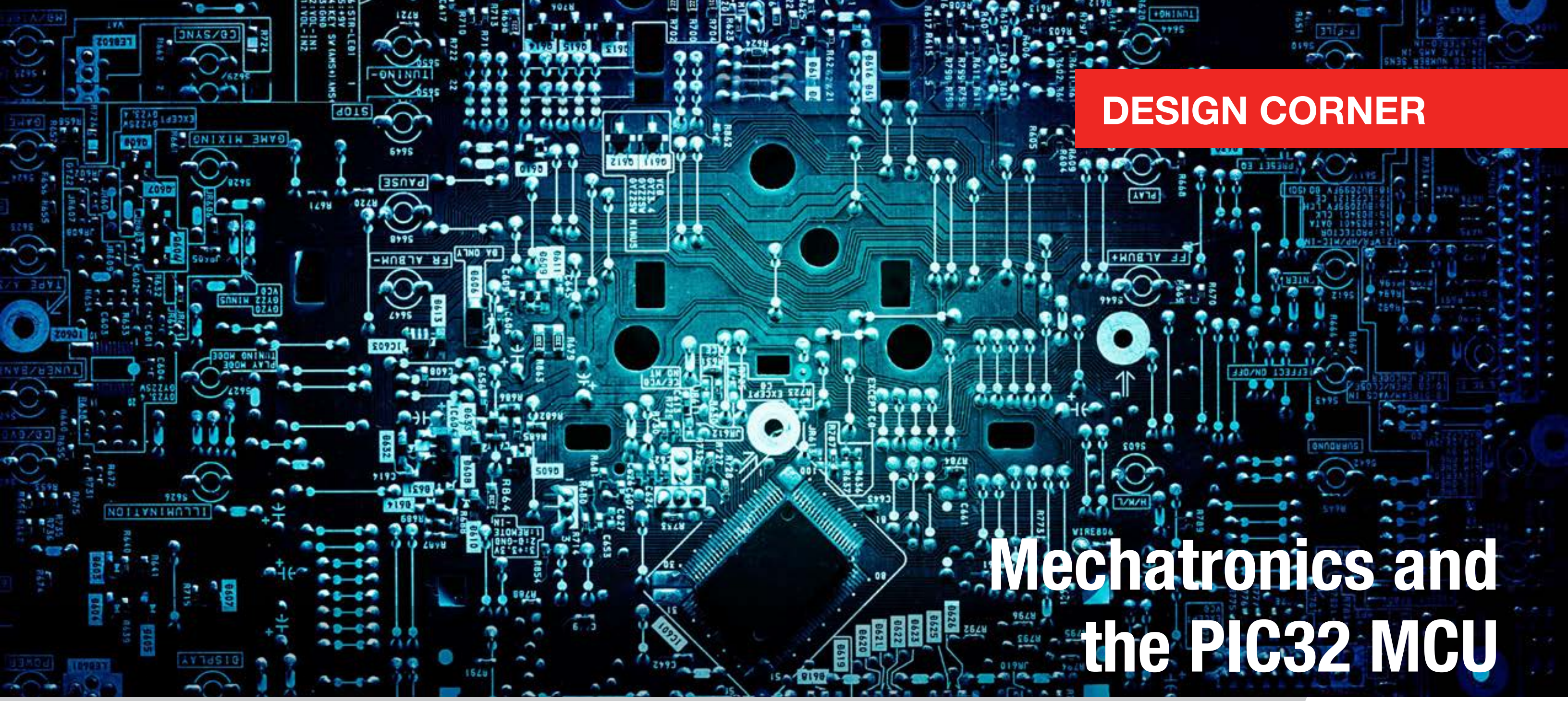


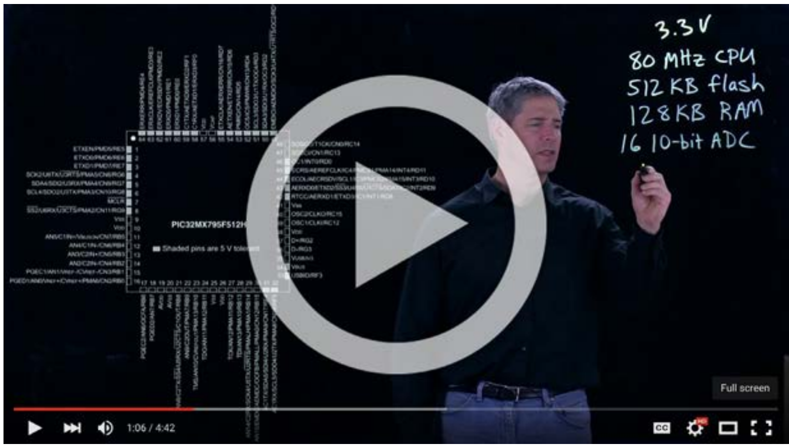
DESIGN CORNER

Mechatronics and the PIC32 MCU

Embedded Computing at Northwestern University

In 1997, Kevin Lynch, currently Professor and Chair of the Mechanical Engineering Department, joined Northwestern University. At that time, the annual Robot Design Competition (DC) was already a popular event with undergraduates. Teams of students build autonomous mobile robots to compete on obstacle courses that change every year. Despite the popularity of this competition, Northwestern did not offer a course on mechatronics, the study of microprocessor-controlled electromechanical systems incorporating sensors and actuators. After obtaining funding and support from the University, Lynch developed a new lab and a new elective course—ME 333 Introduction to Mechatronics—that was first offered in 1999. His inspiration came from his training in robotics, as well as his love for a course he had taken at Princeton many years earlier, MAE 412 Microprocessors for Measurement and Control (taught by Professor Michael Littman), affectionately known as "Trains" for its use of model railroads as demonstrators.

He structured ME 333 to spend the first six weeks covering the electronics of interfacing sensors and actuators with a computer, plus rudimentary feedback control, accompanied by traditional three-hour labs. The last five weeks were devoted to a final project of the students' own design. Students worked in teams of three on both the labs and the final project, and the course culminated with a fair where teams demonstrated their projects in the lobby of Northwestern's Technological Institute. This successful course, which grew in popularity each year, followed this format for about ten years. The computer or microcontroller used by the students was occasionally updated, but otherwise the format and content of the course changed little.



Video lectures were created for ME 333 Introduction to Mechatronics.

Lynch was dissatisfied, though, with the compartmentalization of effort by students when working on the final project. "Typically one member of the team would do the machining and mechanical prototyping, one member would do the electronics, and one member would do the programming," said Lynch. Believing that all mechatronics engineers should be competent in embedded computing, he was troubled that nearly two-thirds of the students completed the course without having a significant experience in programming.

To adress this issue, Lynch coordinated with his colleague, Professor Michael Peshkin, who created two new courses. The

(continued on page 30)

first course, ME 233 Electronics Design, replaced an existing required course and puts an emphasis on practical electronic design. The second course, ME 433 Advanced Mechatronics, has teams of two or three students work on a course-long project. The addition of ME 233 as a prerequisite for ME 333 allowed Lynch to reduce the electronics content in ME 333, and the addition of ME 433 allowed him to drop the team project. Students interested in pursuing mechatronics further after taking ME 333 are now able to undertake a more significant project in ME 433.

A New Focus

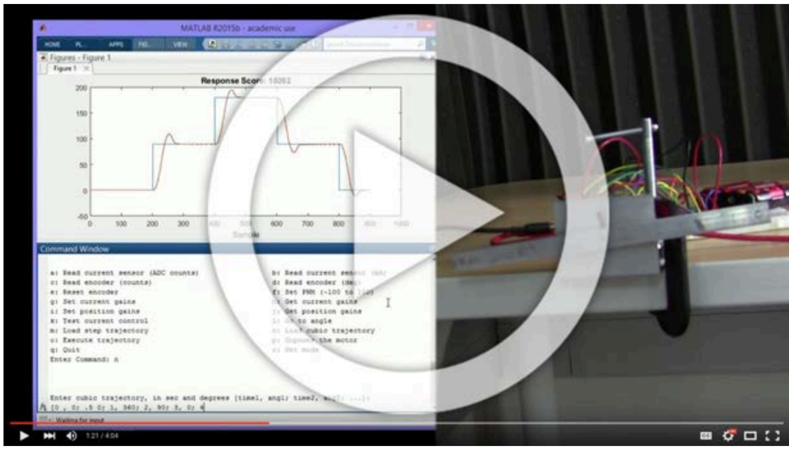
Taking advantage of this new flexibility, Lynch reorganized ME 333 to focus on embedded computing and control using a PIC32 microcontroller, specifically the **PIC32MX795F512H**. Each student individually completes all assignments, including a significant final project. In this final project, students implement the electronics and software for a professional nested-loop brushed DC motor control system, where the outer-loop motion controller, running at hundreds of Hz, uses a motor reference trajectory and encoder feedback to generate a commanded torque (equivalently, motor current), and the inner-loop current controller, running at 5 kHz, controls the Pulse-Width Modulation (PWM) signal to an H-bridge to achieve the commanded motor current. Sensors for the project include a motor current sensor and a motor encoder, and the software uses multiple interrupts and peripherals such as analog input, digital I/O, timers, output compare, and SPI and UART communication. Optional extensions save data to Flash memory and use the Parallel Master Port (PMP) to drive an LCD display.

Students also create a MATLAB® interface to the PIC32 MCU via communication over a USB cable. The interface provides options that allow the user to set feedback gains, test the current-control loop in isolation, specify the coefficients of a motor trajectory, and plot the results of the controller's trajectory-tracking performance. Knowing that commercial motor amplifiers with these kinds of capabilities cost hundreds of dollars is good motivation for the students.

Students develop their PIC32 software from scratch in C, based on concepts learned earlier in the course. Many ME 333 students enter the course with no prior experience in C.

Selecting a Microcontroller

Lynch's approach to teaching embedded control is geared toward providing students with a foundation based on the lowest-level building blocks they will likely need for professional integration of microcontrollers with sensors and actuators. For example, detailed coverage of the design of microprocessors is too low-level for most mechatronics engineers. On the other hand, use of prepackaged Arduino® like libraries is too high level to be the foundation for professional embedded computing. While Arduino hardware and software are useful for quickly getting started with microcontrollers, the software abstractions keep the budding mechatronics engineer at an arm's length from important concepts in embedded computing. "Since ME 333 is taken by MS and PhD students specializing in robotics, in addition to undergraduates, it is important for this course to provide them with a professional foundation," Lynch said.



Students create a MATLAB® interface to the PIC32 MCU.

The PIC32 MCU was selected because it has a modern architecture, is well priced, has a plethora of peripherals and good performance (80 MHz, 512 KB Flash, 128 KB RAM), and has good market penetration, making it an appropriate first microcontroller for the future professional.

The "No Magic Steps" Philosophy

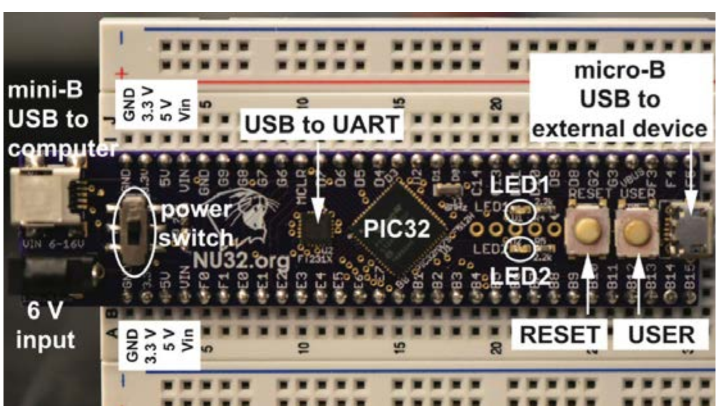
According to Lynch's "no magic steps" pedagogical philosophy, once the lowest-level building blocks have been determined, there should be no missing or "magic" steps that mystify the student. In the case of ME 333 as a course, it meant that there should be no software abstractions to prevent a clear understanding of how the software connects to the hardware. Once the students understand how their low-level code interacts with peripherals, then abstractions in the form of software libraries—such as Microchip's **MPLAB® Harmony** software framework for use with PIC32 MCUs—can be introduced. This philosophy impacted these aspects of the course:

(continued on page 31)

Choice of Programming Language – Assembly programming fits the "no magic steps" criterion, but it is processor-dependent. C was selected for ME 333 because it is processor-independent, while still remaining relatively low level. Students read and write the peripherals' special function registers directly, reinforcing the documentation in the reference manual and data sheet. They do not start with the MPLAB Harmony software framework (although it is introduced in the textbook), because for the beginner, the abstractions that MPLAB Harmony introduces obscure the connection between the software and the hardware.

Command Line Tools vs. MPLAB X IDE – Students in ME 333 use text editors to create their programs, and they compile and load their programs at the command line using a Makefile. Following the "no magic steps" philosophy again, they do not use MPLAB X IDE. The Makefile clearly demonstrates how a program is built from the source text files, while the role of the IDE in the build process can be mysterious to the student.

The NU32 Development Board – This custom, 60-pin, breadboardable, DIP-style development board uses the PIC32MX795F512H. Its primary purpose is to break out the pins of the PIC32 MCU. It also contains voltage regulators to regulate an input voltage to 3.3V and 5V; an FTDI FT231X chip that converts USB to UART communication, allowing the host computer's USB port to communicate with a UART on the PIC32 MCU; an on-board 8 MHz resonator that is used to generate an 80 MHz CPU clock; two LEDs and two pushbuttons; and USB connectors to connect to the host computer and an external device, like a smartphone. The PIC32 device has an on-board bootloader, allowing it to be programmed via the host computer using just a USB cable. The bare-bones board is in keeping with the "no magic steps" philosophy: users must construct their own external circuits to interface with the PIC32 MCU, as they would in a professional setting. It also makes the board small and inexpensive enough to be used as a component in many embedded projects.



The NU32 Development Board features a PIC32MX795F512H MCU.

The nScope portable function generator and oscilloscope.

A Different Learning Format

In the original ME 333 course, students participated in traditional, scheduled, three-hour labs in a facility with function generators, oscilloscopes, electronic components and other tools. In the new ME 333, students use a portable lab kit to do their sensor and motor integration lab exercises at any time and place. Each lab kit consists of a multimeter, a breadboard, a power supply, a wire stripper, wire and various other electronic components, and the inexpensive and portable **nScope** oscilloscope and function generator. The nScope uses the student's laptop as its display. It was developed at Northwestern based on the PIC24FJ MCU and, after a successful Kickstarter campaign in June 2015 that generated over 1700 backers, it is now available to anyone.

Taking advantage of these portable labs, ME 333 uses a flipped classroom format. Videos of lectures were created using the Northwestern Lightboard, an ingenious and easy-to-use device created by Professor Peshkin, and are available on a **YouTube video channel**. This allows students to work on projects during class time while the instructor and TAs circle the room, offering help. Students bring their laptops and lab kits to every class. Brief lectures are interspersed to reinforce the videos and to elicit questions and discussion. "Time spent with students in this format adds more value than repeating a lecture from the previous year," Lynch said.

A New Textbook

In December 2015, Newnes released ***Embedded Computing and Mechatronics with the PIC32 Microcontroller***, written by Lynch and his co-authors Nick Marchuk, Lynch's co-instructor in mechatronics and Matt Elwin, a PhD student. Based on—and

(continued on page 32)

expanding beyond—the curriculum in ME 333, approximately half the book covers the hardware and software of the PIC32 microcontroller, as well as an introduction to programming in C. It provides a systematic explanation of the build process and how the software you write, plus the software Microchip provides, turns into a final executable that connects to the hardware described in the PIC32 MCU's data sheet. A peripheral reference describes in detail the operation of most of the peripherals, including sample code and projects using them. To provide a logical and coherent presentation, the book brings together key material from the PIC32 Reference Manual, the PIC32MX5xx/6xx/7xx Data Sheet, the XC32 C/C++ Compiler User's Guide, and MPLAB Harmony documentation.

The other half of the book covers fundamental topics in mechatronics, such as a circuits review, common sensors, digital signal processing and feedback control with the PIC32, brushed DC motors, brushless DC motors, gearing and motor sizing, and other actuators like stepper motors, RC servos, solenoids, and voice coil actuators. While the book was written as a university textbook, it is also appropriate for serious hobbyists and as a reference for professionals.

Visit the **NU32 website** for more information about the book including free downloads of sample chapters, sample code, and videos supporting the book. You'll also find details about the NU32 development board, the nScope portable oscilloscope, and the Lightboard.

Kevin Lynch is Professor and Chair of the Mechanical Engineering Department at Northwestern University, where he teaches courses in mechatronics and robotics. Before joining Northwestern, he received a BSE in Electrical Engineering from Princeton and a PhD in Robotics from Carnegie Mellon University. He has received Northwestern's Engineering Teacher of the Year award and the Charles Deering McCormick Professorship of Teaching Excellence, the top teaching award conferred by Northwestern. He publishes and lectures widely on his research in robotics, and he is co-author of "Principles of Robot Motion" (MIT Press, 2005). He is a Fellow of the IEEE.