A SOFTWARE PACKAGE TO FACILITATE TEACHING
INTRODUCTORY LEVEL MECHATRONICS

Gokhan Bayar\textsuperscript{1}, Orhan Olcucuoglu\textsuperscript{1}, Bugra Koku\textsuperscript{1}, and Ali Sekmen\textsuperscript{2}

\textsuperscript{1}Department of Mechanical Engineering
Middle East Technical University
Ankara, Turkey

\textsuperscript{2}Department of Computer Science
Tennessee State University
Nashville, TN

Abstract

This paper describes a software package that is designed to facilitate teaching an introductory course on Mechatronics in the Department of Mechanical Engineering at Middle East Technical University. This course attracts students from multiple disciplines. Being offered at the third semester, rather than focusing on theoretical aspects of different disciplines, this course focuses on the system integration aspect of Mechatronics systems and emphasizes this with hands-on laboratory exercises and a semester project. The semester project requires the design and construction of a simple closed loop system. To make the programming aspect of these projects easier, it is intended to develop the necessary software on a PC rather than an embedded platform. To facilitate this, a software package is developed. With the help of this package and a microcontroller board (Basic Stamp 2), students can easily setup simple closed loop systems, interface them to the computer and design a controller in Matlab or Simulink.

1. Introduction

Mechatronics is commonly described as the synergic integration of mechanical engineering (ME), electronics, computer science (CS), and electrical engineering (EE) disciplines for development of intelligent systems. When Mechatronics is defined, the emphasis is generally not on the interdisciplinary nature of this discipline but rather on the synergetic integration of knowledge from different disciplines towards designing and manufacturing machines. Rzevski defines Mechatronics as “An Engineering discipline concerned with the integration of mechanical, electronic, computer software and other technologies with the aim of designing and manufacturing effective machines” \cite{1}. Similarly, Histand defines Mechatronics as: “A rapidly developing, interdisciplinary field of engineering that deals with the design of products whose function relies on the synergistic integration of mechanical, electrical, and electronic components connected by control architecture” \cite{2}. Being a product oriented discipline, Mechatronics covers a wide range of application areas such as robotics, manufacturing, system
engineering, and aerospace. Since integrated design approaches have been widely accepted for designing complex intelligent and engineering systems, Mechatronics education is rapidly becoming critical [3].

Due to its highly interdisciplinary nature, Mechatronics field attracts students from diverse backgrounds including ME, EE, and CS. This diversity imposes some constraints on teaching Mechatronics especially at the introductory level. For example, ME and EE students may need to spend more time than CS students to be proficient in software design. Likewise, CS students do not have much interest in hardware design. Non-engineering majors may have even harder time in dealing with low-level programming involving hardware interaction [4]. Software systems that encapsulate hardware systems and hide implementation details may help students concentrate on system design and implementation rather than improving prerequisite programming skills.

Introduction to Mechatronics is a sophomore-level course offered every fall semester in the Department of Mechanical Engineering (DoME) at Middle East Technical University (METU). It is a core course for the Mechatronics minor program at the DoME and mainly attracts ME and EE students.

This course is expected to attract interests for introductory-level Mechatronics and provide experience on working in interdisciplinary teams. The ability to function on multi-disciplinary teams is one of the ABET required attributes of engineering graduates [5]. Mechatronics courses in general provide a unique opportunity to satisfy this ABET criterion. Working in interdisciplinary projects may be very difficult for students if they do not have the same level of prerequisite background. In addition, the misconceptions listed below may discourage students form taking such courses.

1. Mechatronics discipline is closer to ME than EE and CS.
2. Prerequisite requirements are too high.
3. Teamwork is harder compared to the other disciplines.

This paper introduces a software package developed to help students overcome the burden of low-level microcontroller programming to shift the students effort and concentration towards completing a project which focuses on system integration. This paper is organized as follows: Section 2 and 3 discusses didactic and architectural issues considered in this study. The motivation behind this work is presented in section 4. Overview of the hardware and software structures of the development platform are described in Sections 5. Finally, some conclusions are given and future work is motivated in Section 6.

2. Didactic Issues

Mechatronics engineers, by definition, integrate knowledge from different disciplines in manufacturing machines. Mechatronics education, therefore, not only has to offer courses from various disciplines, but should also provide the chance to practice the integration of this knowledge in designing and manufacturing machines. When hands-on applications for Mechatronics education is considered, it is thought that in order to increase the efficiency of the course, the students should be motivated by competition, should have fun during the course,
observe how theories taught in classroom can be used in practice in creating a product or solution. These factors are not only involved in improving the effectiveness of a course, but are also involved in attracting students to the course. Surveys conducted by Piguet et al. have shown that students are motivated most when they are given the chance to apply theory in practice and design a machine (i.e. a robot) [6]. Practical aspects of a course (hands-on labs and project assignments) generally attract curious students to the course. Studies have shown that up to 90% of what we see and touch is remembered whereas only 10-15% of what we hear remains. Therefore, beyond having the potential to attract curious students, hands-on laboratory practice is very important in order to facilitate learning of the theoretical information presented in classroom. Kolb defines learning as a process whereby knowledge is created through the transformation of experience [7]. Evidently the best way to experience something and learn how to do it is to actually do it [8] (sometimes over and over again).

Other major motivating factors stated in student surveys are competition and project level development [6]. Apparently the desire to win motivates learning, and project level assignments do provide the proper medium. Competitions in general are performed in teams and hence, competition not only motivates individual learning but also facilitates teamwork and cooperation among students. Within certain constraints, we do encourage inter-group interaction and cooperation as well.

All of these factors are considered when shaping the curriculum of the “Introduction to Mechatronics” course. We tried to add some fun factor to the course by properly setting the content of the semester projects. Even though the survey results of Piguet et al. does not suggest that fun is not one of the major motivating factors for students [6], we do think that fun acts as a catalyzing factor and its effects in teaching engineering courses has to be investigated further. Building robots and competing with them inherently introduces fun into the equation. Our experience shows that, a project that has a higher fun factor attracts the attention of more students.

To sum up, the “Introduction to Mechatronics” course is designed so that lectures and laboratory exercises are equally balanced throughout the course. There is a semester project which is suitable for competition-based evaluation. This project requires the design and implementation of a simple closed loop control system. Each year different project topics are assigned. Next section focuses on the alternative software-hardware integration environments in which such projects can be realized.

3. Architecture Issues

The product oriented nature of Mechatronics offers ways of synergistically integrating software and hardware in a single product. When the operation of such products are analyzed it is observed that there are basically three categories of architectures that are used in realizing these products: low-, high- and mixed-level architectures. In teaching low-level architectures, generally machine code or assembly programming language is preferred to create embedded applications on microcontroller or DSP based boards. High-level architectures are thought via programming languages such as C++, Java and Visual Basic by using desktop or laptop PCs
running different operating systems. Each of these approaches have pros and cons when taught alone, hence a mixed-level approach is adopted by some educators [9].

Using low-level architectures in teaching have the advantage of exposing the student with the lowest level implementation details of the hardware used (such as registers and interrupts). Going through a low-level course, the student will be able to design a system while knowing all the inner-workings of its components. Programming at this level however, can easily become tedious and cumbersome. High-level architecture based teaching on the other hand, offer students better programming and debugging environments. With the advances in the serial communication hardware (i.e. USB and IEEE 1394) PCs of today are more capable of being used in control applications. Larger scale implementations might be constructed faster in a high-level architecture basically due to the tools and utilities available on these platforms. However, if all implementations are taught by using high-level languages and their eye-catching integrated development environments (IDE), the PC might remain to be (or become) a black box. A mixed-type approach attempts to combine the advantages of teaching at low- and high-levels while eliminating the disadvantages.

In teaching “Introduction to Mechatronics” we have adopted a mixed-level approach. Even though this is an introductory course offered at the third semester, in this course we introduce students with a microcontroller. Various simple microcontroller-based labs (such as moving lights, VU meter, RC servo control) are implemented by directly programming the microcontroller. However, as the application gets more complex, the resources of a microcontroller might become restrictive. At this level, neither software nor design optimization is not the objective of the course. Hence, the semester project of the class would seriously benefit from a high-level programming environment providing the student transparent access to the low-level resources. Following sections focuses on the motivation behind the details of this mixed-level implementation and its implementation details.

4. Motivation: Why PICit?

Mechatronics being a product oriented discipline, its education requires the integration of knowledge from several disciplines. What is more important than knowledge in different fields of science and engineering is the practice of their integration. An introductory Mechatronics course at METU is offered at DoME. The purpose of this course is to introduce Mechatronics to sophomores. At the sophomore level focusing on establishing background in different disciplines is not very practical due to the vast amount of materials that has to be covered. Therefore, in this course, we focus on the integration aspect of Mechatronics and while doing this we try to encourage student by trying to add a fun factor to the projects in a competitive environment. Throughout the course students are introduced with the very basics of sensing, actuation and programming.

After a brief introduction of Mechatronics, students are introduced with different sensors, and in the lab practice use an infrared (IR) range finder (Sharp GP2D02) and light sensitive resistors. Similarly, actuation mechanisms are briefly introduced in lecture, then students practice with RC servo motors in the lab. Finally, a simple closed loop control system is implemented in the lab where the student sets up a (by using an RC servo and a light sensitive resistor) light tracking
system. In all of these laboratory exercises the students use Basic Stamp 2 (BS2) microcontroller board.

Due to students coming from different backgrounds and being sophomores, teaching low level microcontroller programming (i.e. assembler) is opted out. This is the main reason why BS2 is chosen as the microcontroller platform. BS2 is programmed with a Basic derivative language which is not hard to learn or use, therefore, in the past years we taught this language and done the experiments with BS2. However, learning Basic specifically for BS2 does not add much to students background in terms of preparing him/her for the upcoming years. Therefore, an alternative approach is adopted and rather than a low level architecture type, a high level architecture type is set up yet the low level microcontroller structure is preserved within this structure. By programming at the PC level students spend more time on the details of their algorithms and implementations rather than spending time to figure out tricky details of embedded programming.

PICit\(^{1.0}\) software package provides an easy to use interface and a simple set of commands that enables students constructing closed-loops systems in reasonable times without spending too much time on the details of microcontroller level programming and without the restriction of the microcontrollers memory resources. With the use of PICit\(^{1.0}\) it is intended that the students are given an environment which helps them to focus on system design and integration. The PICit\(^{1.0}\) platform can also be used in teaching introductory level control concepts (where there are no real-time constraints) as well as teaching introductory level mobile robotics classes.

5. PICit\(^{1.0}\) System

PICit refers to a system that is composed of one or more PIC microcontroller [10] based board(s) that is (are) connected to a PC via RS232 or a similar serial port. PICit\(^{1.0}\) is the first limited version of this design paradigm where BS2 is chosen to be the PIC based microcontroller board, and a set of its low- and high-level functions are exported to Matlab. The hardware and software components of this system are introduced below respectively.

5.1 Hardware: Overview of Basic Stamp 2

The Basic Stamp 2 (BS2) microcontroller from Parallax Inc. is an improved version of the original Basic Stamp. Stamps are easy to program and use, hence very popular at elementary level education applications. With its small footprint, these microcontrollers ideal for small scale Robotics and Mechatronics projects. However, at the development stage of a project, or in laboratories where various new applications are setup around these controllers every week, it is not very practical to use the chips as is. A prototyping board that Parallax calls the "BS-2 Stamp Carrier Board" provides the necessary solution. In our department a similar board had been built for facilitating the use of BS2 based application development.

Basic Stamp 2 is preferred mainly because its simplicity and availability. Table 1 summarizes the features of BS2. Since it is programmed with Basic language, learning curves of students is acceptable. Yet, when applications becomes complex and requires intensive computation, BS2 resources for programming becomes restrictive. Semester projects generally forces these limits.
Therefore, with software support BS2 is interfaced to a desktop PC and a set of high-level functions are provided to the student to facilitate programming and let students focus on the system design and integration rather than hassling with low-level programming details as explained in the following section.

Table 1. Features of Basic Stamp II

<table>
<thead>
<tr>
<th>Features of Basic Stamp II</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processor</strong></td>
</tr>
<tr>
<td>The BASIC Stamp II is built around a PIC microprocessor from MicroChip. The PIC is pre-programmed with a BASIC Interpreter that reads bit encoded tokens out of a serial EEPROM.</td>
</tr>
<tr>
<td><strong>Development Environment</strong></td>
</tr>
<tr>
<td>The program is compiled on a PC and then downloaded using a serial cable.</td>
</tr>
<tr>
<td><strong>I/O Capabilities</strong></td>
</tr>
<tr>
<td>The BS2 has 16 I/O lines. These can be used as digital inputs and outputs, and with a simple capacitor circuit as pseudo analog out or analog in lines. Any I/O line can be used for serial data as well.</td>
</tr>
</tbody>
</table>

5.2 Software: Basic Stamp 2 and PC Programs

“Introduction to Mechatronics” course is a lab oriented course that is completed with competitions among the semester project designs. Laboratory exercises throughout the semester prepare the student for the project. In these exercises students acquire basic practical experience in electro-mechanical systems (sensing and actuation) and microcontroller programming. The purpose of this course is to introduce Mechatronics to the students as implied by the name of the course. This introduction, however, also aims to attract the attention of the students to the concept at an early stage of their education. By providing a perspective about how they theory taught in class can be applied in practice, students keep their attention and curiosity alive throughout the rest of their education. For this reason, the semester project is assigned in parallel with labs. By the time the project is assigned the students have almost no idea about how they can solve the problem presented with the project. However, by providing the problem upfront, they do pay more attention to the labs, concurrently start thinking about the project and come up with purposeful questions. BS2 provides an easy to use IDE yet, its computational resources are limited when an application requires intensive computation. At this level, students will either

Software component of PICit\textsuperscript{1.0} is developed to facilitate rapid small scale application development. PICit\textsuperscript{1.0} enables high-level languages (such as Visual Basic and Matlab) to be easily integrated with a microcontroller (BS2) via the RS232. This is the same connection that is used in programming BS2. The microcontroller serves as a bridge between hardware (sensors and actuators) and software (the PC platform). PICit\textsuperscript{1.0} has two programs running in connection: The firmware on BS2 and a device driver on the PC.

A simple firmware is developed on BS2 to communicate with the PC and allow some high level functions to be executed by sending commands over the serial port. On the PC side, a small function library is written to communicate with BS2. This library is written as an ActiveX control (a COM component); hence, it is quite portable within and among Windows based
platforms. The combination of the firmware on BS2 and the libraries on the PC are referred to as PICit\textsuperscript{1.0} software package.

After PICit\textsuperscript{1.0} is prepared, a programming platform has to be chosen through which the student will interact with BS2 via PICit\textsuperscript{1.0}. Even though high level programming environments such as VC++ or VB are popular, we have preferred Matlab and have written wrapper functions around this ActiveX control to enable Matlab users have access to the necessary BS2 utilities. Matlab is specifically preferred to introduce students with a software tool that will be useful for them throughout the rest of their engineering education.

It is intended to keep PICit\textsuperscript{1.0} as simple as possible in order to minimize students’ adaptation time to the software without compromising functionality. In order to do this, all of the laboratory experiments/exercises, and possible projects have been analyzed and a minimal set of functions are determined.

As mentioned in Table 1, BS2 has 16 bidirectional digital channels. Within PICit\textsuperscript{1.0} these pins are divided into two ports. First of port is devoted to output, and the second one is devoted to input. Use of only digital I/O imposes limitations on the variety of applications that can be implemented with BS2. Therefore, with added high-level functions, 3 RC servo motors can be controlled, position of a potentiometers and distance measurement from one IR range finder can be read by calling simple functions. On the output port, first 5 pins are devoted to be Digital Outputs, and the last 3 pins are setup to be configurable between a Digital Output and an RC servo driver. Similarly, on the input port, first 5 pins are devoted Digital Input pins. Sixth pin is either a Digital Input, or used in reading a potentiometer by measuring the discharge time of it over a capacitor. Seventh and the eight pins are either separate Digital Inputs, or these pins are used in reading range information from Sharp GP2D02 IR range finder. Configuration is easily carried out by a graphical user interface (GUI) as illustrated in Figure 1. In this GUI the schematics of the BS2 board is given to the students so that the mapping between the input / output ports and microcontroller pins can easily be established by the student.

![Figure 1. Configuration Interface of PICit\textsuperscript{1.0}](image-url)
In order to help students learn this software tool faster and also help in calibrating sensors a manual access GUI is also provided with PICit\textsuperscript{1.0} package as illustrated in Figure 2. In this specific illustration, the board is configured for 5 digital outputs, 3 RC servo controls, 5 digital inputs, one potentiometer input and one Sharp IR range finder input.

![Figure 2. PICit\textsuperscript{1.0} Manual Control GUI.](image)

All of these inputs and outputs are accessible by simple function calls within Matlab and Simulink. PICit\textsuperscript{1.0} introduction and software tools are available at the URL: www.me.metu.edu.tr/picit

Depending on the board configuration, update rates with PICit\textsuperscript{1.0} are between 10-30 Hz. This performance is not very challenging, however, the projects and labs are not performance demanding either. Main focus on project assignments is shifted towards integration issues so that at the worst case 10 updates per second on any input or output channel is satisfactory.

6. Conclusion

To facilitate teaching and promoting Mechatronics a sophomore level course is offered. Being a third semester course, rather than providing detailed low-level theoretical information, the students are involved in a semester project for which they work in groups. To keep the focus on the system integration aspect of the project, a software tool that is referred to as PICit\textsuperscript{1.0} is provided to the student.

PICit\textsuperscript{1.0} provides an easy to use high-level programming environment which is coupled with a microcontroller board, i.e. Basic Stamp 2 (BS2). A mixed-level teaching environment is created by using a high-level design and programming environment that is coupled with a microcontroller board.

Initial response from the students indicates that, PICit\textsuperscript{1.0} is proper for introductory Mechatronics courses as well as other mechanical engineering courses.
Sophomores being exposed to this mixed-level architecture, are encouraged to think about the details of this system. When these students become seniors and enroll in one of the two other Mechatronics courses offered by DoME, they are required to design a system that is more complex than PICit\textsuperscript{1.0}. This way, they do experience the details of the low-level design in conjunction with high-level programming environment.

**Acknowledgement:** Special thanks to Mr. Ali Emre Turgut (Ph.D. candidate in DoME, METU) for designing a BS2 prototyping board and letting ME220 students use these boards in class.

**Bibliography**

3. Hargrove J. B., Curriculum, Equipment And Student Projects Outcomes Or Mechatronics Education In The Core Mechanical Engineering Program At Kettering University, Mechatronics 12, 343-456 (2002).
8. Piguet Y., Mondada F., Siegwart R., New Environment For Learning By Doing In Mechatronics Education, 1st EURON Workshop on Robotics Education and Training, Weingarten, Germany, July 21
10. www.microchip.com

**Biography**

**Gökhan Bayar**

Gökhan Bayar is a Research Assistant of Mechanical Engineering at Middle East Technical University. He received his B.S. degree in Mechanical Engineering from Eskisehir Osmangazi University, Eskisehir, Turkey. His current research interests are Mobile Robots, Mechatronics, Robotics.

**Orhan Olcucuoglu**

Orhan Olcucuoglu is a Research Assistant of Mechanical Engineering at Middle East Technical University. He holds B.S. degrees in Mechanical Engineering and Industrial Engineering from Sakarya University, Sakarya, Turkey. His current research interests are humanoid robots and virtual reality. Orhan Olcucuoglu is a member of the Chamber of Mechanical Engineers of Turkey (TMMOB-MMO).
Bugra Koku, Ph.D.

Ahmet Bugra Koku is an Assistant Professor of Mechanical Engineering at Middle East Technical University. He received his Ph.D. degree in Electrical Engineering from Vanderbilt University, Nashville, Tennessee. He holds B.S. degrees in Mechanical Engineering and M.S. degrees in Systems and Control Engineering from Bogazici University, Istanbul, Turkey. His current research interests are memory organization of robots, swarm robotics, qualitative/topological navigation, low cost Mechatronics equipment design for robotic applications and robotics education. Dr. Koku is a member of the Institute of Electrical and Electronic Engineers (IEEE).

Ali Sekmen, Ph.D.

Ali Sekmen is an Assistant Professor of Computer Science at Tennessee State University. He received his Ph.D. degree in Electrical Engineering from Vanderbilt University, Nashville, Tennessee. He holds B.S. and M.S. degrees in Electrical and Electronics Engineering from Bilkent University, Ankara, Turkey. He has published over 40 research papers in robotics, intelligent systems, and signal processing. He was a member of Intelligent Robotics Laboratory of Vanderbilt University between 1997-2000. Previously, he was an Assistant Professor of Electrical and Computer Engineering at Tennessee State University. He has been involved in research projects including human-robot interaction, intelligent systems, mobile robots, humanoid robots, and component-based software systems development. Dr. Sekmen is a member of the Institute of Electrical and Electronic Engineers (IEEE).