acquisition and control algorithm for the safe traveling of all the trains, avoiding collisions and allowing dynamic modifications in railroad topology.

Step1: After analyzing the scenario, we have discovered the following solution: we will split the entire network into regions, so that collisions would appear only if the trains are coming from different regions. In other words, if the two or more trains are in the same region, there is no collision danger. It is also important to consider that all the trains are traveling at the same speed. If a collision is imminent, the train that first detects this dangerous situation will stop, allowing the others to go on.

Step2: There are two kinds of components here: sensors for detecting the train passing by (implemented with magnetic proximity sensors) and actuators for changing the railroad-switches and for stopping the trains. (implemented with electrical relays). There are 8 different regions, so we will have 8 sensors, 8 relays for the railroad-switches and 8 relays for stopping the trains (by turning off the power source). So, we will have 8 data inputs and 16 command outputs, all of them being digitized.

Step3: It's time for the first tests. Here, the sensors and the actuators have to be tested. It is enough – for now - to use only one train to see if the sensors are responding correctly, to verify that the train stops when turning off the power and to check the switches commands. For example, this testing session allows the students to see that by placing the sensors too far from the joint point, they will detect too many possible collisions. A very close placing will not leave the train enough time to stop, so the collision would appear anyway.

Step4: The students have to choose between two microcontrollers. They should select the one with a serial interface and multiple I/O ports. Because this MCU (8052 family) has enough numerical ports, there is no need to activate the external interfaces. The link between Layer1 and Layer2 is a simple connection, the signal levels being also compatible.

Step5: It's time for developing the software application. By using the assembly language, the students are implementing the readings and writings to the corresponding ports. After that, they have to think about implementing the possible collision conditions. Finally, the algorithm will have to stop the endangered trains and also start them after the danger is gone. Keep in mind that this control software shall synchronize all the actions.

Step6: This is the second testing session. The students are testing now with two or more trains to see if they implemented everything correctly. Usually, after lots of collisions and "back to 5" steps it finally works! The result of these 6 steps is a functional control application that rules the traffic on the railway network.

Step7: When designing the interface for the user, the students shall allow him/her to change the switches, they shall present the sensors as active or passive (train passing by or not) and to announce possible emerging collisions. A screenshot from teacher's solution is presented in figure 3. Keep in mind that, even if the program is a high-level one, it is also a bridge between Layer3 and Layer2, allowing the user defined topologies to be sent to the MCU in a real-time manner, without stopping anything. This is a real challenge, in both high-level and low-level programming!

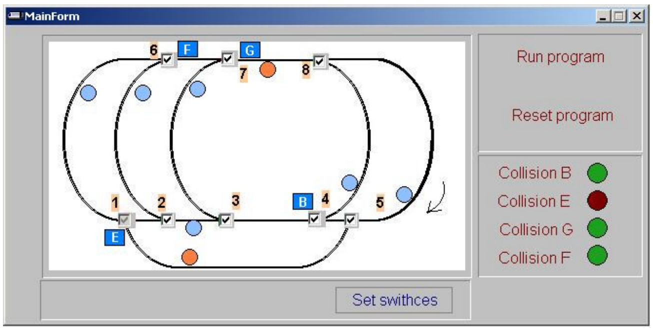

Fig. 3. A screenshot from the high-level software application

Step8: This is the final test: if the system passes the performance tests, it is declared finished and can be downloaded as a "final release" in the MCU. The problems that may appear are related to the delays added by the Layer3 software. Worst-case scenarios must be tested. As an example, we had another version of this system using sensors based on photocells. Unfortunately, they were not functioning well in different light and temperature conditions, so we did go back to Step2 to choose another solution for the sensors.

Step9: In this case, because the user is dynamically configuring the application, the system will not be disconnected from the Layer3 computer. All three layers are working "full-time". The only problems that might appear here are related to memory constraints (the control application may not fit into the program memory of the MCU).

In our lab, we present this solution to our students. They will have it as an example, without seeing any of the constructive parts. Still, they are really challenged and it seems that building something from the blueprint to the working device is far more appealing than smaller lab applications.

7. FURTHER DEVELOPMENTS

The next goal for our laboratory is to provide some of these features online.

Basically, there are two different directions:

- e-learning – the discussions between the teacher and the students are not live – all the protocol is implemented by posting messages
- distance learning – the entire learning process is live, by creating a virtual laboratory environment.

For e-learning, the system will loose part of its efficiency. The students cannot experiment everything by themselves if there is no-one in the real laboratory to provide technical support. For this type of application, we suggest a scenario based on simulators. After the simulation is considered correct – this appreciation can be made automatically, by a software application replacing the teacher – the student gets access to the real process. In the real process, the student gets only the right to actually see if the simulation works for the real case. He/she has no permission to change the parameters from the simulation. In other words, no case study is allowed without first getting the ACK from the lab-authority software.

For distance-learning, the system is still working very well. The only loss is that the students cannot see the real process, but a movie with all the action. All the discussions are made live. The people that are in the physical laboratory are responding to the students' requests, they are correcting their decisions and, finally, they test the results together with the students.

So, the method can be adapted for online learning, with various scenarios. The most important problems to be solved are:

- building the virtual lab environment – an Internet site allowing the students to participate in the class
- designing and developing all the simulators needed for students to improve their understanding of the processes.

We have already started a project for the e-learning environment, based on simulators and only final-phase testing. The results are to be presented in a future paper.

8. CONCLUSIONS

We decided to present this learning system because it was highly appreciated by the students. They are interested in using real platforms and in finding solutions for real problems that imply knowledge from several different fields.

Major system advantages are :

- the students have to analyze a real application, specified by a constructive structure, with its parameters and performances.
- all the platforms support several applications with many different solutions, so the students can exercise their creativity
- the solutions to be built are very well specified and they can be easily tested, so the students will know whether they solved the problem or not.
- the platforms require students to apply their knowledge in several fields and to link them together; this integrated approach is far closer to the real life than presenting each field separately.
- the students are going to prove their hardware and software abilities, and they will be able to prove that they can manage all these fields and, also, they can control the links between them.

REFERENCES

[1] G. Ionescu, R. Dobrescu, R.Varbanescu, "*Industrial Automation Transducers*" / "*Traductoare pentru automatizari industriale*" , Ed. Tehnica, Bucharest, 1985

[2] R. Varbanescu, "*Computer-aided measurement systems*" / "Sisteme informatizate de masurare", Ed. MatrixRom, Bucharest, 1999

[3] K. Arnold, "*Embedded Controller Hardware Design*", Independent Publishers Group, 2002

[4] H.P.Messmer, "*The Indispensable PC Hardware Book*", 3rd Edition, Addison-Wesley, 2001

AUTHORS:
prof.dr.ing. Radu Dobrescu, POLITEHNICA University of Bucharest, Faculty of Automation and Computers, Department of Data Processing and Transmission, Spl.Independentei 313, Bucharest, Romania, 77206, +40214029105, radud@aii.pub.ro;

conf.dr.ing. Radu Varbanescu, POLITEHNICA University of Bucharest, Faculty of Automation and Computers, Department of Data Processing and Transmission, Spl.Independentei 313, Bucharest, Romania, 77206, +40214029105, b30ara@hotmail.com

prep.ing. Ana Lucia Varbanescu, POLITEHNICA University of Bucharest, Faculty of Automation and Computers, Department of Computer Science, Spl.Independentei 313, Bucharest, Romania, 77206, +40214029105, analucia@cs.pub.ro