

# **Innovative ‘HCS12 Microcontroller Lab Station Using Limited Lab Resources**

**Christopher R. Carroll  
University of Minnesota Duluth**

## **Abstract**

Many current introductory microcomputer system design courses employ the Freescale ‘HCS12 sixteen-bit microcontroller as a vehicle for lab exercises. That processor, a descendent of the very popular Motorola MC68HC11 8-bit microcontroller, includes a flexible instruction set that supports teaching modern memory addressing modes, register manipulation at the bit, byte, and word levels, and advanced applications such as fuzzy logic. The ‘HCS12 also includes a wide variety of input/output features embedded in the microcontroller that can support many different lab exercises. Wytec’s Dragon development board, based on the ‘HCS12 microcontroller, provides an excellent platform for exercising the ‘HCS12 features, and is often used as a lab station in microcontroller classes. However, the Dragon development board is intended for use with a dedicated personal computer at each lab station. There are several disadvantages to supplying each lab station with its own personal computer. First, it’s expensive. Second, it clutters the lab station with extraneous hardware. Third, the elegant and efficient design of the Dragon board is overshadowed by the glitzy personal computer, its keyboard, and display. This paper describes a simple and less expensive lab station that maintains the Wytec Dragon board and all its features as the foundation but removes the need for the dedicated personal computer at each station by enhancing the Dragon board with new software and innovative input/output additions. This lab station design demonstrates the Dragon board and its capabilities to students, without hiding the ‘HCS12 behind a personal computer interface.

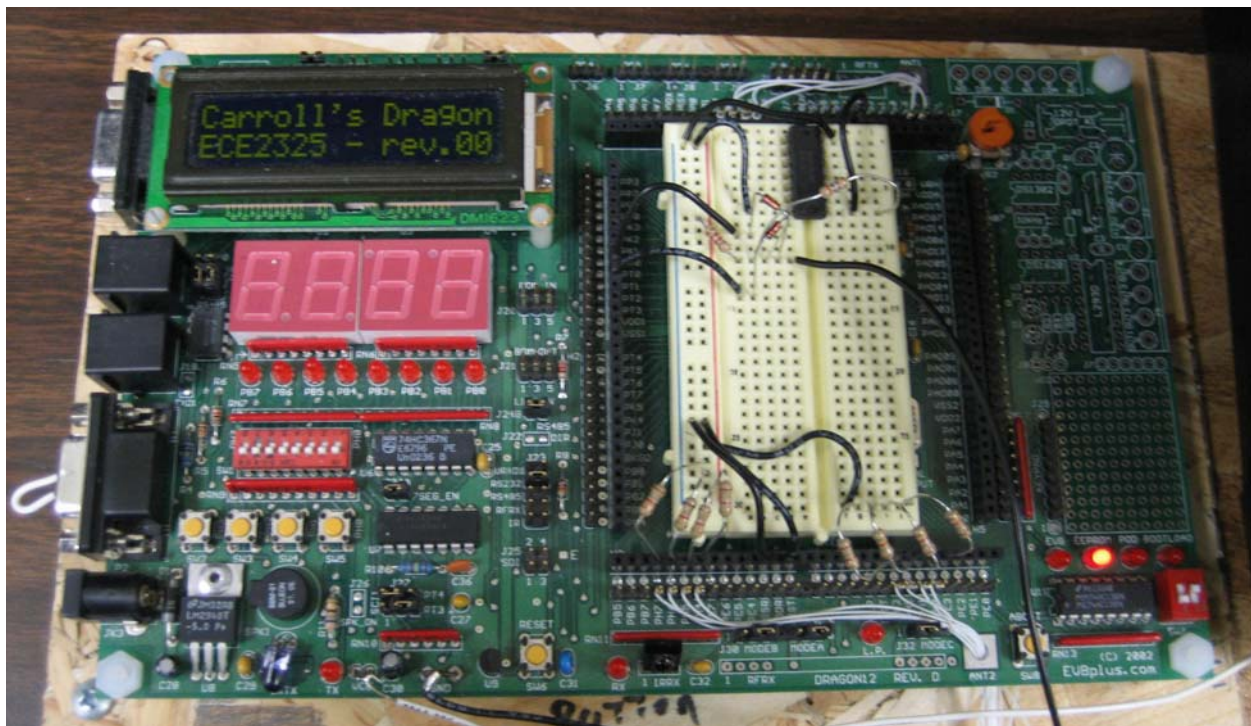
## **Background**

The lab environment created by this innovative lab station has been described in another ASEE paper (Carroll, 2008). This document describes how this lab station has been used in class. The station was introduced in the Microcomputer System Design class in the Fall semester, 2007, and has been used now for two semesters as the lab station for that class. Experience with this lab station shows that students adapt easily to working directly with the Dragon development board, rather than facing yet another lab experience where they interact with lab equipment only through a personal computer interface.

Included here are examples of lab exercises that have been assigned using this lab station in the Microcomputer System Design course. These range from basic programming skill development to advanced applications of the timing system, communications capabilities, and analog signal processing capabilities of the ‘HCS12 and the Dragon development board. These sample lab assignments demonstrate the wealth of capability offered by this simple and inexpensive, yet powerful microcontroller lab station.

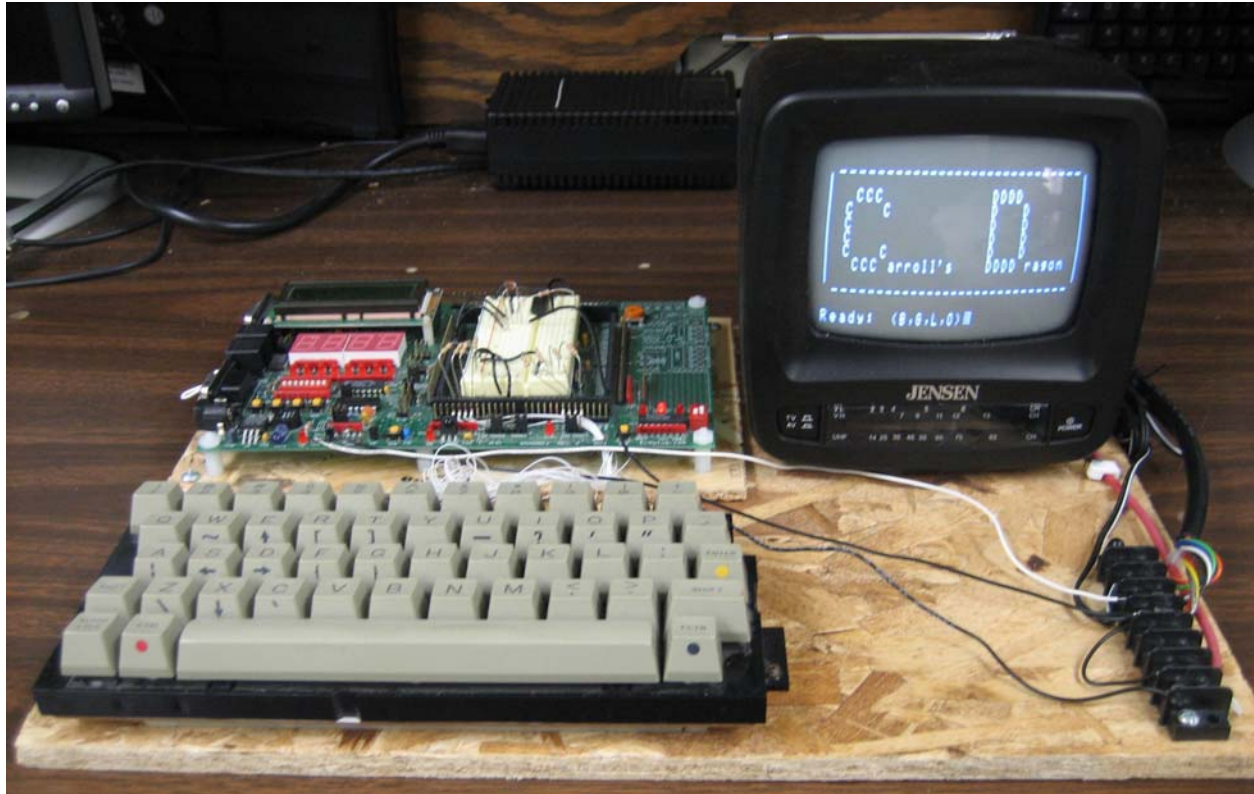
## The Processor

The Freescale 'HCS12 sixteen-bit microcontroller is a descendent of the very popular Motorola MC68HC11 8-bit microcontroller (Carroll, 2002). The instruction set of the 'HCS12 is a full superset of the instruction set of the MC68HC11, so all programming experience that users have with the MC68HC11 is directly applicable for writing programs on the 'HCS12 (Carroll, 1998). However, memory addresses and peripheral device addresses are different in the 'HCS12 processor, and in some cases, the internal peripheral devices operate slightly differently from those in the predecessor processor, so some effort must be employed to adapt to the 'HCS12. However, the 'HCS12 offers significant enhancement of capabilities through an expanded and more powerful instruction set, new addressing modes for accessing memory operands, and improved and expanded input/output capabilities in the internal peripheral devices included on the microcontroller. Figure 1 shows the Wytec Dragon development board used in this station.



*Figure 1. The Wytec Dragon development board as used in the lab station*

The lab experiments described in this paper use this Dragon development board as the foundation for a custom lab station. The station expands the Dragon board with an alphanumeric keyboard and an alphanumeric display, as discussed in an earlier ASEE paper (Carroll, 2008). Figure 2 shows the lab station as used in the lab. The goal of the lab station design was to make the station free-standing and autonomous, not needing a dedicated personal computer for each station in the way that the Dragon boards are intended to be used. By adding an alphanumeric keyboard and display to the boards, and linking those boards over terminal lines to a multi-user single host computer, this goal was achieved. The resulting lab station is easy to use, uncluttered, and fully exposed to student users so that they can see the hardware with which their programs interact, and so that they can see directly the effect of the input/output devices that their software uses.



*Figure 2: Lab station using the Wytec Dragon board, plus keyboard and TV display.*

Given the lab environment provided by the custom 'HCS12 lab stations described above, many lab exercises are possible. In the class supported by this lab, topics start with basic programming skills and simple parallel input/output. Once the basics are mastered, more complicated input/output interactions are addressed. Interrupts are added to provide a new technique by which software affects or responds to external events. The timing system of the 'HCS12, perhaps the most flexible feature of the microcontroller, forms the foundation for many lab experiments. Finally, when students have a good grasp of the hardware capabilities of the lab station, exercises that combine several I/O features into a fully functioning application round out the semester.

### **Experiments that Develop Programming Skills**

The first few experiments assigned to students in this microcontroller class involve exercises in which students get acquainted with the lab stations and system software, and write programs that do not require sophisticated input/output interaction. In these experiments students typically use just the simple pushbutton or DIP switch input provided on the Dragon board, and just the discrete LEDs as output. No complicated interaction with input/output devices is required.

#### **...First, Second, Third Assignments**

In the first three experiments, students perform some calculation on a number supplied via switch input, using the arithmetic instructions in the 'HCS12. Results are simply displayed on the eight individual LEDs as an 8-bit binary number. This assignment can range from very

simple, straightforward calculations to more sophisticated, iterative code structures. Calculations requiring some of the various multiply and divide instructions in the 'HCS12 instruction set force students to understand the capabilities and limitations of the various byte- and word-formats supported by these instructions, and encourage students to pay attention to the signed vs. unsigned number representation used by the various arithmetic instructions. The focus for these lab assignments is to understand how programs are written and how the various microcontroller instructions interact with data. No effort is spent on learning complicated input/output features.

### **...Fourth Assignment**

The fourth lab introduces the fundamental idea of stack data structures, subroutine/procedure implementations, and parameter/result passing into and out of procedures using the stack. This usually requires a significant step up in programming skill, and encourages students to consider structuring their code to improve readability and maintenance. Students must use indexed addressing modes to access data on the stack. A typical lab assignment of this type might require students to take a byte from switch input, reverse the order of bits in the byte, and display the resulting byte on the discrete LED output lights. Again, no sophisticated input/output techniques are needed to complicate the basic programming skills being addressed.

### **...Fifth Assignment**

By the fifth week of the semester, when this lab is assigned, students have developed some proficiency in assembly programming and are becoming comfortable with the lab environment provided by the lab stations. They are ready to begin interacting with more interesting input/output devices. One significant and very prominent output device on the Dragon boards is the four-digit seven-segment multiplexed LED display. Having students produce a four-digit output on that display requires that they develop some feel for the timing that's involved in instruction execution. Loops and other programming structures are required, and students can identify assembly language constructions that correspond to high-level language implementations that they have studied in other courses. Timing issues also can be addressed using the speaker on the Dragon board to produce sounds at various frequencies under software control, simply by having student programs toggle an output port bit at a specific frequency. Again using switch input, students write programs that simulate an organ keyboard or other musical instruments on which songs can be played.

The lab assignments up to this point merely require straightforward assembly language programming and interaction with simple input/output devices. The focus at this point is on writing the programs clearly and correctly, and avoiding unnecessary code clutter.

### **Experiments Using Interrupts**

By the sixth week of the semester, the course introduces interrupts. This adds a significant new capability for students to use in writing their programs. The 'HCS12, as well as most other microcontrollers, is designed with interrupt capability built into nearly all of the input/output devices on the chip. Interrupts are an essential part of assembly programming that students typically have not seen in their earlier experience with high level languages.

### **...Sixth Assignment**

The first interrupt-based lab assignments use interrupting sources that are very straightforward. These include the software interrupt (SWI), timer/counter overflow interrupt (TOV), and real-time interrupt (RTI). Each of these interrupt sources simply interrupts the running program when the associated event occurs, and typically needs no initialization other than establishing the interrupt vector to locate the service routine and enabling the interrupt. Experience with these initial interrupt-based lab assignments gives students the experience they need for upcoming experiments that involve more complicated interrupt situations. Typical lab assignments at this stage involve, for example, incrementing a number displayed (using one of the output devices studied earlier) at some regular interval, while the main program returns to the monitor and allows students to interact with the host computer or run various monitor commands while the interrupt occurs. This helps solidify the interrupt concept in students' minds.

### **Experiments Using the 'HCS12 Timing System**

The timing system in the 'HCS12 microcontroller is the most complex and versatile input/output feature of the microcontroller. Using this system, programs can take action or cause external actions to occur at pre-programmed times, capture the time at which external events occur, count external events, or a variety of other actions. Timing is an important feature of how assembly language programs interact with the external environment through input/output devices.

### **...Seventh Assignment**

An instructive way to introduce the 'HCS12 timing system capabilities is to revisit some of the lab assignments from earlier in the semester and solve them using timer interrupts. For example, generating sounds on the Dragon board speaker becomes a much easier and more straightforward task using the Output Compare feature of the timer, compared with timing the toggling of the speaker bit using software delay loops. Controlling the multiplexed seven-segment display using timer interrupts shows students how the processor can be freed of the task of running the multiplexing of the display. Adjusting the rate at which the multiplexed display is scanned and observing the effect on the operation of the main program teaches students that interrupts, though valuable and efficient, do not come for free, and that time taken by the interrupt routine execution is time that is not available to the main program to complete its tasks.

### **...Eighth Assignment**

A fun lab experiment, that involves input timing, captures infrared signals received from consumer-product infrared controllers, with which students are very familiar. The Dragon board includes an infrared sensor whose output can be applied to the 'HCS12 timing system input. The format for data transmitted by these infrared controllers is decoded easily by capturing the time at which the infrared light starts and stops, thus decoding the incoming 1's and 0's in the data being transmitted. Once the data bits are received, any number of actions can be undertaken by student programs in response to different button presses on the infrared controller.

## **Experiments that Implement Real Applications**

With all the capabilities of the 'HCS12 microcontroller available, students are finally ready to address real applications. In addition to the resources of the Dragon development board itself, students can also use the alphanumeric keyboard and alphanumeric display added to the Dragon board to enable interaction with the host computer. The 'HCS12 controls the switch-matrix keyboard through two eight-bit parallel ports, and the television display uses timing signals generated by the 'HCS12 timing system and serial data generated by one of the serial output devices, the Serial Peripheral Interface. The system software that interacts with these devices is discussed in class, but writing such software is beyond the scope of a normal lab exercise.

Instead, the system software provides several useful subroutines that can be called from user programs. These subroutines simplify the job of collecting keystrokes from the keyboard, and putting characters out to the display. Several handy subroutines print bytes or words as strings of hexadecimal characters, or print prompts or other messages to the screen in easily managed ways. The subroutines are not introduced to students until they have seen in class what actions are involved, so that nothing is hidden from students behind the veil of a "magic" subroutine. The provided routines simply give handy access to the keyboard and display input/output so that students can focus more precisely on the job at hand.

### **...Ninth Assignment**

A fun collection of lab assignments can be assigned where students create games in which they can interact either with each other, or with the computer, or competitively against the clock. Games that measure typing skill are easy to create. Playing a musical song defined in tables of pitches and durations is a good application of the timing system. The game of NIM, with the computer acting as the opponent, was a fun assignment.

### **...Tenth Assignment**

Character graphics on the television screen is easily added. Students were asked to plot various functions on the screen, using character graphics. Game playing fields are easily created using character graphics on the screen, and assignments involving video game activities are fun towards the end of the semester.

### **...Project Assignment**

The final lab assignment can combine applications of many different features of the 'HCS12. One such assignment required students to create a "jukebox" capable of playing any of three different songs selected by an infrared remote control. Buttons on the remote were used to raise or lower the pitch of the song being played, or to increase or decrease the playing speed of the song. Characteristics such as the song number, speed, and pitch were displayed on the seven-segment display, and the display brightness was controlled by a pulse-width modulator using an analog potentiometer input. Solutions to this assignment involved seven or eight interrupting sources all controlling activities related to various inputs or time intervals.

## Summary

This lab station using Freescale's 'HCS12 microcontroller and Wytec's Dragon development board has been used in an introductory microcontroller course during the 2007-08 academic year. It continues to be the core instructional facility in the microcomputer lab here. The sample lab exercises described in this paper are examples of actual assignments used during the past year of classroom activity. Students worked hard to implement solutions to these assignments. The lab station and its processor form a powerful vehicle for teaching microcomputer system design. Expansion of lab assignments into other more exotic areas, such as bit-mapped graphic display (Carroll, 2004), should be possible, and will be investigated in upcoming semesters.

## References

- Carroll, C. R. (2008). Innovative lab station using the Freescale 'HCS12 microcontroller and Dragon development board. *2008 ASEE Annual Conference Proceedings*. Pittsburgh, PA.
- Carroll, C. R. (2004). Video graphics using the SPI on the MC68HC11 microcontroller. *2004 ASEE Annual Conference Proceedings*. Salt Lake City, UT.
- Carroll, C. R. (2002). New life for the MC68HC11 evaluation board. *2002 ASEE Annual Conference Proceedings*. Montreal, Canada.
- Carroll, C. R. (1998). Design workshop experience using the Motorola 68HC12 microcontroller. *Proceedings of the 1998 North Midwest Section Meeting of ASEE*. Winona, MN.

## Biography

CHRISTOPHER R. CARROLL earned academic degrees at Georgia Tech and Caltech. He is Associate Professor of Electrical and Computer Engineering at the University of Minnesota Duluth. His interests are digital systems and microprocessor applications, especially as they relate to educational environments.