HO Scale Model Trains Provide a Vehicle for Curriculum Fusion Between Electrical Engineering and Computer Science

Bryan S. Goda James H. Wise Mark S. McConkey James R. Loy

INTRODUCTION

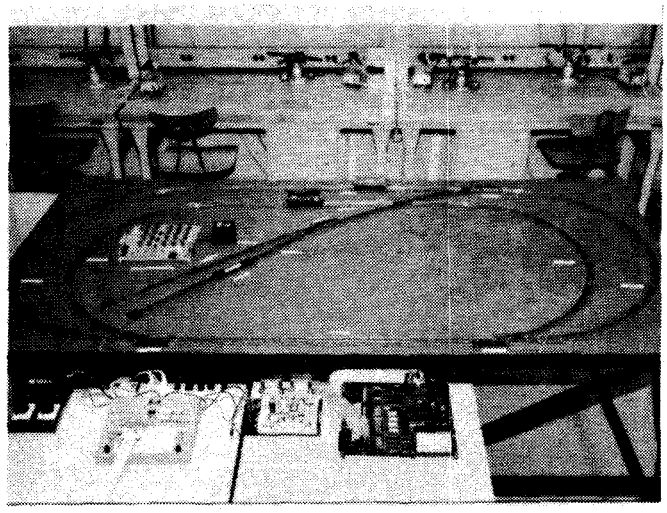
Model trains provide an ideal vehicle for promoting digital computer design concepts while simultaneously integrating the computer science and computer engineering curriculums. Our research employed two categories of HO scale model train systems. The first was the traditional version available at most hobby shops. The second was an intelligent version which uses independent microprocessors in each cab for advanced control. Each system proved well suited for certain specific lab objectives, and together spanned the applications spectrum. Model trains have also helped to energize the curriculum fusion process.

There are numerous reasons for using model trains in a computer science and electrical engineering curriculum. Model trains stimulate student interest because they are fun and can be used to simulate complex systems. Model trains have proven to be a flexible tool, being utilized in the basic digital logic course through the capstone design project. Considering the relatively small cost of a model train set, model trains can be a cost effective tool to model such things as a state machine, collision avoidance, and deadlock avoidance.

Approximately 60% of the faculty members at the United States Military Academy (USMA) are rotating faculty, who are active duty Army officers that the Army sends to various academic institutions to earn master's degrees before coming to West Point to teach for three years. Rotating faculty members come to the academy with state of the art knowledge and represent a good cross section of a variety of academic programs. Several of the rotating faculty had observed the use of model trains at the University of Colorado and the Georgia Institute of Technology during their master's programs and decided to interject this concept into their work with cadets at USMA. Currently, we are utilizing two different model railroad sets.

USING CONVENTIONAL HO TRAINS

The first model railroad set uses standard HO rail, switches, locomotive, and power supply. This standard set has been modified with the addition of six photo sensitive detectors to determine train position and an in-housed develop digital interface to a PC expansion card. The PC expansion card consists of two 8255 programmable peripheral interface (PPI) integrated circuits. This interface allows the rail system to be controlled by a programmable logic device (PLD) under PC monitoring, a PC software control program, or any microprocessor embedded system. The general track layout appears in figure 1.



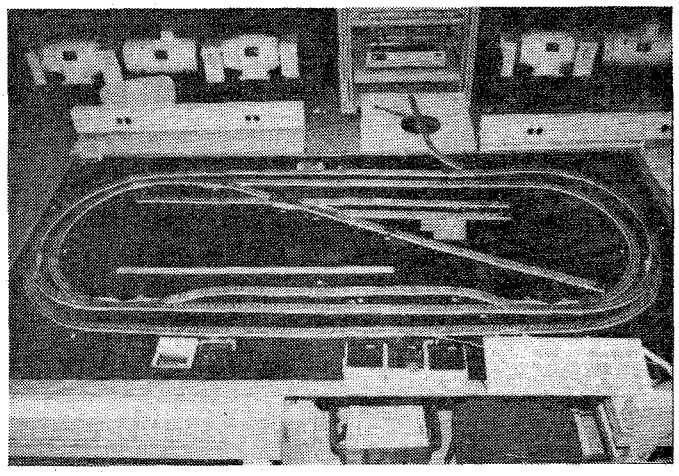
Conventional HO Train Set
figure 1

The two ovals are divided into five electrically isolated segments. Each segment may have one of two mutually exclusive power supplies applied in either the forward or reverse direction to control locomotive movement. The interface also allows digital control of each track switch direction.

USING THE MARKLIN DIGITAL TRAIN SYSTEM

The second model railroad set employed is the Marklin digital train set. The Marklin digital train set is a state of the art model railroading system which easily permits the integration of hardware and software. Each train has an on-board microprocessor that can set the train's speed, direction, and decouple cargo cars. By using the serial port of either an IBM PC or an Apple Macintosh, the trains, switches, and track detectors can be monitored by the computer and integrated into a real-time hardware/software design system.

There are several advantages to the Marklin system over a conventional HO railroad set. The Marklin system runs on a 3 rail AC power system, thus certain sections of track do not have to be turned on or off. The on-board microprocessor selects the required components of the AC signal to maintain the required speed and direction, permitting up to 80 trains to be run on the same section of track. A toggle signal is sent to a digital decoder buffer when a train passes over a sensor and this buffer is read by the computer interface every 3ms. This signaling system permits multiple trains to send signals to the computer without sequence timing problems. We can utilize the power of the Marklin system to model sophisticated real-time systems.



Marklin Digital Train System
figure 2

CURRICULUM AND LAB APPLICATIONS

The remaining sections of this paper describe the breadth of course applications that have utilized model trains. The fusion effects of model trains help to foster teamwork among electrical engineers and computer scientists. The marriage of hardware and software has stimulated student interest and generated excitement in the classroom.

Finite State Machine Control

The conventional rail system is used in the introductory digital design course to facilitate a student designed and built real time control system using a finite state machine. The student project must determine train position (using the photo sensitive detectors), apply the correct power supply and direction on the each segment of track, and throw the switches in the proper manner to make two locomotives travel in a specified pattern. The design process consists of five steps. The student must interpret the design requirement presented as a word problem, synthesize a state diagram, develop the text file to program a PLD, program the device, and verify correct operation of the rail system.

The benefit of using the model train system is twofold. From the student's perspective, he gets immediate feedback of the success of his state machine design. This makes the design project seem "real", provides tangible results, and is enjoyable for the student. The fact that the student learns state machine design is almost secondary to their enjoyment in making the system work. From the instructor's perspective, he has a design project that is extremely flexible. To avoid project duplication from term to term, he simply needs to change the train pattern. Each new pattern constitutes a significantly different state machine without having to create a new design projects each term.

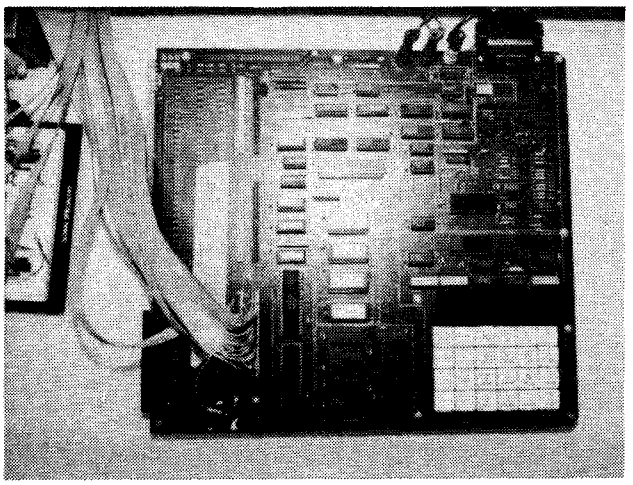
PC Control

The conventional rail system is easily controlled using a standard PC executing a graphical interface written in the C programming language. This provides the student with a highly flexible software control system. Changes can be quickly implemented for any desired train pattern. As a capstone design project, students have implemented this software control using the ADA programming language. Both

applications clearly illustrate the design tradeoffs between software and hardware control.

Microprocessor Control

Train control is a excellent example of a design application in the microprocessor design course. Students utilize the SDK-86, which is an integrated microprocessor embedded control system based on the Intel 8086 central processing unit. Both address and data buses with all associated control signals are terminated to fifty pin jumper strips. This allows easy access and utilization in any student's digital interface design. Its versatile indigenous input/output (I/O) components include two 8255 PPI integrated circuits providing six *parallel* ports, one 8279A integrated circuit controlling a 22 button keypad and eight, seven segment LED displays, and one 8251A Programmable Communication Interface (PCI) integrated circuit providing a synchronous or asynchronous CPU polled *serial* port.

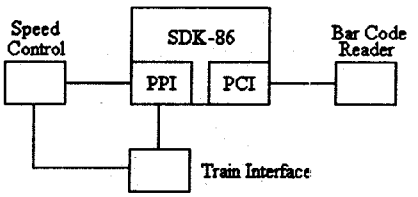


SDK-86 Microprocessor
figure 3

The design concept is a three tier modular procedure. First, the student utilizes either indigenous or their own digital design as the hardware interface between the SDK-86 and the external controlled device. This allows complete flexibility in the design and implementation of the interface while encouraging ingenuity and efficiency. Second, the student develops a software program to provide or increase monitoring and control capacity. This program is complied into a binary file and downloaded over the serial port from a personal computer to the SDK-86. Third, the student uses both the monitor interface program and a digital logic analyzer as methodical debugging tools to detect and correct errors. This three step process provides the student enough structure to solve any design problem.

Speed determination and regulation models realistic situations where the rate of movement is critical to the train's correct, safe and timely operation. The student is required to regulate two standard HO scale train power supplies or "throttles" each with three push button inputs: speed up, brake, and quick panic stop. Additionally, the student must decide how to determine the train throttle speed and provide the information back to the SDK-86. This is a classic analog to digital interface design problem. Speed regulation is achieved by using open collector buffers to drive low current reed switches that open and close the three push button inputs.

To increase efficiency and minimize control signals, a three input to eight output decoder allows three control signals to select one of the six push button throttle inputs. Speed determination is achieved by using an ADC0820C analog to digital converter (ADC) to monitor the analog voltage sent to the train engine and translate it to an eight bit digital value. Each ADC circuit requires only a single control signal to initiate the conversion process. Combining the interface requirements, the student can achieve both speed determination and regulation using two, eight bit input buses and a four bit output control bus.



SDK-86 Block Diagram
figure 4

Using a bar code reader, each train's bar code provides train identification, designation, routing, and inventory information. The bar code reader detects the train's presence through an infrared beam reflection, scans the bar coded information sticker, and presents a serial information sequence. This requires a simple serial interface that accepts a serial data stream from an external device, performs a serial to parallel conversion, and offers that parallel data to the CPU. The PCI performs this function flawlessly. Consequently, the train can be identified, designation determined, route verified and altered as track conditions dictate, and load

inventory identified. By reading a strategically positioned bar coded inventory label located on each train car, the microprocessor can analyze the car's inventory to determine and control the switch yard railing, train coupling, and track routing requirements that ensures proper inventory movement.

Inventory accountability and tracking is enhanced by the ability to monitor each railroad car's load as it transits through any switching yard. Additionally, comparison of shipping orders to a master inventory listing allows the microprocessor to determine and adjust the current inventory and restocking requirements for any warehouse location. If the bar code reader reports an error, the embedded controller can reroute and slow the train for another reading attempt. This creates a natural relationship between deterministic routing and speed control.

Any embedded controller design must have some degree of software control and intervention. The student is provided with several C functions which control the aspects of the model railroad system common to any possible interface design. These functions control the photo sensor detection, track switch direction and power supply status, assignment, and motion direction. The remaining functions and integrating software logic will be unique to the student's interface design. Consequently, they become a portion of their design solution. This exploits the strengths of a combined hardware and software control system. The throttle speed determination and regulation software routine must periodically poll the two ADC input PPI ports and adjust the train's speed by selecting the appropriate decoder input. Using the C input and output byte commands, it is a straightforward code sequence to read the ADC input, make a software decision, and output the desired decoder value.

To improve ergonomics, each ADC value can be displayed on a set of seven segment LED displays. The routine to read the bar coded information is slightly more complex. The PCI, operating as a CPU polled device, requires the CPU to periodically check the PCI status register to determine if the bar code reader has information to pass. Once passed to the CPU, the software logic must correctly decide what action, if any, must be taken. The decision matrix and implementing code ranges from easy to quite complex. Factors influencing the decision include a successful read, train route, inventory, and destination capabilities Obviously, the read must be successful. Otherwise, the train must be slowed and rerouted to allow another reading attempt. The train route determines the desired destination and rerouting possibilities. Inventory information allows stockage tracking and whether the desired destination is ready to process the load. Under adverse conditions, the final destination may be altered to facilitate inventory off load and prevent both delays and spoilage. The entire decision matrix can be implemented using a combination of *switch* and *if, then, else* statements. The number of sequential statements depends on the student's programming proficiency.

Stack Operations

Trains systems have been used in computer science to model the conversion of infix formulas to reverse polish notation We can also use trains to model stack operations such as push and pop. Suppose we have two locomotives pulling a mix of two different types of cargo cars and we want to have one locomotive pulling one type of cargo car and the other locomotive pulling the other type of cargo car. If we combine the two trains together we have in fact done a push operation. Then by breaking apart the train one car at a time and routing it to the correct spur based on its type, we have completed a single pop operation. The moving and combining of cars is made possible by a solenoid coupler on each locomotive and a decoupler switch to separate the cars. When all of the cars have been correctly routed, we are back to an empty stack condition and our simulation is complete. Learning to program these operations in C gives a student a physical model how stack operations occur in calculators and computers.

Operating Systems

Operating systems is one of the best areas that model trains can be used to illustrate concepts. In figure 2 we have one train running on the inside loop, one train running on the outside loop, and one train alternating between the two. Collisions are possible in several sections of track. An algorithm must be developed to first detect when collisions are possible and send control signals to the trains to avoid collisions. These decisions require a real-time use of computer programming and provide an excellent model to illustrate real-time hardware/software design. With three or more trains a possible deadlock condition could occur where there could be no trains moving. Thus we can implement the dining philosopher's problem in a train model.

Capstone Design Projects

Model train sets offer an excellent vehicle for creating undergraduate capstone design projects. Cadets at USMA work one on one with a faculty

member over their last semester to produce a project that demonstrates they have the knowledge to apply what they have learned in the classroom. Train related projects won both the computer science and electrical engineering prizes for excellence this year. Three computer science majors programming in ADA created a train control system that avoided collisions. Programming a train set using ADA is a good small scale model for an embedded system used in guided missile and tracking control. An electrical engineering major utilized the computer signaling available from the digital train set to create his own model turntable. A stepper motor was used to gain precise rotational control and a photoelectric detector circuit was designed to monitor when track alignment was correct. Cadets working with the trains were highly motivated to finish their projects and took great satisfaction with their finished products.

CONCLUSIONS

The HO scale model railroad design project is truly a diverse, convenient, and cost effective vehicle for stimulating the students' desire to learn. Design and study is no longer viewed as "work" and is fun for both the faculty and student. It has provided the thread of continuity to fuse together topics throughout the electrical engineering and computer science disciplines. Starting with a seemingly difficult design requirement, the student's interest and ingenuity is challenged by application of the modular design process to a tangible and realistic control problem. Each student can readily monitor their work's progression and receive immediate feedback as they observe the model trains operate. The student's desire to successfully make the trains operate is paramount. The students are motivated to solve the train design problem and readily absorb the learning objectives along their route to successful train operation. If the task is interesting and personally satisfying (i.e. fun), the student will attain and retain their gained knowledge. Which, after all, is our primary objective.

The authors would like to acknowledge the funding provided by the Faculty Development and Research Fund, Academic Research Division, Office of the Dean, United States Military Academy, who made this research possible.

REFERENCES

1. Tanenbaum, A., Structured Computer Organization, 1990.

2. Catherall, T., Marklin Digital User's Guide, 1991.

*****

Bryan S. Goda

Bryan Goda is an active duty Army Major specializing in mobile subscriber communications. He is a 1982 graduate of the United States Military Academy (BS) and a 1993 graduate of the University of Colorado (MSEE). He is currently assigned to the Electrical Engineering and Computer Science Faculty at the United States Military Academy. He has previously served in Germany and Operation Desert Storm. His research interests include parallel processing and real-time hardware software design.

James H. Wise

Jim Wise is an active duty Army Captain specializing in tactical communications. He is a 1984 graduate of the United States Military Academy (BS) and a 1992 graduate of Georgia Institute of Technology (MSEE). He is currently assigned to the Electrical Engineering and Computer Science faculty at the United States Military Academy. His research interests include design with programmable logic devices and microprocessor based systems.

Mark S. McConkey

Mark McConkey is an active duty Army Major specializing in aerial detection and identification of moving objects. He is a 1983 graduate of the United States Military Academy (BS) and a 1993 graduate of Georgia Institute of Technology (MSEE). He is currently assigned to the Electrical Engineering and Computer Science faculty at the United States Military Academy. He has served in Germany and Operation Desert Storm. His research interests include design with programmable logic devices and microprocessor based systems.

James R. Loy

LTC James Loy received the B.S. degree from the U.S. Military Academy, West Point in 1974, and M.S. , M.E. and Ph. D. degrees in Computer Science and Computer Engineering from Rensselaer Polytechnic Institute. He is currently serving as a Permanent Associate Professor in the Department of Electrical Engineering and Computer Science at West Point. His research interests are in the area of high speed RISC processors and Computer Aided Design tool development. He is a member of IEEE, and currently serving as Vice Chair of the Mid-Hudson Chapter of the IEEE.