Microcontrollers in Education: Embedded Control – Everywhere and Everyday

Eduardo Montañez
Freescale Semiconductor

Abstract

Microcontrollers (MCUs) are used for embedded control in virtually every field of science and engineering. The smallest MCUs have only six pins and are used in toys, appliances, and for tasks as simple as interfacing a few switches in a car door. More sophisticated MCUs have processing horsepower rivaling that of the most powerful desktop computers and are used to perform complex real-time fuel and spark timing computations in modern automobile engines. Everywhere and everyday, hundreds of MCUs drive the many appliances that simplify our daily tasks. To the consumer, MCUs typically go unnoticed, but in industry they are very important.

This paper stresses the importance of establishing a strong curriculum surrounding MCUs at various levels of engineering education. Most importantly, we must spark the student's interest in MCUs in introductory engineering courses by introducing the subject in a very simple form. Students can be easily discouraged by the subject if they are initially drowned by complex MCU architectures and robust programming languages. Though these two topics are necessary to have a full understanding of MCUs, students gain more interest in MCUs if they were first given a glimpse of an end application. This gives students a better sense of what they can achieve by learning more about MCUs.

This paper introduces one of the smallest and least expensive 8-bit MCUs. This MCU is in the MC68HC08 Family of MCUs and is available in both 8- and 16-pin DIP packages. The specific MCU used in this paper is the MC68HC908QY4[1] in a 16-pin DIP package, which features:

- *4K bytes of Flash and 128 bytes of RAM memory*
- *2-channel 16-bit timer with selectable input capture, output compare, and PWM*
- *4-channel 8-bit analog-to-digital converter*
- *Flexible high-current I/O and keyboard interrupts*

This MCU is ideal to ease students with little or no previous knowledge of MCU architecture into the world of MCUs. I will cover all the fundamentals to get started with this device, allowing the student to focus on the end application. Specific topics to be covered include:

- *Microcontroller Student Learning Kit (MCUSLK)[2]*
- *Monitor modes for serial communication*
- *Metrowerks' CodeWarrior development tools[3]*

In conclusion, a simple application will incorporate all the elements discussed in the paper.

"Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition Copyright © 2005, American Society for Engineering Education"

Microcontrollers in Industry

Have you ever stopped and retraced your daily routine to count the total number of electronic components that play a part of your life? You will be amazed to know how much we rely on electronics to get through our daily routine and be even more surprised that the majority of those components have a MCU embedded inside. MCUs have been heavily used in the automotive, industrial, and commercial business space. To give you an idea, I have listed several applications that use MCUs in the automotive, industrial, and commercial industry in Table 1.

Automotive	Industrial	Commercial
Safety equipment	Thermostat	Sleep number bed
Window lift	Fire detection	Alarm clock
Chassis	Alarm system	Digital television
Body electronics	Gas pump	Entertainment system
Transmission control	Traffic signals	Remote controls
Airbags	Credit card validation	Refrigerator
Radar detector	Automatic toll collection	Stovetop
Anti-lock brakes	Vehicle tracking	Dishwasher
Fuse relay replacement	Tank level monitor	Toaster
Sunroof systems	Fast food equipment	Coffee maker
Climate control	GPS	Microwave
Tire Pressure Monitor System	Medical equipment	Electric razor
Trip computer	Barcode scanner	Cordless curling iron
Seat systems	Cash register	Light dimmer
Engine control	Security camera	Weight scale
Lighting	ATM machine	Exercise equipment
Instrument cluster	Power over Ethernet	Electric toothbrush
Remote keyless entry	HVAC	Cellular phone
	Computer keyboard	Pager
	Computer mouse	PDA
	Inspection robot	Garage opener
	Warehouse management	Washer / Dryer
	Cellular base station	Exterior lighting
	Safety automation	Keyless entry
	Electronic window	Locks
		Sprinkler system
		Bicycle speedometer

Table 1: Applications with MCUs

This list reinforces the importance of making sure that all engineering students receive a basic understanding of MCUs. The task of introducing MCUs to students will be easier if the student can focus on an end application familiar to them. Also, with some MCU experience the engineering student is much more prepared and marketable to employment opportunities in the automotive, industrial, and commercial industries.

Copyright © 2005, American Society for Engineering Education"

Microcontrollers in Curriculum

From my experience in an electrical engineering program, I recall that the first reference to MCUs was not until I registered for a required sophomore level course titled, Introduction to Microcontrollers. In this course we spent the majority of the semester learning the Motorola MC68HC12 instruction set and how to write software algorithms in assembly programming language. The course did spend some time explaining the MCU architecture, but did not fully illustrate how an MCU is involved in a completed application. Therefore, most students in our program only associated MCUs with complex assembly programming and did not see an MCU as a crucial hardware component in a larger system. Since this was the only required MCU course in our curriculum, many of my peers justifiably did not see the importance of MCU technology in comparison to other digital logic.

I know that my MCU experience does not apply to all engineering programs, but through talking to various professors the majority of introductory courses seem to focus on teaching students assembly programming. Strangely, there must have been something about assembly programming that was intriguing to me since I did register to take an elective course titled, Microcontroller Applications. This course along with the next upper division elective titled, Microcontroller Interfacing, led me to a job in the semiconductor industry. It was the hands-on experience and application specific labs in these upper division courses that allowed me to visualize how a tiny chip can process and control various factors. These courses taught me that MCUs went beyond programming. Specifically, they taught me more about selection process of other digital components to interface to the MCU. Also, because MCUs are so flexible in comparison to other digital logic components that are not programmable, the courses taught me the various ways of implementing an application.

Learning how to write software and knowing the MCU fundamentals are very important. However, this paper stresses bringing a glimpse of the MCU end application into an introductory engineering course. For example, when a student is tasked with performing a digital logic function with the minimum number of components (NOR gates, NAND gates, INVERTERS, etc.) they should be introduced to using an MCU as an alternative solution. With an MCU, a student could reduce the total number of components necessary for performing a digital logic function to a single chip. Unlike digital logic components, which perform a set task, MCUs have internal memory that can be programmed to perform much more complex functions without additional components. Adding complexity to a function requires only more software in a single MCU. By configuring the software, an MCU can be used to sense inputs, decode data, and generate outputs. In addition, an MCU like the MC68HC908QY4 contains various peripherals like a timer, pulse-width modulator, analog-to-digital converter, and keyboard interrupts. These peripherals provide configurable functions without excess software algorithms. MCUs are becoming smaller and less expensive. The MC68HC908QY4, available in an 8- or 16-pin package and costing less than a dollar is a good fit be combined with digital logic courses at an introductory engineering level.

*"Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition
Copyright © 2005, American Society for Engineering Education"*

Getting Started with the MC68HC908QY4

These remaining sections will guide you through all the steps necessary to begin development on an HC08 family MCU using Freescale's Microcontroller Student Learning Kit (MCUSLK). The MCUSLK is an all inclusive kit designed to provide students with all the components necessary to learn MCUs. The kit is intended to be used in introductory level courses, MCU interfacing and applications courses, and in senior level design courses. In addition to an MCU design environment, the project board included in the kit can be used stand-alone for prototyping and testing non-microcontroller electronic circuits. Figure 1 shows the MCUSLK and the components that it includes:

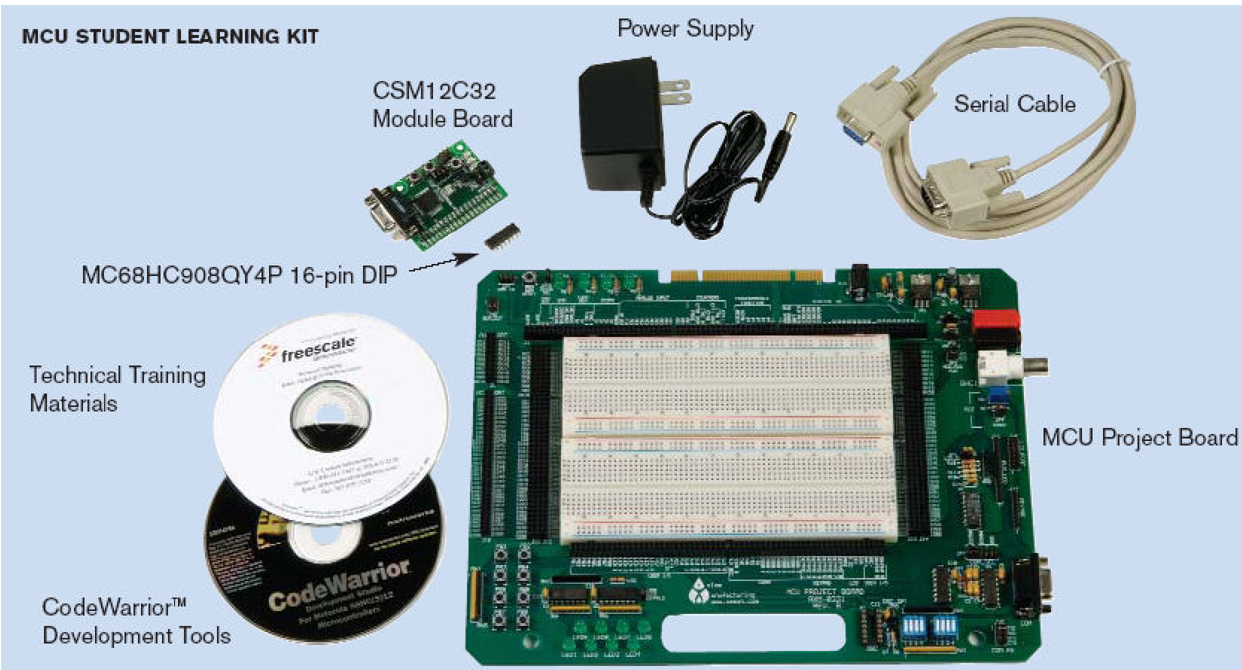

Figure 1: MCUSLK

The MCU project board is similar to a typical student prototyping board with the addition of several MCU specific features. Some of these features include:

- MCU interface connector: Allows students to plug in various available module boards for development:
  - CSM12C32 module board (HCS12 Architecture) - included
  - CSM08RG60 module board (HCS08 Architecture) – purchased separately
  - CSM56F801 module board (DSP Architecture) – purchased separately
- Prototyping area accommodates the MC68HC908QY4 16-pin DIP MCU.
- MCU signal breakout logically arranged around prototyping board.
- User selectable voltage to on-board logic devices and MCU (3.3V or 5V).
- User Input / Output components: 8 push button switches, 8 user LEDs, 8 user DIP switches, 1 multi-turn user potentiometer, 2 banana connectors, and 1 BNC connector
- LCD module and 8-pin keyboard interfaces

Copyright © 2005, American Society for Engineering Education"

- COM port with selectable RS232, Background Debug Module (BDM), and Monitor Mode (MON08) interfaces
- MON08 interface for programming and debugging HC08 MCUs
- BDM pod for programming and debugging HCS12 / HCS08 MCUs
- Compatible with National Instruments Educational Laboratory Virtual Instrumentation Suite (NI ELVIS), which is a LabVIEW based, hands-on design and prototyping environment. NI ELVIS consists of LabVIEW virtual instruments, a multifunction data acquisition (DAQ) device, and a custom-designed bench-top workstation.

The hardware setup section explains how to configure the MCU project board for use with the 16-pin MC68HC908QY4 MCU. This includes wiring up the HC08 family MCU in normal monitor mode to perform a one-time load of user monitor software, which allows you to program and debug your application in a simplified user monitor mode setup. The hardware setup section also provides more information on normal and user monitor modes. The software setup section introduces the Metrowerks' CodeWarrior development tools, which are needed for uploading the user monitor software and for compiling, uploading, and debugging your application software.

Hardware Setup

The reason the MC68HC908QY4 is very adequate for an introductory MCU course is because it does not require many components to get the device up and running. Once the student receives the 16-pin device, all they will need to do is apply power, add a simple mode switch, and connect a single wire serial communication line to the device, which are all available on the MCU project board. Following these three simple steps, the student can begin developing and debugging their application in hardware and software.

The MC68HC908QY4 can be configured to run in multiple operating modes, but for simplicity this paper recommends that students perform all their application development in user monitor mode. User monitor mode allows a student to program, erase, and debug their software through a single wire serial communication interface called MON08. The Metrowerks' CodeWarrior development tools are able to communicate to the device through the MON08 interface on the MCU project board. However, prior to entering user monitor mode a one-time procedure to load a user monitor software must be performed. The user monitor software is a program that resides in a protected section of the device's flash memory. The user monitor software is necessary in this mode to interpret debugging commands from the development tools. Another important thing to note about user monitor mode is that the device executes user software from an internal generated clock versus other operating modes that require an additional external clock component. It is user monitor mode that provides students with the simplest wiring setup to get started with their application.

To perform the one-time procedure to load the user monitor software, the device must be set up in normal monitor mode. Normal monitor mode is a device operating mode that allows a user to program, erase, and debug their software through a single wire serial communication interface. Unlike user monitor mode, normal monitor mode does not require user monitor software to be preloaded into the device. The normal monitor mode wiring is not as simple as in user monitor mode since it requires several device pins to be pulled high or low on power up. One pin requires a high voltage input during device power up to enter the mode. This mode also requires a 9.8304 MHz external clock device to operate. A high voltage input (VTST) and a socket for an external clock are available on the MCU project board. It is recommended that the steps necessary to prepare the device for the student with user monitor software be performed by the professor or a teaching assistant. See Figure 2 for the normal monitor mode wiring on the MC68HC908QY4 to load the user monitor software.

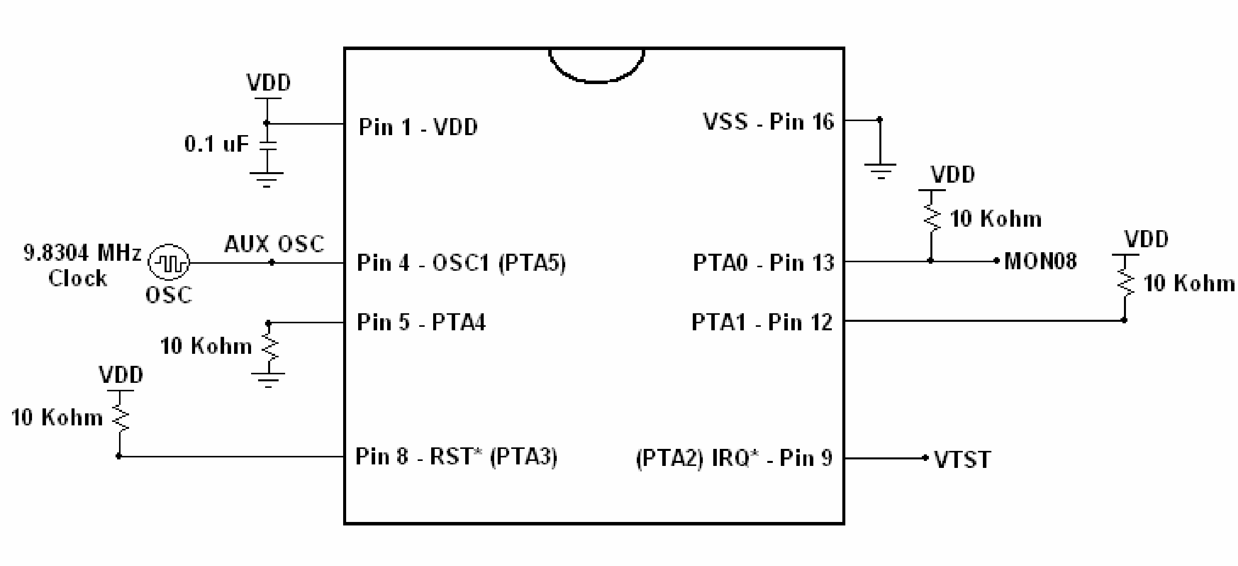

Figure 2: Normal Monitor Mode Circuit on MC68HC908QY4

After the user monitor software has been loaded through normal monitor mode, the student can take the device and begin their development through the simplified user monitor mode wiring shown in Figure 3:

*"Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition
Copyright © 2005, American Society for Engineering Education"*

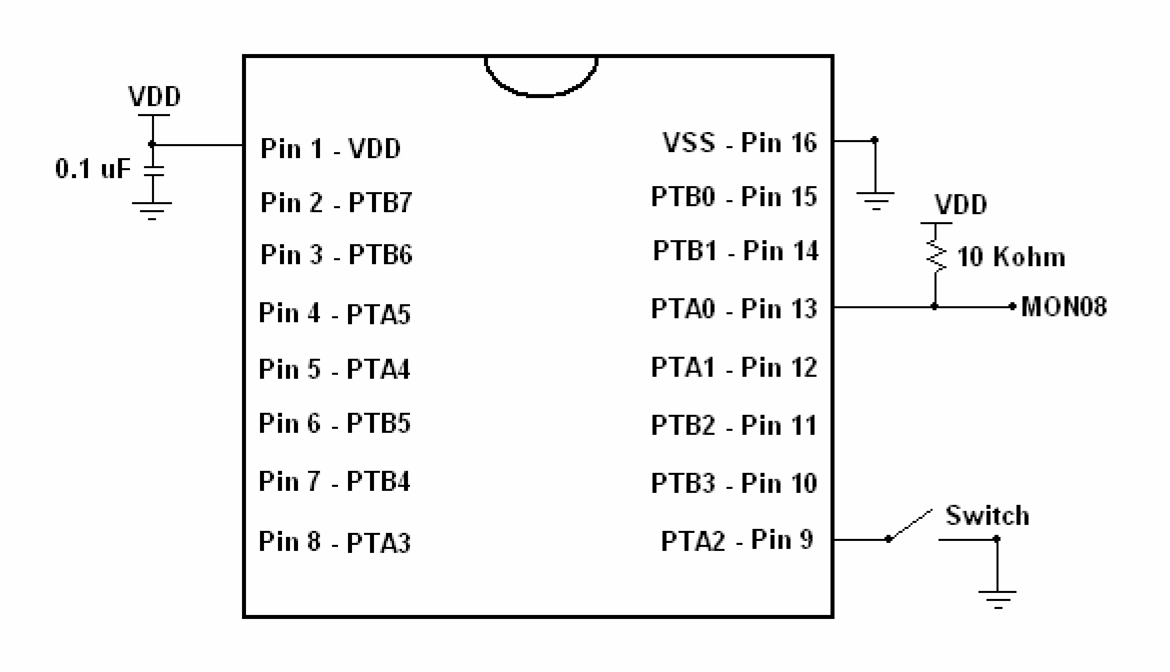

Figure 3: User Monitor Mode Circuit on MC68HC908QY4

For detailed instructions on configuring the MCU project board for MON08 communication and wiring up the MC68HC908QY4 in both normal and user monitor modes refer to Getting Started with the Microcontroller Student Learning Kit (MCUSLK): Using the MC68HC908QY4[4] user guide on the Freescale university program website.

Software Setup

Included in the MCUSLK are the Metrowerks' CodeWarrior development tools. We recommend the CodeWarrior development tools because they support the same user interface in their integrated development environment (IDE) and debugger across their HC(S)08, HC(S)12, and DSP architectures. Therefore, a student starting to learn about MCUs in an introductory course with the MC68HC908QY4 would begin using the CodeWarrior development tools for the HC(S)08. Then, if the student decides to pursue the next MCU course, the professor can introduce a more complex MCU like the HCS12 with a similar environment in the CodeWarrior development tools for the HC(S)12. The CodeWarrior development tools include an assembler, a C compiler, a simulator, and real-time debugger.

To perform the one-time procedure to upload the user monitor software on the MC68HC908QY4 the HC(S)08 development tools must be installed. With power applied to the device in the normal monitor mode configuration the user can proceed to program the MCU with the user monitor software associated with application note AN2305. For detailed information on the user monitor mode software refer to AN2305: User Mode Monitor Access for MC68HC908QY/QT Series MCUs[5]. For detailed instructions installing the CodeWarrior tools and programming the MC68HC908QY4 with the user monitor software refer to the Getting Started with the Microcontroller Student Learning Kit (MCUSLK): Using the MC68HC908QY4 user guide on the Freescale university program website.

Copyright © 2005, American Society for Engineering Education"

After that procedure is completed, the student can create their own applications on the MC68HC908QY4 by using the user monitor template project accessible through the HC(S)08 Board Support Stationery in the CodeWarrior IDE project wizard. Once the student writes the software and compiles it successfully they can program the device through the CodeWarrior debugger. The debugger gives students the ability to trace or step through their application software. It also allows the student to erase and reprogram the device at anytime to make modifications to the functionality inside the device. The graphical debugger shown in Figure 4 provides students with several windows to view the loaded source and assembly code, internal device memory, software variables, and central processing unit (CPU) registers. For detailed instructions in creating a user monitor project, programming the device, and debugging their application with the CodeWarrior development tools refer to the Getting Started with the Microcontroller Student Learning Kit (MCUSLK): Using the MC68HC908QY4 user guide on the Freescale university program website.

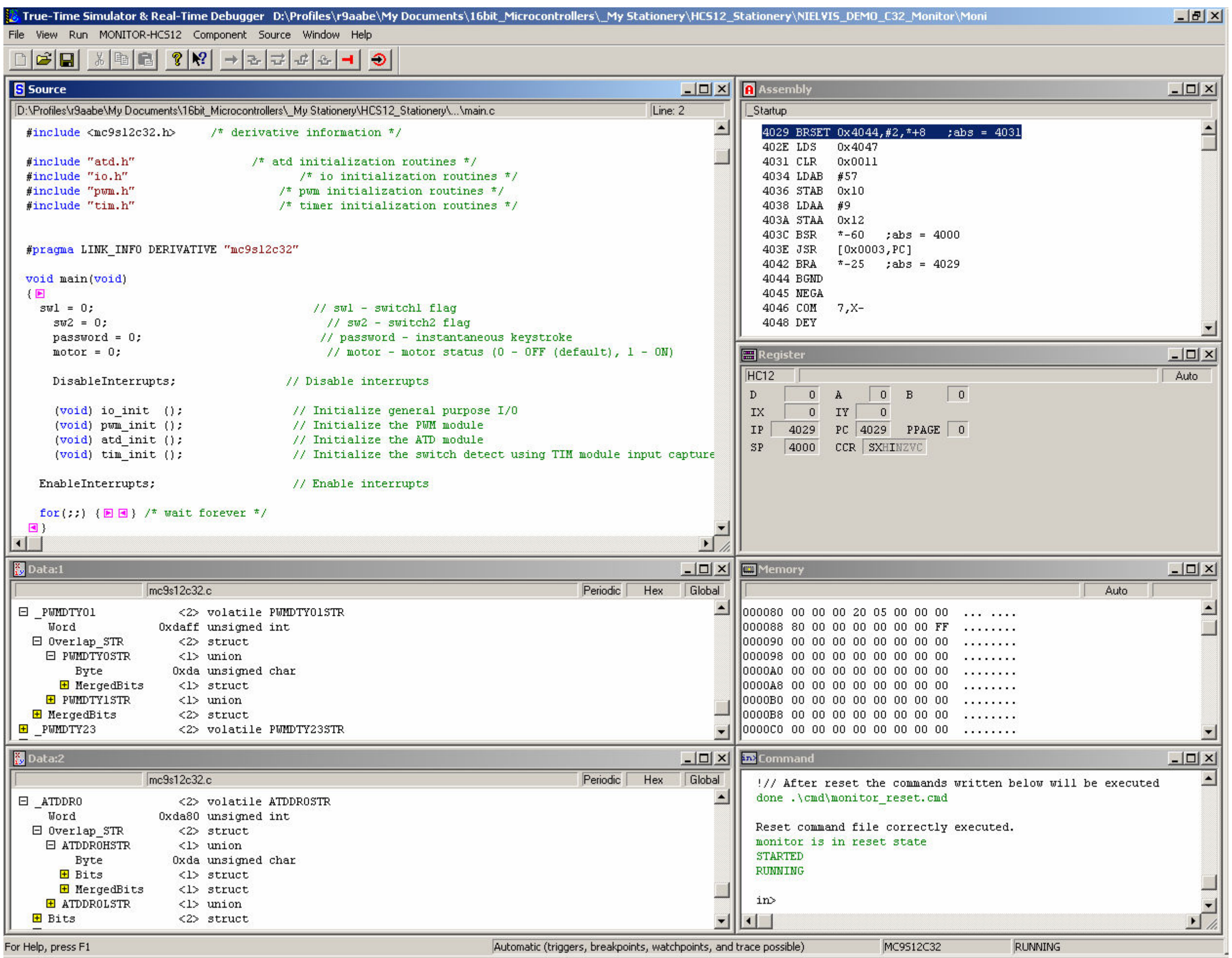

Figure 4: CodeWarrior Debugger

Application Example

To demonstrate the appropriateness of introducing the MC68HC908QY4 in an introductory engineering classroom the application example which I selected is a lab problem in a sophomore level digital logic course. The problem states: Design an "Identichron" with the minimal number of digital logic components. This is an access code lock with a 4-bit input code, a 7-segment display, and two LEDs (Enter and Alarm). The input code bits are labeled W:X:Y:Z respectively. The 7-segment display segments are labeled a = top, b = upper-right, c = lower-right, d = lower-left, e = bottom, f = upper-left, and g = middle. The truth table (Table 2) is the following:

Person	Status	Code	7-segment display	Enter LED	Alarm LED
Craig	President	0101	'C' = a,d,e,f	On	Off
Frank	Fired	1101	'F' = a,d,f,g	Off	On
Don	Vice President	0111	'D' = a,b,c,d,e,f	On	Off
Paula	Spy	1111	'P' = a,b,d,f,g	Off	On
Others	---	----	'U' = b,c,d,e,f	Off	Off

Table 2: "Identichron" Truth Table

After analyzing the problem, I calculated that the solution required 4 digital inputs for the person's code identifier plus 6 digital outputs to drive the respective segments to illuminate the first letter of the person's name and 2 digital outputs to signal the Enter or Alarm LED. Even though this problem could be solved in handful of digital logic components the 16-pin MC68HC908QY4 MCU is a good alternative for a single chip solution. After 3 of the 16 pins were committed to power (Vdd), ground (Vss), and the single wire serial communication line (MON08), I used the 13 remaining pins to wire up the required digital input and output lines to solve this problem. See Table 3 for the MC68HC908QY4 pin assignments to complete this lab problem.

Port Pin	Label	Direction
PTA[1]	Z	Input
PTA[2]	Y	Input
PTA[3]	X	Input
PTA[4]	W	Input
PTA[5]	Enter LED	Output
PTB[0]	a	Output
PTB[1]	b	Output
PTB[2]	c	Output
PTB[3]	d	Output
PTB[4]	e	Output
PTB[5]	f	Output
PTB[6]	g	Output
PTB[7]	Alarm LED	Output

Table 3: MC68HC908QY4 Pin Assignment

Now, all I had to do is write the software to decode the person's identifier code and drive the corresponding 7-segment display letter and LED. This required the software to configure the device's port pins appropriately and a few lookup tables matching up the input code with the corresponding outputs. This is a simple application example demonstrating the simplicity of the MC68HC908QY4 and how it can be associated with introductory engineering courses.

Copyright © 2005, American Society for Engineering Education"

Conclusion

This paper shows the large demand for MCUs in industry, but most importantly demonstrates the importance of stressing MCUs in engineering curriculum. Offerings like the MCUSLK allow students to learn various MCU architectures, while learning how to debug their own end application. And with MCUs becoming smaller, simpler, and less expensive, the MC68HC908QY4 is a device that will allow students to learn the fundamentals of MCUs at an industry level. Therefore, if we provide students with the right tools and find ways to spark their interest in MCUs, they will encounter many more opportunities in industry.

References

[1] Freescale Semiconductor, *MC68HC908QY/QT Data Sheet*, Freescale Semiconductor, Austin, TX (2004)

[2] Freescale Semiconductor, *Microcontroller Student Learning Kit: Product Preview*, Freescale Semiconductor, Austin, TX (2004)

[3] Metrowerks, *CodeWarrior Development Studio for HC(S)08 Microcontrollers (Special Edition)*, Metrowerks, Austin, TX (2004)

[4] Eduardo Montañez, *Getting Started with the Microcontroller Student Learning Kit (MCUSLK): Using the MC68HC908QY4*, Freescale Semiconductor, Austin, TX (2005)

[5] Jim Sibigtroth, *User Mode Monitor Access for MC68HC908QY/QT Series MCUs*, Freescale Semiconductor, Austin, TX (2003)

Biographical Information

Eduardo Montañez received a BS degree in Electrical Engineering with a technical concentration in Computer Engineering and Integrated Electronics from The University of Texas at Austin in 2001. For the past 3 years, Eduardo has worked for Motorola Inc. (now Freescale Semiconductor) as an Applications Engineer in their 8/16 Bit Microcontroller Division.

Copyright © 2005, American Society for Engineering Education"