AC 2011-342: DEVELOPING A COURSE AND LABORATORY FOR EMBEDDED CONTROL OF MECHATRONIC SYSTEMS

M. Moallem, Simon Fraser University

Prof. M. Moallem is with the School of Engineering Science, Simon Fraser University. He received the Ph.D. degree in electrical and computer engineering from Concordia University, Montreal, QC, Canada, in 1997. From 1997 to 1999, he was a Postdoctoral Fellow at Concordia University and a Research Fellow at Duke University, Durham, NC. He was with the Department of Electrical and Computer Engineering, University of Western Ontario, London, ON, Canada. His research interests include control applications including embedded systems, mechatronics, and renewable energy systems.

Yaser M. Roshan, Simon Fraser University

Yaser M. Roshan received the B.S. degree in electrical engineering from Ferdowsi University of Mashhad, Iran, in 2006, the M.S. degree in control engineering from Sharif University of Technology, Iran, in 2008, and is currently a Ph.D. student in Mechatronic Systems Engineering department of Simon Fraser University, BC, Ca.

Developing a Course and Laboratory for Embedded Control of Mechatronic Systems

Y. M. Roshan M. Moallem
Mechatronic Systems Engineering
School of Engineering Science
Simon Fraser University
Surrey, BC, V3T 0A3, Canada

Abstract

There has been a tremendous growth in the use of modern embedded computers in various applications in the past few years. Some aspects of embedded computer systems are covered in courses such as control systems, microprocessors, and circuits and systems. However, there exist few courses that integrate the above topics for designing embedded computer controlled systems. In this paper, we present an overview of laboratory testbeds for a course entitled “Embedded and Real-Time Control Systems” offered in our Mechatronics program. The objective of the course is to integrate concepts from previously taken courses such as programming, control systems, microcontrollers, and electronics. The laboratory component of the course is project oriented involving several low-cost mechatronic testbeds. The students go through the design of an embedded computer system using open-architecture mechatronic testbeds and integrated development environments. Furthermore, the students experience automatic C code generation techniques using high level code generation tools in the Matlab/Simulink environment which is further discussed in this paper.

1 Introduction

Embedded computer applications have experienced a rapid growth in the past few years [1,2]. Developing embedded computer applications requires multidisciplinary skills [3–6]. Recent progress in computer aided design tools enables the embedded systems professionals to design embedded systems in an efficient manner. Such tools help the students to apply more complex theoretical concepts in a practical system more easily. Examples include the CodeWarrior Integrated Development Environment (IDE) which is freely available from Freescale (www.freescale.com) for small applications, or high-level tools such as “Real-Time Workshop” and “Real-Time Workshop Embedded Coder” in the Matlab/Simulink environment. The latter tools can be integrated and utilized with the former one, which would facilitate implementation of more complex algorithms on a micro-controller. In the following sections we present further details related to the mechatronic systems used and utilization of the above tools in implementing embedded computer control applications.

2 Learning Objectives and Pedagogical Approach

Due to a growing need in industry for engineers who can perform software design and system integration for embedded applications, a new senior undergraduate/graduate course was recently introduced in the Mechatronic Systems Engineering program at Simon Fraser University. The main objective of the course is to introduce procedures for designing embedded computer-controlled systems by integrating concepts from sensors, actuators, dynamic systems, feedback control, and electro-mechanical hardware. The course has two major components. The lecture material includes topics such as the embedded systems design process; a review of instruction sets for microprocessor architectures; basic hardware and software platforms for embedded computing; multitasking systems for embedded applications; and practical issues related to computer based control systems such as PID tuning, anti-aliasing filters, integrator saturation and windup, and selection of sampling rates. Topics such as high-performance microprocessor architectures; high-level programming and its relationship to assembly code; optimization for power, speed, and memory size; and embedded systems networking are covered [7], followed by practical issues such as selection of sampling rates, discretization, and PID controller implementation [8].

The laboratory component of the course is project oriented. Students are divided into project teams with two to three students per team. Each team is expected to present their progress in class at different stages of the course. The students receive feedback in class from the instructor and students, while sharing their experiences with other groups. The students are expected to go through the embedded system design process and build their application using high-level languages such as C using integrated development environments such as CodeWarrior "Real-Time Workshop Toolbox" in the Matlab/Simulink environment. Several laboratory platforms are utilized in this course, from which the students can choose their projects. An air levitation system, a washing machine emulator, an industrial control trainer, a 1-DoF robotic arm with a magnetic gripper for pick and place operations, are some of these systems. All the above systems are open-architecture, and can be interfaced with any microcontroller or embedded computer system. While some of the platforms were purchased, some were (can be) developed in-house such as the air levitation system (Fig. 1), a motor pendulum system [9] (Fig. 2), and the 1-DoF robotic arm (Fig. 3). While most of these systems are simple and low-cost to build, they are complex enough to illustrate key concepts in development of embedded computer controlled systems.

In this paper, the 1-DoF robotic arm is presented as a simple in-house laboratory system for the course project. The system is depicted in Fig. 4 consisting of a DC motor, an encoder/potentiometer for sensing the position of the arm, and an HCS12 microcontroller

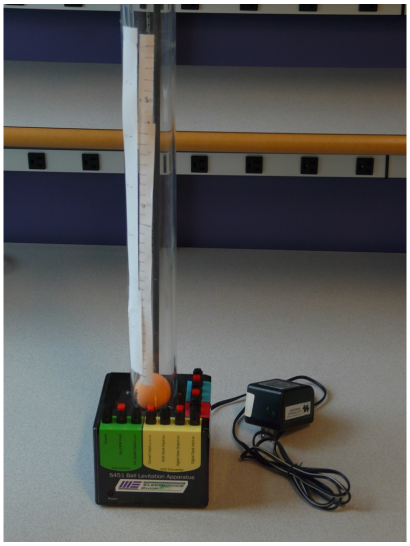

Figure 1: Air levitation system.

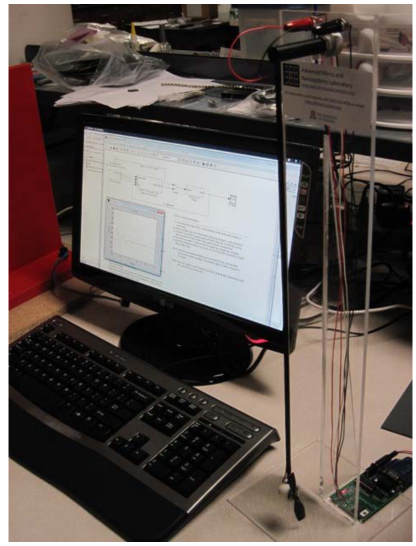

Figure 2: Motor Pendulum system[9].

development kit (DEMO9S12XDT512 from AXIOM Manufacturing). The microcontroller unit (MCU) MC9S12XDT512, from Freescale Semiconductor [10], is used as the embedded controller which provides analog and digital I/O, and pulse-width-modulated (PWM) signals. One of the objectives of the course is to familiarize the students with the capabilities of Real-Time Workshop and Real-Time Workshop Embedded Coder in the Matlab/Simulink environment with the C code generated automatically. To this end, proper Simulink blocksets for position sensing by an encoder and generating PWM signals have to be built as discussed in section 3. The students are required to design the feedback control system, such that the arm can move back and forth between two pre-defined points and pick/drop metal objects using a magnetic gripper. Figures 3 and 4 illustrate different hardware components of the system. In the following, the steps in using Simulink blocksets to generate C code is described.

3 C Code Generation with Matlab

The Matlab Real-time Workshop can be used for automatic code generation used in complex applications. Using such tools, Simulink models can be converted to C code and run on various microcontroller targets. Developing a host-target platform in Matlab/Simulink embedded coder requires two different steps [11]. In the first step, the target directory structure is built followed by appropriate target files, make files, and hook files. In the next step, the appropriate Target Language Compiler (TLC) files and S-functions of each block are created. TLC files are the script files that specify the format and content of output source files. S-functions are computer language description of a Simulink block written in Matlab, C, C++, or Fortran. C, C++, and Fortran S-functions

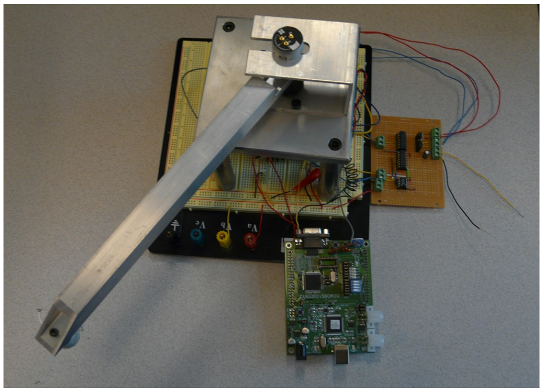

Figure 3: 1-DoF robotic arm with magnetic gripper for pick and place operations.

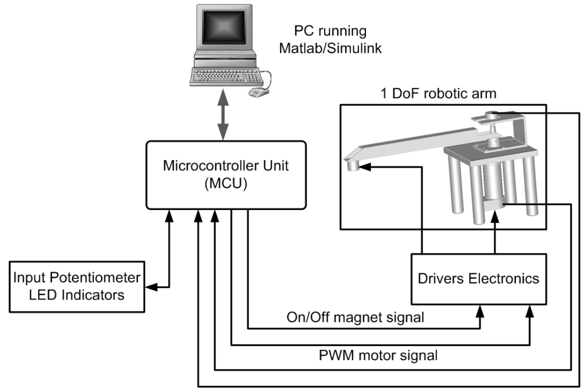

Figure 4: Components of the 1-DoF arm position control system.

are compiled as MEX files using the *mex* utility in Matlab. MEX-files (Matlab Executable files) are dynamically linked subroutines produced from C, C++, or Fortran source code that, when compiled, can be run from within Matlab in the same way as Matlab files or built-in functions are used.

4 Design of the Control System

A major task in designing modern embedded computer control systems is the software design. The designer should consider practical issues such as functionality, user interface, safety, reliability, power consumption, and cost. Various case studies in embedded system design are presented during the lectures that follow the software engineering design life cycle including requirements, specifications, architecture design, components design, and system integration. Other project management techniques such as top-down and bottom-up design, and use of software design diagrams such as classes, states, and sequence diagrams are discussed through the course. The first step in the design process of every embedded system is to define the set of requirements. A summary of requirements for our laboratory setup is indicated in Table 1. A typical class diagram is illustrated in Fig. 5. This diagram illustrates different components of the software architecture in the Unified Modeling Language (UML) format. The 1-DoF arm position controller class is comprised of the user interface, controller, and actuator software classes. The classes which represent hardware components are denoted by asterisks. In this diagram, the main module (position controller) is responsible for initializing other modules, providing communication between the modules, and running different tasks.

4.1 Implementation of PWM and Encoder Blocks for HCS12 in Matlab/Simulink

The code generation process in Matlab is illustrated in Fig. 6 [12]. The Simulink model (*model.mdl*) is converted into the Real-Time Workshop file (*model.rtw*), which has all the

Table 1: Requirements Form for 1-DoF Arm Position Controller

Name	1-DoF Arm Position Controller.
Purpose	Control the position of a 1-DoF arm according to a predefined path (e.g., between two set points for pick and place operations).
Inputs	Potentiometer or predefined values in Simulink environment as the set points of the source and destination points.
Outputs	- PWM drive signals applied to motor. - ON/OFF signals to the magnetic gripper on and off. - LED indicators: System status.
Functions	- Movement of the arm to desired locations. - Picking and placing metal objects. - Emergency shutdown.
Performance	Small regulation error, e.g., %5.
Cost, power, etc	...

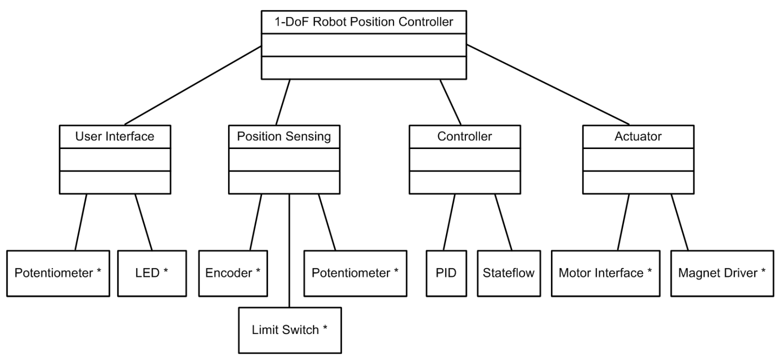

Figure 5: UML class diagram showing composition of subsystems.

information about the model. Then different TLC files will be used to generate the general structure of the C code and modify the code for each block (model-wide and block TLC files). Using TLC files and S-functions of the blocks, Matlab will create the source file (*model.c*) and also make-files (*model.mk*). These files will be used during the make process to build the executable final file (*model.exe*) to be run in the CodeWarrior environment.

Using the above procedure we implemented Simulink blocks for interfacing with incremental encoders. A quadrature encoder interface chip (LS7184) is used in the hardware part of the encoder interface system. The inputs of the IC are the encoder channels while the output is a signal which can specify the direction of rotation (1 and 0 for different directions). The S-function and the TLC files were used to define a new block called "encoder block" with its output equal to the difference between the values of

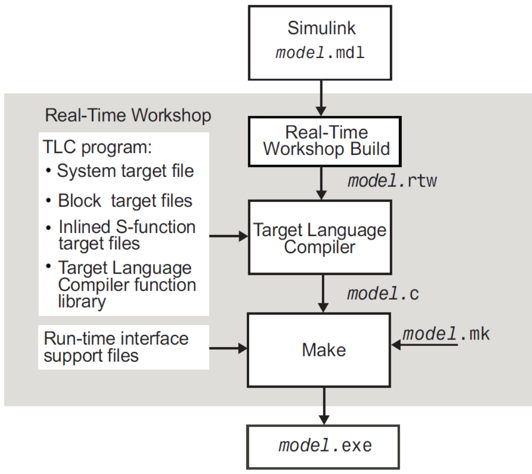

Figure 6: Code generation process in Matlab.

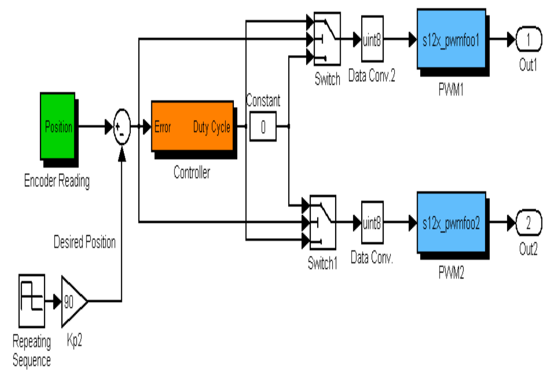

Figure 7: System modules in the Simulink environment.

the two pulse accumulator registers. In the S-function, the features of the block are defined and in the TLC function the appropriate timer registers are enabled.
The output of the control system (in Simulink model) is fed to a PWM block. PWM registers of the HCS12 microcontroller are programmed to drive the DC motor in the desired direction. A simple MOSFET bridge driver (TA7267BP, Full Bridge Driver H-Switch) is used to drive the DC Motor. Inputs of the chip are PWM signals for both directions while outputs are connected to DC motor.

Fig. 7 illustrates the controller designed in the Simulink environment. Based on the error value, the controller will send a proper duty cycle to the proper PWM block. The PWM block is chosen based on the sign of error and will determine the direction of rotation.

5 Conclusion

In this paper, laboratory platforms and a software environment to be used in an undergraduate/graduate course on embedded computer control was presented. The objective of the course is to integrate important concepts from courses such as microprocessors, control systems, and electronics. The proposed testbeds are low-cost and complex enough to allow the students to learn important concepts in designing embedded computer controlled systems. Using high-level tools such as the Real-Time Workshop Toolbox from MathWorks, the students get a hands-on experience in working with the state-of-the-art automatic code generation software, as well as developing their own software using embedded programming languages such as C. The multi-project and group oriented approach in performing the course project would allow a good learning experience in the design of embedded systems. At the time of writing this paper (Winter

2011), this course is currently offered for the first time to undergraduate students. More results will be presented at the conference regarding the learning experience by students and outcomes of the projects.

Acknowledgment

The authors would like to thank *Mathworks, Inc.* for supporting of this project.

References

1. C. C. Shih and L. J. Hwang, "Learning Embedded Software Design in an Open 3A Miltiuser Laboratory," *IEEE Transactions on Education*, in press, 2011.

2. M. Moallem, "A Laboratory Testbed for Embedded Computer Control," *IEEE Transactions on Education*, Vol. 47, No. 3, PP. 340-347, 2004.

3. S. Srivastava, V. Sukumar, P. S. Bhasin, and D. A. Kumar, "A Laboratory Testbed for Embedded Fuzzy Control," *IEEE Transactions on Education*, Vol. 54, No. 1, PP. 14-23, 2011.

4. T. Shirley, J. Wagner, R. Collins, and A. Gramopadhye, "A Mechatronics (and Material Handling Systems) Course-Classroom Topics, Laboratory Experiments, and Project," *ASEE Annual Conference and Exposition*, Austin, TX, 2009.

5. H. Y. Liu, W. J. Wang, and R. J,Wang, "A Course in Simulation and Demonstration of Humanoid Robot Motion," *IEEE Transactions on Education*, in press, 2011.

6. D. J. Jackson, and P. Caspi, "Embedded Systems Education: Future Directions, Initiatives, and Cooperations," ACM SIGBED Rev., Vol. 2, No. 4, PP. 103-105, 2006.

7. W. Wolf, *Computers as Components: Principles of Embedded Computing System Design*, San Francisco, CA: Morgan Kaufmann, 2001.

8. B. Wittenmark, K. J. Astrom, and K. E. Arzen, *Computer Control: An Overview*, Department of Automatic Control, Lund Institute of Technology, Sweden, 2009. Available online: www.control.lth.se/kursdr/ifac.pdf (accessed January 2011).

9. E. T. Enikov, V. Polyzoef, and J. Gill, "Low-Cost Take-Home Experiment on Classical Control Using Matlab/Simulink Real-Time Windows Target," *Proceedings of the 2010 American Society for Engineering Education Zone IV Conference*, PP. 322-330, 2010.

10. *MC9S12XD Family*, Product Brief, Rev. 2.14, Freescale Semiconductor, 2005.

11. *Real-Time Workshop 7*, User's Guide, MathWorks, 2010 (www.mathworks.com).

12. *Real-Time Workshop 7*, Target Language Compiler, MathWorks, 2010 (www.mathworks.com).