AC 2009-558: KEEPING IT SIMPLE: AN INTRODUCTORY MICROCONTROLLER COURSE USING THE HCS08

matt gallagher, Vermont Technical College

Andre St. Denis, Vermont Technical College

John Murphy, Vermont Technical College

Keeping it Simple: An Introductory Microcontroller Course using the HCS08

Abstract

The current trend in microcontroller courses is to use demonstration circuit boards with many built-in devices that showcase the input-output capabilities of a state-of-the-art device. Current microcontrollers have fifty pins or more, come in surface mountable packages and have external clock sources. The combination of many interesting peripherals and a complex processor appear to serve the immediate needs of the introductory microprocessor course. Students who chose to use a microcontroller in a future project were forced to design circuit boards requiring fabrication and assembly at outside vendors at considerable cost; this tended to discourage their use in our capstone projects course. We discuss our current introductory course that uses a MC9S08QG8/4 microcontroller from Freescale. It comes in a sixteen pin DIP package, has an internal clock and can be programmed using the same Codewarrior software IDE Freescale uses for all of its microcontrollers and can be programmed through a USBSPYDER08™ interface. We developed a set of four simple input/output boards so students can exercise their ability to interface with pushbuttons, LCD displays, seven segment displays, keypads and the serial port among other things. As with more complex boards, they gain experience developing firmware in both assembler and C languages and learn all the basic attributes of a microprocessor. We will show how students were more likely to use these microcontrollers in their capstone projects compared to previous years where more complex controllers were used. Using simpler, applications oriented boards and more lab friendly MPU devices we met our goal of having students actually want to use these devices.

Introduction

Microprocessor/microcontroller courses are standard components of both two year and four year engineering programs due to their widespread use in everything from automobiles to remotely situated data loggers. A balance needs to be struck in offering these courses: basic concepts must be conveyed along with skills that will allow students to be productive in future situations where they will use these devices. Basic concepts such as interrupts, memory maps and parallel ports can be taught in a classroom with accompanying text and laboratory. Many standard demonstration boards exist that can serve as a vehicle for showcasing these concepts in the lab[1]. Many courses have been developed with these types of boards[2,3,4]. What can be lacking in such an approach is giving the ability to use these devices practically. In addition, there tends to be so much capability in state-of-the-art microprocessors devices (MPUs) and so many options in multipurpose trainers that students are overwhelmed by them and don't realize that they can simplify and use a subset of functionality in their designs.

At Vermont Technical College we give our students freedom to choose projects in their respective capstone courses. Naturally, we want them to draw from their academic experiences and implement as much complexity as we feel they are capable of handling. Projects employing MPUs almost always satisfy this desire since they consist of software/firmware code development, circuit debugging, board layout and fabrication skills. In the past we have used commercially available demonstration kits and found it difficult to get students to use MPUs in their projects because they felt they were too complex. There was no option of using SOP packages with 40 pins or more since we do not have surface mount fabrication capability. Also, it was hard to justify a 40 pin device to use in a project that required only a handful of I/O ports.

We had been using a demonstration board constructed by Technological Arts which featured a HC12 MPU, MC9S12C32[1]. Although this device satisfactorily met the basic goals of the class we felt it was not adequate for the reasons given above. We were also concerned that it was approaching the end of its life cycle and noticed that the costs were escalating. Our search for another board led us to the HCS08. We took advantage of experience gained in a Control Systems course being taught in our BS Electromechanical Engineering Program that used Freescale MC9S08QG8 MPU. The MC9S08QG8 chip is readily programmed through an inexpensive USBSPYDER08™ module and comes in a DIP package for ease of use[5]. Review of the capabilities of it indicated that it had all the features we needed to cover in our introductory microprocessor class. It has a convenient package, limited number of pins and a low cost that made it a good candidate to implement in our class. Another advantage which we did not initially consider was the internal clock on the MC9S08QG8 that required no external circuitry. This greatly simplifies any application where timing is not critical.

We will show that this choice of device and the exposure to several application specific boards resulted in a greater number of students opting to use an MPU in their senior capstone class for the two-year AS in electrical engineering technology (ASEET) degree. Using an MPU immediately expands the functionality and capability of projects compared with others that that consist only of discrete logic gates, transistors and other simpler devices and frequently reduces the number of ICs and size of the final circuit. Students gain an appreciation of embedded processor control, increase their own confidence and make better, more useful engineering technicians and engineers.

Introductory Microprocessor Course at VTC

At Vermont Technical College, our introductory microprocessor class serves majors in ASEET, BS in Electromechanical Engineering Technology (BSELM) and both AS and BS majors in Computer Engineering Technology. Students majoring in the BS in Computer Engineering Technology continue on with a second course in microprocessors. The introductory MPU course is a first semester, second year class for all of the disciplines. Prerequisites include introductory courses in digital circuits and C programming. We begin programming the MPU with assembler language but migrate to C roughly halfway through the course. Our overall approach is not markedly different from introductory courses at other institutions.

We would like to give our students as much confidence as we can for them to undertake reasonably complex and modern circuit projects in their capstone classes. We balance theoretical, classroom education with hands-on applications that gives them laboratory skills. To achieve this confidence, we require each student to own the hardware and assemble it. Soldering the boards does take some time (especially the first one), but the students get better at it and get a lot of satisfaction out of the process. Motivated students can do a lot of lab work and experimentation at home, after hours.

Our program outcome goals for the class are shown below. Students are assessed each year for the purposes of ABET accreditation:

- Be able to write programs in assembly language and in an appropriate high level language.

- Be able to utilize current hardware and software tools and devices such as laboratory instruments, MultiSim, embedded controllers and PLCs.

Our objectives for this the introductory MPU class is are summarized in the table below.

Topic	**Time devoted (hrs)**
Review of digital concepts	2
MPU architecture	3
Assembly programming	2
Memory map	2
Parallel Ports	3
Stack and subroutines	3
Interrupts	3
C programming	3
Timer module	2
Analog to digital conversion	2
Liquid crystal display	2
Keypad	2
Low power modes	2
Seven segment display	3
Pulse width modulation	1
Quizzes and exams	6

Table 1. Course objectives for Vermont Technical College introductory microprocessor class

Choosing a microcontroller

History: our Controls Systems Class

The choice of which MPU device to use in our MPU course was in part informed by our experience with our control systems courses where MPUs are also employed. Historically, controls systems has used HCS08 based MPUs. Their use of these devices was limited to being users and they were not required to write a lot of code or develop their own applications.

Nonetheless, the low cost, lower complexity made the HCS08 ideal controllers for DC motors. We used the USBSPYDER08™ in the controls systems classes so this was a natural path for us to follow for our MPU classes. This is the main reason we did not pursue the similar PIC controllers made by Microchip[6].

History: our Introductory Microprocessor class

In our microprocessor course, we had been using a demo board M68DKIT912C32DM which uses a 32 pin module with an HC12-based MC9S12C32 MPU since 2005 [1]. The entire docking station and module cost about $60. The module which sat on the docking board consisted of a crystal oscillator, a switch to toggle between stand-alone and debug mode and several passive components. The cost of the module alone in the event of damage was $30. Burned out chips required the purchase of replacement modules. For a text we used a softcopy of Thomas Almy's HC12 text book[7]. The M68DKIT912C32DM did not have several standard I/O capabilities so an additional board had to be designed and fabricated in-house. Included on the board was a seven segment display, LCD, speaker and a potentiometer for analog input. It connected to the demonstration board via a 40 pin ribbon cable. They were ordered from a vendor and students were required to assemble. A 40 pin ribbon cable connected the two boards. As this board was specific to the M68DKIT912C32DM kit we had to migrate to another solution for our Fall 08 semester.

How we chose the HC08 and changed our notion of a demonstration board

When reviewing the various options for replacing the M68DKIT912C32DM we came upon several extremely well thought out demonstration boards. They employed moderately complex HC12 or other MPUs, had on-board displays, LEDs, keypads, and even some new features like accelerometers. These would have been a natural migration path given the board we had been using but we felt our students would still be lacking the confidence to use these more complex processors in future projects. While their demonstrated abilities on exams and labs in the MPU class was acceptable, when it came time to use these devices in their capstone class or when asked if they liked working with microcontrollers they cited the complexity and difficulty of using them. Running cables or use the MPUs in a way that implemented the whole demonstration board, not just the chip, was unsatisfying for them.

We began to explore the experiences gained in our control systems classes and entertained the idea of using the HCS08 MPUs and a set of simpler demonstration boards. The prospect of using these MPUs had some distinct advantages already cited. An advantage which we did not fully appreciate beforehand was the internal clock source for the HCS08 device. For applications where timing accuracy <10% was not required the internal 4MHz clock was sufficient. Without requiring an external clock, the MPU circuitry was reduced to the chip and a 3V power supply. This allows using these ICs and "universal replacements" or glue logic in other circuits (and courses). A disadvantage of using DIP devices with less than 20 pin is that there are few demonstration boards built for them and they are limited in their I/O abilities[8]. Also, often they are soldered directly to their boards making replacement difficult.

Design of Input / Output Boards

A key change in philosophy was to design and use a set of four I/O boards. One was modeled in the typical way, with various options which could be enabled or ignored by the student. The others were intended to be more task specific.   Taken together, they represent several different embedded solutions using small MPUs, rather than one multipurpose, generalized training board that students found to be cumbersome to use later in a practical way. Since the HC08 MPUs we chose all support IIC, all these board can be configured and used in ad-hoc systems, demonstrating a modular approach to design.

- The "LCD/Encoder" board shown in Figure 1 uses the 16 pin version of the MPU and is the closest to the multipurpose trainer model. It includes the interface circuitry to operate either an industry standard HD44780-based LCD module or a bank of LEDs and a set of pushbuttons or a rotary encoder. This board also includes the crystal oscillator circuitry (if needed), an on-board potentiometer, provision for a slotted optical switch, and a small prototyping area. This board was designed with the intent of programming it in lab as a security/fire alarm controller, but it can be useful in a multitude of practical applications which require a few buttons and a small display (e.g., digital setback thermostat, operator's station for a simple machine …). Additionally, the IIC port is an option which allows this board to serve as a display subsystem with one or more other boards. A lab has been suggested to build a network of simple voting machines using all the students' boards working together in one lab.

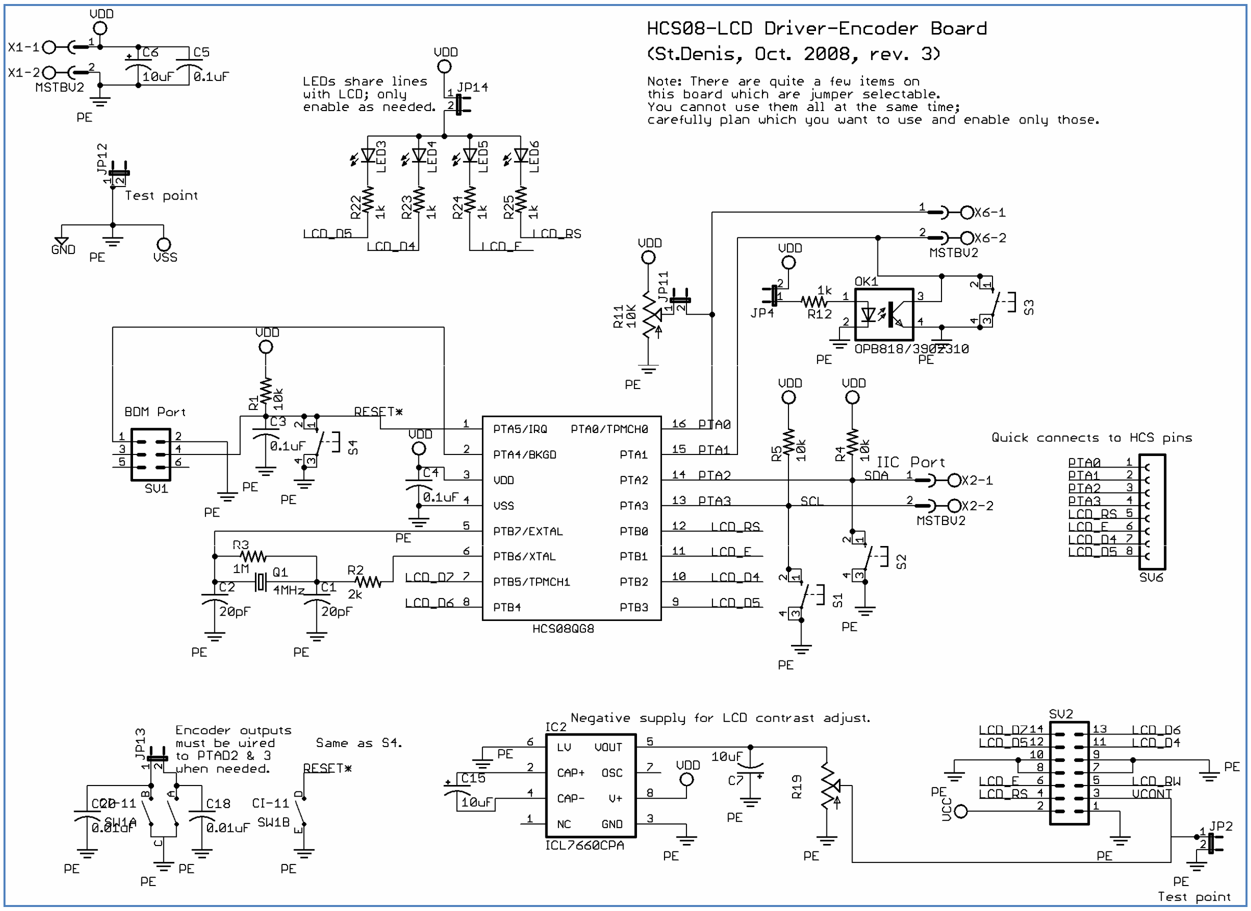

Figure 1. Schematic of the LCD trainer board.

- The "7 segment display" board shown in Figure 2 also uses the 16 pin version of the MPU and drives a dual digit 7 segment display. Two pushbuttons are included and provisions for cascading these boards into larger displays (using a 'ripple out/ ripple in approach or the IIC bus). This allows us to discuss multiplexing strategies in our class and implement them in software without 'explaining away' the other output devices typically found on multipurpose boards.

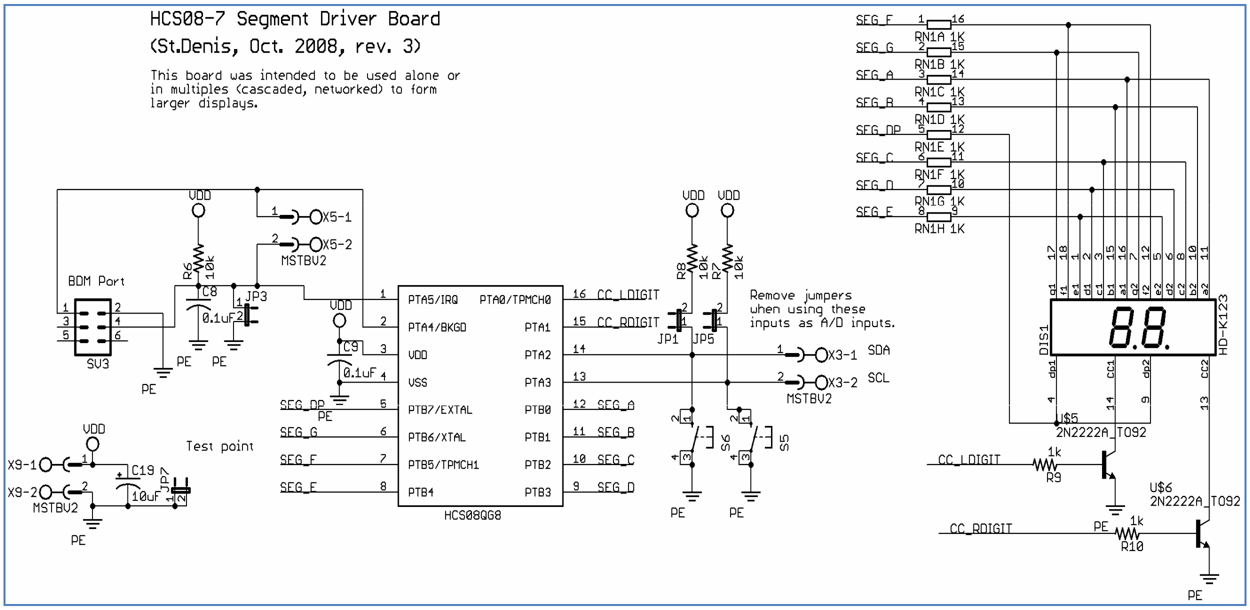

Figure 2. Schematic of the seven segment display board.

- The “Accelerometer” board in Figure 3 uses the 8 pin version of the MPU an is physically the smallest (1.4” x 1.6”). It includes a surface mount 3-axis accelerometer (MMA7260), 2 LED indicators, the IIC port and provision for a single pushbutton. This board has been used to create a simple ‘tilt indicator’ (using the LEDs) and also demonstrate a “smart sensor” subsystem by using the IIC connection to the LCD board and creating a “digital level”.

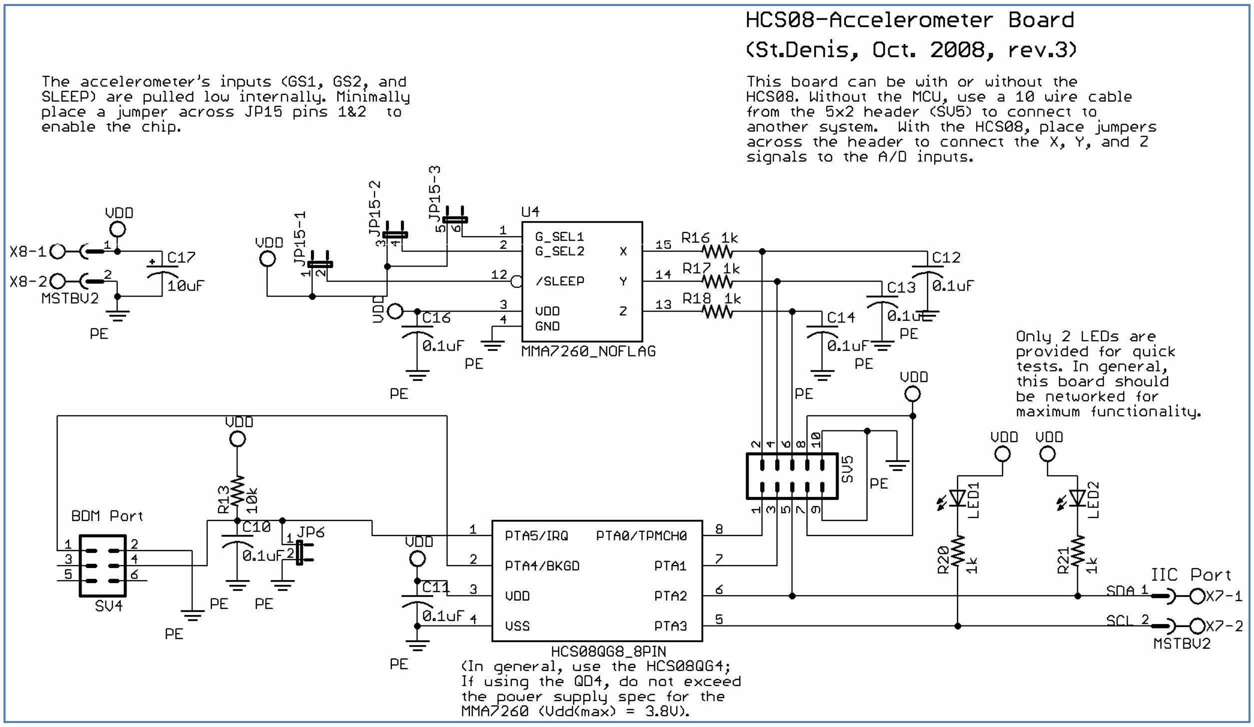

Figure 3. Schematic of the accelerometer board.

- The "Motor Controller" board in Figure 4 was a design borrowed from the Controls Systems class and uses the 16 pin version of the MPU. This board includes a switching regulator, 4 pushbuttons, 4 MOSFET transistors (with LED indicators), a MAX232 based RS-232 port, and a 16 pin "expansion" header. This board was used to implement a stepper motor controller and interface a keypad (via the expansion header).

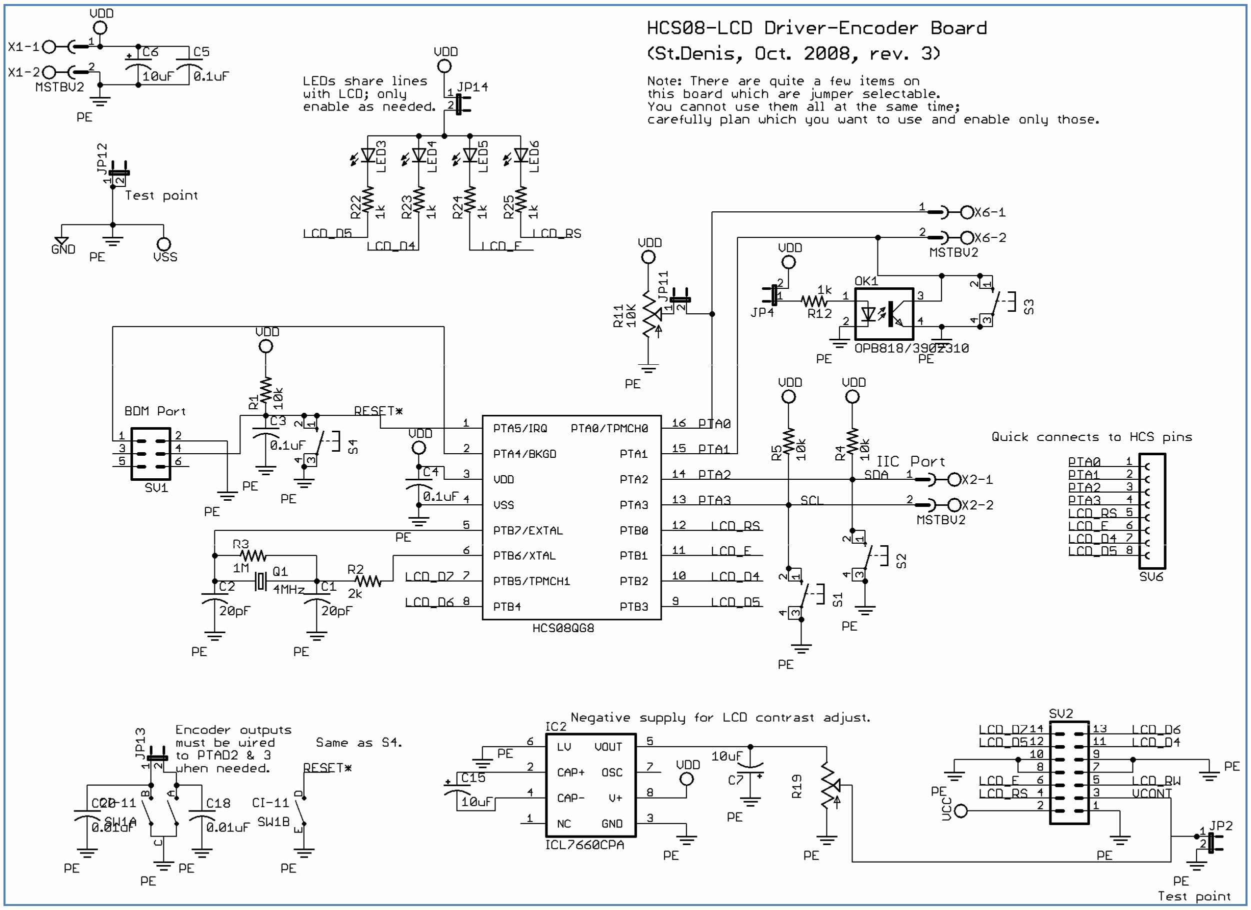

Figure 4. Schematic of the motor control board

All the boards were designed using similar components:

- All use several (screw) terminal blocks for power leads and key external I/O connections.
- All have the standard 6 pin BDM (Background Debugger Module) port to allow in-circuit programming with the Spyder ™ programming tool.
- All use standard single and dual row (100 mil spacing) headers.
- All have sockets for quick replacement of burned out ICs.
- All can be battery operated from a dual AA battery pack (given to the students).

All these simple and common features contribute to easing the students' fear and making it easy to convince students to work with all of them in a single semester (and beyond).

The first 3 boards were designed as a set and are actually one large pc board (approximately 3” x 4”); before giving them to the students, we cut them apart. The PCB layout is shown in Figure 5. This has help keep the cost down and we have found that the total cost of these 4 simpler boards plus the Spyder programmer is the same as the older, bulkier HC12 kit plus the add on I/O board.

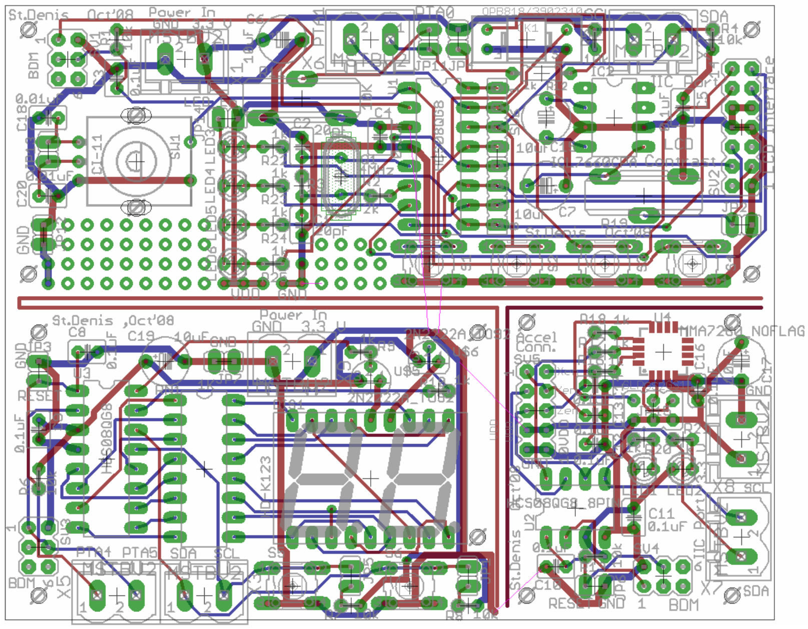

Figure 5. PCB layout of the LCD, seven segment display and accelerometer boards. Fabricated together, then split apart.

Course and lab schedule

After adopting the HC08 we had to modify both labs and lectures to reflect the host of differences between the MPU types; both the MPU and the I/O boards were different compared with the HC12 demo board. Laboratory exercises were particularly in need of revision. We maintain our overall course objectives while taking advantage of some of the additional I/O features we have on our homemade boards. Even though many of the labs are introductions to various topics, the ability to apply the software to the application specific boards has changed the nature of the lab experience a bit and made the use of these MPUs seem more reasonable to our students.

We chose a textbook "HCS08 Unleashed: Designer's Guide To the HCS08 Microcontrollers" by Fabio Pereira to accompany the labs and lectures[9]. We also distribute Freescale data sheets for the MPU[10].

Week	HC12 Lecture	HC12 Lab	HC08 Course Lecture	HC08 Lab
1	Course introduction	Code Warrior/file management	Course introduction	Code Warrior/file management
2	MC9S12C32 architecture, memory map	Using the M68DKIT912C32DM kit	MC9S08QG8 architecture, memory map	Using the Spyder kit
3	CPU instruction types	Push buttons flow charts	CPU instruction types	Construct I/O board
4	GPIO basics, logic commands	Construct I/O board	GPIO basics, logic commands	Push buttons flow charts
5	Branching logic and using LEDs	Exercise I/O board	Branching logic and using LEDs	Instruction timing, LEDs
6	Stack and subroutine mechanics	Instruction timing	Stack and subroutine mechanics	Seven Segment display
7	Exam, Seven segment display	Stack and Subroutine	Interrupt mechanics	IRQ and RTI interupt
8	Seven segment display and timing analysis	Switches and seven segment display	Exam, timing analysis	Controlling a stepper motor
9	Using the LCD	Counter and seven segment display	Keypad, intro to C language	Using the Keypad (assembler)
10	Interrupt mechanics	Using the LCD	Keypad redux in C	Keypad in C
11	Timer module	RTI and IRQ interrupts	Timer and PWM	Analog to digital converter /Accelermoter
12	Using the PWM	RTI clock and LCD	A to D converter	PWM and A to D
13	Using the A to D, more PWM	Timer and Pulse measurement	Using the LCD	Timer and LCD
14	Review	PWM and A to D	Review	Low power modes

Table 2. Summary of course versions using the HC12 and HC08.

Results

The introductory microprocessor class is offered in the fall semester of the second year in our two year ASEE program. The capstone course is offered in the following semester when their knowledge of MPUs is still fresh. Students are free to choose projects on their own and are given suggestions if they want them. We consistently encourage students to use MPUs in their projects as they can quickly get reasonably complex circuits and they invariably use the knowledge they have been taught. Table 3 summarizes the numbers of students who have opted to use MPU-based projects in their capstone over the last three cycles. A separate row is details how many students were successful in their attempt at incorporating an MPU into their project. Although students may have undertaken MPU-based projects in previous years often they were not successful; either requiring too much external support or not functioning at all. The results suggest that students do embrace the HC08 approach we have taken.

	Spring 2007	**Spring 2008**	**Spring 2009***
No. of students	18	20	17
Fraction of students using MPU-based projects	39%	35%	80%
Fraction of students using MPU-based projects successfully	22%	15%	*

Table 3. Summary of students undertaking MPU-based projects in their capstone class. *Results are pending on success rate.

While these results are encouraging there is not a corresponding increase in student's ability to use the MPU devices. We compile data each year on how well students meet the program objectives listed earlier. The migration from the HC12 to the HC08 has not appreciably improved student's ability in these areas as shown in Tables 4 and 5.

	Fall 2006	Fall 2007	Fall2008
Uses flow chart or other descriptive aid	86	80	69
Correctness / Lack of logical errors	86	80	69
Assembler Syntax	71	90	88

Table 4. Percentage of student meeting the program outcome: Be able to write programs in assembly language and in an appropriate high level language.

	Fall 2006	Fall 2007	Fall2008
MCU features and mechanics	75	92	75
Interrupt routine Mechanics	70	77	67
Questions relating to Port Access	40	38	58
PWM Timing	67	62	50

Table 5. Percentage of student meeting the program outcome: Be able to utilize current hardware and software tools and devices such as laboratory instruments, MultiSim, embedded controllers and PLCs.

Conclusion

We introduced a simple HC08 microcontroller device in our introductory MPU class with the intent to give students more confidence in their ability to use these devices in their capstone project class and hopefully into their careers. The simple 16 pin MC9S08QG8 device has all the capabilities we require in an introductory class and has the benefit of being cheap, available in easy-to-use DIP package and requires no external clock circuitry. This makes it ideal for small projects done by relative novice students. In doing this we opted to forgo the wide array of available demonstration boards produced by manufacturers for the purposes of education and training. We developed our own training I/O boards and revamped lectures and laboratories to fit the hardware. While our results did not show improvement in student's ability to use MPU devices practically, they did show that students were more likely to choose the HC08 in their capstone class. We consider this a success and will continue along this path while endeavoring to improve student's abilities in programming MPUs.

References

1. http://www.technologicalarts.com/ , http://www.coridiumcorp.com/ , http://www.microchip.com.

2. M. Parten, "Embedded Microprocessors in a Project Laboratory," *Computers in Education Journal*, Vol. XVI, No. 1, January – March 2006, pp. 95-101.

3. C.Choi. "Microcontroller Application Course and the Freescale Microcontroller Student Learning Kit". Proceedings of the 2008 ASEE Annual Conference and Exposition, Pittsburgh, PA, June 2008.

4. K.Bolding. "Upgrading a Microcontroller Systems Course Using the Cypress PSoC". Proceedings of the 2008 ASEE Annual Conference and Exposition, Pittsburgh, PA, June 2008.

5 http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=USBSPYDER08 .

6. http://www.microchip.com.

7. T.Almy, *Designing with Microcontrollers the 68HCS12*, 2007 edition.

8. http://www.softecmicro.com

9. Fabio Pereira , *HCS08 Unleashed: Designer's Guide To the HCS08 Microcontrollers*, 2008 edition.

10.http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC9S08QG8.pdf