Moving the C Language Course into the Real World

David Delker, Les Kinsler
Engineering Technology Department, Kansas State University – Salina

Abstract

The Kansas State University-Salina Engineering Technology Department offers a C programming course for non-Computer Science Technology majors. The course is designed to teach the C language syntax, structured programming design and implementation, and to link the language to applications within the students' areas of specialization. In the Fall 2000 semester, a local manufacturer requested help in upgrading an existing microprocessor-based control application. The industry partner agreed to allow the C programming class to write the application as a class project. This project required the students to incorporate interrupts, timing, signal monitoring, real-time calculations, and extensive interfacing to input and output devices. This paper focuses on the unique interaction between students and industry and the benefits that this real-world process brought into the course and into the academic program.

Introduction

Applications in C Programming for Engineering Technology, CMST 222, is a course designed to introduce non-Computer Science Technology majors to the C language. Most of the students in this class have limited exposure to third-generation programming languages. Applied Basic Programming is the only course prerequisite to CMST 222. The students in this class come from a variety of majors including Mechanical Engineering Technology, Electronic Engineering Technology, and Computer Engineering Technology. Consequently, the class is made up of a diversity of backgrounds, abilities, and technical interests.

The course is a three credit hour course that meets three days a week in a lecture setting. The students do homework and laboratory assignments on their own time. The course works on a traditional scheme of introducing students to top-down program design. The first half of the semester is devoted to understanding constructs, creating flowcharts, and developing algorithms.

During the second half of the semester, the class was divided into teams of five students. Each group was to design and code a statistics program. Each team was given a data file format along with a specification of statistical data that were to be calculated. Each team was to break down the problem into functions. Teams then assigned functions to specific team members. Each member was required to submit the function prototype: the function name, return type, and the list of parameter types. The program assignment required that no global variables be used. Once the prototypes were completed, each member of the team was to design and code their assigned functions. In addition, each member was required to write a main program to call the functions, produce the required statistical values, and output the program results.

The final step of this project required that each member of the group submit their coded functions. Copies of the functions were made for each member of the group and then each individual debugged the program independently.

This group project provided the students an opportunity to learn both about programming and about working with other people. The students learned to evaluate the capabilities of the other group members and to make assignments accordingly. Obviously, the students learned a great deal about functions, parameters, multi-file projects, linking modules, and debugging other people's code.

Traditionally, the semester would be filled with additional programs, and then each student would be required to develop an individual project and present it to the class in the final week of the semester. However, this year a local manufacturer, Bergkamp Inc., requested help in upgrading a microprocessor-based monitoring system for a slurry paver. The Bergkamp family are alumni and strong supporters of engineering and engineering technology education at Kansas State University and agreed to support the upgrade as a project for the C programming class.

Paver System Description

The micropaving process involves the mixing of four primary components: rock, emulsion, water, and cement. These components are mixed in a pug mill, dropped into a spreader box, and then spread in a thin layer to extend the life of existing pavement. The monitoring program is designed to capture data on the amount of rock, emulsion, and cement used, along with the distance traveled and the area surfaced. In Figure 1, the emulsion tank is on the left side, rock bin in the center, water tank is on the right, and the cement is in small bin on the right side of the operator's platform. The pug mill is centered and lies just below the operator's platform. You can see the rubber spout leading from the pug mill to the spreader box. The spreader box is not shown in the picture below, but would hang from the loops of the rear hitch.

Figure 1: Paver operator's platform

The paver itself is powered by a diesel engine driving hydraulic pumps that, in turn, provide power to hydrostatic motors. The paver's rock bin feeds a hydrostatic motor-driven belt that moves the rock through a gate to the pug mill for mixing. The emulsion and water are delivered to the pug mill via pumps powered by one hydrostatic motor. These two quantities are statically linked and therefore only the emulsion is monitored.

The cement is the minor component and is augured into the pug mill again by a hydrostatic motor. Each component's rate of deliver is determined by the rpm of the hydrostatic motors. Each motor can be adjusted independently by the operator riding the rear platform. The motors have Hall Effect switches that provide 60 pulses per revolution; these pulses provide input to the microcontroller. Four hydrostatic wheel motors propel the spreader. The original program was designed to take the pulse output of one of these wheel motors to determine the distance covered. The update to the program was to use an industrial radar gun to determine the speed and distance instead of the wheel motor. This update was done solely on the request of the owner. The owner thought the radar gun would provide more accurate readings. The spreader box can vary from eight to fourteen feet in width.

The front receiver hitch and hopper allow nurse trucks to supply the paver with emulsion, rock, and water. The paver has a capacity of up to 4000 lbs. of rock per minute and can propel a total weight of up to 130,000 lbs. up a 15% grade.

Figure 2: Paver receiver hitch and driver's platform

Monitoring Equipment

A Sauer-Sunstrand KDC2 System 90 Microcontroller[1] was used to monitor the system. This microcontroller was selected because of its rugged construction and wide use in the paving industry. The KDC2 is based on an Intel 8XC196KC microcontroller. This controller has many features including:

- 16 bit ALU
- Fast hardware multiplication and division
- On-chip A/D converter
- High speed inputs
- High speed outputs for generation of digital waveforms
- Two Hardware and four software timers
- Full duplex serial port
- Dedicated digital I/O pins
- Interrupt system allowing generation of internal and external events.

The cross compiler for the project was obtained from BSO Tasking. The display used on the paver is a Densitron 4 by 40 LCD display with backlighting and an RS232 interface.

Because the microprocessor's limit of eight digital inputs, four button switches were mounted below the display and the last line was used to define the current function of these four switches. A fifth button switch was labeled "Escape" and served as a return to the parent screen. A sixth keyed switch was used to reset accumulated totals.

The radar gun is a product of Dickey-John Corporation. The unit is an industrial hardened unit designed for applications involving harsh environments, including large temperature changes, dirt, and vibration. The radar gun outputs 44 Hz. per MPH.[2]

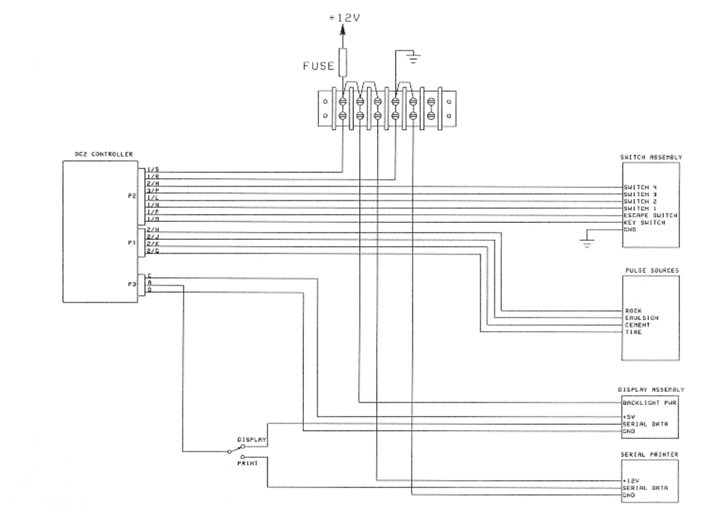

Figure 3: Circuit block diagram

A serial printer is mounted on the paver. The printer used the same RS232 interface as the display. A toggle switch was installed to direct the output to the printer or to the display. The serial printer provided the operator with a hard copy of summary information.

The Assignment

The class was separated into four teams of five students. The instructor created the teams to balance the expertise and to distribute evenly the students from differing majors.
Students were given background on the operation of the machine. They were also given the instruction manual of the existing monitoring system. This manual included user screens (as the one displayed in Figure 4[3]) and operation instructions and expectations of the system.

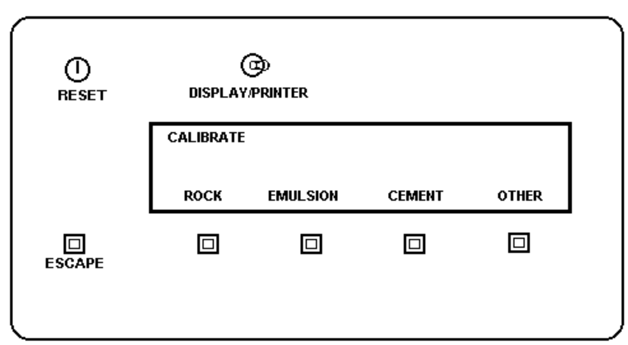

Figure 4: User screen

The students were also given a set of functional requirements for the system. The system must:

- Monitor the amount, the rate, and percentages of the slurry components.
- During the run mode, display in real time the accumulated amount, rate, and percentage of the materials, distance traveled, and current speed.
- Allow the user to calibrate the rock, emulsion, cement, and the radar gun to determine the value associated with one pulse.
- Store and retrieve the calibration constants and accumulated totals from an EEPROM.
- Print to the printer all the accumulated totals, percentages, distances, and area covered.
- Allow the user to reset the totals to zero.

The hardware was set up in a special projects lab. The microcontroller, display, keyboard, along with function generators to simulate the motor pulses were set up in a central bench. Each team had a single computer to compile, debug, link, and download to the microcontroller.

The use of an existing user interface and defined hardware setup provided a great deal of structure to the students' assignment. The students could concentrate on the problem of implementation rather than system analysis, since programming is the primary goal of the C for Technology course. The first assignment for each team was to develop a list of equipment that would be needed and a list of topics to be covered in order to complete the project. These lists were to be developed outside of class. Classroom time was used to develop an algorithm of the tasks required to implement one of the major components. The class as a whole developed the calibration routine. This activity was very rewarding and got the entire class involved in the project. The design created in the classroom activity identified a series of tasks that were added to the students' topic list to be covered in the lectures. The students ended up outlining the lecture topics for the rest of the semester. Because the students generated the topic list, they took great ownership in discussions of implementation details and possible variations. The following is a list of major topics identified by the students as essential to the completion of the project.

- Special Function Registers: This topic allowed the review of the microprocessor's architecture, along with discussion of bit masking and manipulation techniques.

- Interrupts: This project required the students to understand the enabling, generation, and handling of interrupts. The high-speed inputs were used to monitor the rock, emulsion, cement, and radar gun. The internal timer register overflow was also enabled.

- Timing: Timing was accomplished by setting the timer register and enabling the overflow interrupt. Students then had to consider how to accumulate time into seconds and minutes. In this application, timing was restricted to changes in time for the sake of calculating flow rates and speeds; no time-of-day computations were necessary. Students did note that this microcontroller lacked an ability to maintain a clock and that a hardware design change using a Dallas Semiconductor memory chip could extend its capabilities.

- Switches: Students learned to de-bounce switches and to determine if the switch button is released. Students quickly learned that in many algorithms determining that a switch is released may be just as important as determining that a switch has been pressed.

- Serial Communications: Students learned to set the baud rate, number of stop bits, and parity of the RS 232 port.

- EEPROM: The Microcontroller used an EEPROM to store data values over periods when the power would be shut down. This was not an automatic operation and students needed to identify the locations in the code when reading and writing to the EEPROM were required. The microcontroller included a library of EEPROM utilities that allowed students to perform I/O. However, in order to use these utilities the data values needed stored in a structured date format or structs. Students also were encouraged to minimize the writing of data to the

EEPROM to extend its life. A simple algorithm to check that the data were correctly copied was also implemented.

- Include Files: Students developed their own include files to define key names and constants for the program. In this project it quickly became apparent to the students that a defined name (e.g. KEY_1) made coding and debugging simpler than a hexadecimal numbered address.

- Cross Compiling: Very few of the students had considered compiling on one platform and downloading to a different architecture. This C course is a prerequisite to other courses dealing with embedded systems applications.

Outcomes

The outcomes from this project were most rewarding. The students were very motivated during the development of the paver monitoring system. They asked more questions and discussed more issues than in the preceding sections of the class. Several students in the class commented that they felt more confident that they could be productive programmers because of their success in implementing a real project rather than a classroom project. Several instructors of other courses even commented how much time the students were spending in the special projects lab and commented on the intensity of there work. Some students from other classes also inquired about what was going on in the lab and what was all that new equipment?

The topics that were covered in the project were more meaningful to the students as they could apply them immediately to a live project. The topic list was not significantly different than in previous years; however, the linkage, depth, and application of those topics were significantly improved.

The group dynamics were good for the students. Students learned to delegate responsibility and to pick up the slack when someone did not perform according to expectations. Students were responsible for evaluating themselves and their teammates based on their contributions to the overall success of the project. Students were required to submit individual reports at the end of the project on the lessons they learned, ranging from programming and design, to group dynamics and communications.

Each group presented their final project to the class. The presentation had to involve all the members of the group. Each group demonstrated their final product. Additionally, each group had to address the most significant problems that they encountered, unique problems that they solved, and finally any weaknesses in their code.

Figure 5: Students presenting projects in the lab

Conclusions

This project was tremendously successful in motivating students to excel and learn. It made the C Programming for Engineering Technology class much more fun and fulfilling for both the students and the instructor. By attaching the project to a real-life application, the students got a much more satisfying feeling of achievement when they finished the task. Since authors Delker and Kinsler designed the original control system, the questions posed by students could usually be anticipated and solutions formulated with a minimum of research on the part of the instructor.

This type of project is not easy to find on an every-semester basis. The class was fortunate that Bergkamp, Inc. had significant lead-time to allow the program modifications to be completed. The school was also fortunate to have supportive alumni willing to participate in this kind of classroom exercise. At the completion of the project, the company, the instructor, and (most of all) the students were quite satisfied with the results.

Bibliography

1. KDC2 Users Manual, Sauer-Sundstrand, August 1993.
2. URL: http://www.dickeyjohn.com/Ag_Products, Manufacturer of the radar gun.
3. Operating Manual for Computerized Material Monitoring, Delker and Kinsler, Bergkamp Inc.

DAVID G. DELKER
David G. Delker is a Professor and Head of Engineering Technology at Kansas State University-Salina. His research and consulting interests include the hardware and software design of microcontroller-based industrial instrumentation. He received an A.T. degree in Electronic Engineering Technology from Kansas Technical Institute, a B.S. in Technical Education and an M.S. in Electrical Engineering from Oklahoma State University.

LESLIE A. KINSLER
Les Kinsler is an Associate Professor of Engineering Technology at Kansas State University-Salina. His research and consulting interests include programming in C, Visual Basic, and implementations of database systems. He received a B.S. from Emporia State University in Physics and Mathematics and a M.C.S. from Wichita State University with an emphasis in Software Engineering.