INFORMATION AND NANO TECHNOLOGIES IN AN INTRODUCTORY ENGINEERING COURSE

Gary H. Bernstein*, Jay Brockman, Matthew Buckle, Andrew Carter, Patrick Fay, Tom Fuja, Natalie Gedde, Stephen Govea, Angela Lewis, Leo McWilliams, Kerry Meyers, Wolfgang Porod, Stephen Silliman, and Clement Suhendra

College of Engineering
University of Notre Dame
Notre Dame, IN 46556
**574-631-6269*
bernstein.1@nd.edu

1. INTRODUCTION

Clearly, information technology is changing the way we live; just as importantly, nanotechnology is transforming information technology, and will continue to do so at an increasing rate. These areas of technology will continue to converge, and in the future will represent considerable employment opportunities for today's young engineers. As a motivational topic, as well as preparation for various engineering disciplines, we have developed a six-week module for Freshmen engineering students to focus on the combined topics of information and nano technologies (IT Nano). The material introduced in the module is intended to give the students a broad overview of these important areas. This paper discusses the context and content of the module as well as the results of surveys that determine student attitudes about their experience.

Many excellent courses on nanotechnology have been reported (see Crone, *et al*., 2003 and references therein.) Uddin and Chowdhury (Uddin and Chowdhury, 2001), in presenting possible undergraduate engineering curricula for nanotechnology, suggest that nanotechnology concepts should be introduced in the Freshman year. Appropriately, nanotechnology is filtering down to the K-12 educational level (Centeno, *et al*., 2004; Lakhtakia, 2006) in order to motivate students for science and engineering careers.

Our 3-credit EG10111 and EG10112 courses form the first-year engineering experience for Notre Dame Engineering intents (Pieronek, *et al*., 2005; Uhran, *et al*., 2005; McWilliams, *et al*., 2004; and Brockman, *et al*., 2002), after which students declare their engineering majors. Enrollment for the Fall of 2005 semester was around 330. These two courses are based on a total of four, six-week modules. In the Fall of 2005, the first module pertained to basic engineering principles using a ball-launcher as a platform, the second was the IT Nano module discussed here, the third, EG10112 in Spring of 2006, is an introduction to computer programming based on MATLAB, and the fourth a demonstration of feedback control using a chemical reactor. The overarching learning objectives of the IT Nano module include familiarization with digital systems and an awareness and understanding of the implications of trends in nanotechnology. This new module is based on an embedded system that uses LEGO Mindstorm with two RCX controllers to demonstrate the IBM Millipede, which is a high-density storage medium under development by IBM based on nanoscale cantilevers and tips (Vettiger, *et al*., 2000). The IBM Millipede tip melts nanometer-sized pits into a substrate and then reads them back as stored data. In this module, the students build the cantilever, or "arm," of a large-scale Millipede model. The arm reads data represented by LEGO bricks on a moving LEGO X-Y table. Students breadboard electrical circuits that communicate through an optical fiber with an RCX to initiate scanning, and proceed to read 64 bits of "data" from the table.

2. COURSE CONTENT AND STRUCTURE

Module topics were chosen to simultaneously teach new engineering skills, inform students about fields related to information technology, and to give them a flavor for what future applications they might find in nanotechnology. Stress was placed on the interdisciplinary aspects of all materials.

Learning objectives for the module were to:

1. Understand what is meant by information and nanotechnology and their impact on society
2. Understand the role of electronics in modern information technology applications
3. Understand the basic ideas underpinning optical communications
4. Understand concepts from electronics and circuits, including:
   A. Mechanics
      i. Use a breadboard to construct a simple digital circuit using basic TTL or CMOS logic and simple discrete devices.
      ii. Be able to identify an integrated circuit, transistors, light emitting diodes, capacitors, and resistors.
      iii. Use a DMM for simple resistance and voltage measurements.
      iv. Understand the concepts of series and parallel electrical connections.

   B. Applications
      i. Describe the manufacturing process for integrated circuits in general terms
      ii. Know the basic components of a computer system and be able to relate them to their use in the system
      iii. Understand sensors and their use in computers and information technology

   C. Microprocessors
      i. Understand the basic structure and function of a microprocessor
      ii. Understand the use of binary numbers in computers
      iii. Understand the concept of bits as physical entities such as voltages and currents
5. Discuss problem decomposition through applications to programming and the project
6. Write basic programs in the Not Quite C programming language applying loops and conditional statements
7. Decompose and run a challenge project
8. Understand Moore's Law and other trends in future technologies

Lectures presented (by Bernstein) in the module were:

L1: Introduction to the module
L2: Feedback and control in the Millipede
L3: Algorithms and introduction to NQC
L4: NQC control statements
L5: Basic electronics using water analogies
L6: Basic electronics continued including breadboarding
L7: Codes and communications
L8: Huffman codes
L9: Guest lecture, Pat Toole, IBM – technology development
L10: Inside a microprocessor – basic logic circuits
L11: Guest lecture, Tom Fuja, Notre Dame – data representations and compression
L12: Guest lecture, Wolfgang Porod, Notre Dame – nanotechnology overview
L13: Integrated circuits and IC fabrication
L14: Wrap-up and student surveys

The course is partitioned into classroom lectures and hands-on laboratory experiences, dubbed "learning centers" (LCs) after the environment used for this purpose. In this module the 14 lectures were held on Mondays and Fridays for 50 minutes each, and 5, 75-minute LCs were held once per week. A final LC was held in which students demonstrated their completed Millipede designs. Finally, a group report detailing their designs and results was required. Throughout both of the EG10111 and EG10112 courses, lectures are presented to the entire class, whereas LCs are held in sections of 30 or fewer. In the LCs, the students work on their projects in groups of 5 or fewer, with many activities performed in pairs. Weekly homeworks and quizzes support the lecture materials as well as lay the groundwork for the Millipede model design.

3. TECHNICAL MATERIAL PRESENTED IN THE MODULE

3.1 The RCX and Millipede project

The Millipede model is based on three distinct components, *viz.* an electronic circuit that initializes the actions of the model, an RCX-controlled LEGO X-Y table with bricks representing the Millipede substrate that holds the data pits, and an RCX-controlled LEGO arm representing the cantilever and tip. An example of a complete Millipede model is shown in Fig. 1.

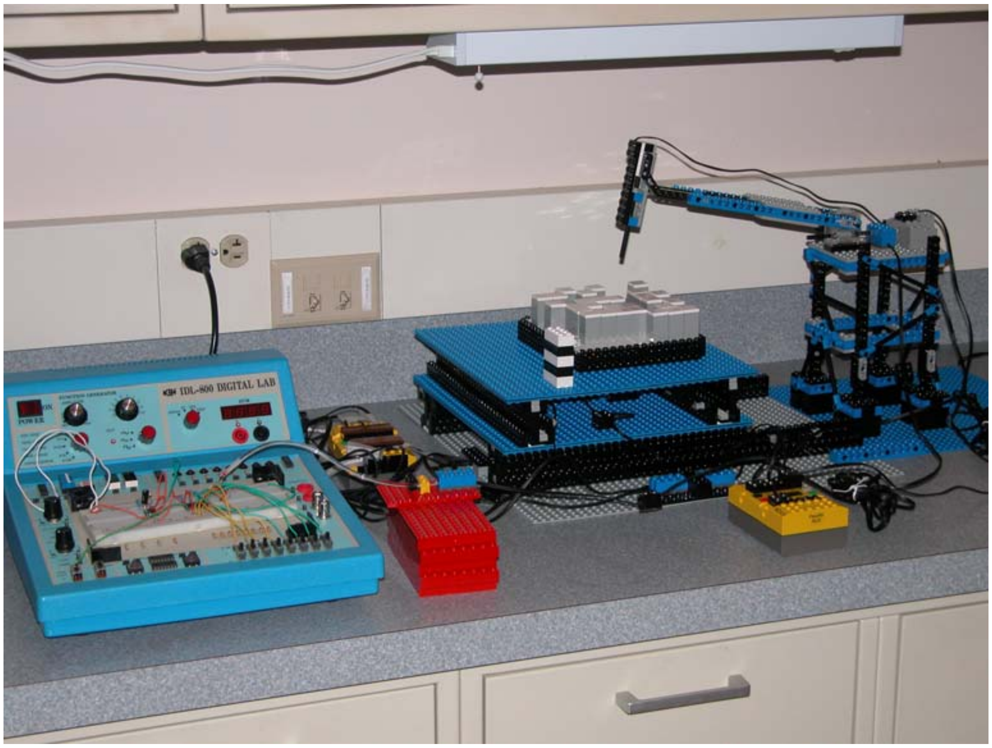

Figure 1: Complete Lego RCX-controlled model of IBM Millipede data storage device. On the left is the IDL-800 breadboard system, in the center is the X-Y stage on which sits the digital "data" under measurement, and on the right is the cantilever, or "arm." A timer and shift register provide a bi-phase coded initialization signal to the X-Y stage RCX via a plastic optical fiber. At the end of the cantilever is the tip connected to a touch sensor. When the arm moves down, a rotation sensor provides information about the height of the bit.

The circuit (schematic shown in Fig. 2) consists of a 555 timer (NE555) as clock and an 8-bit parallel-in, serial-out shift register (LS165A) that transmits a prewired, two-bit, bi-phase coded message via a light emitting diode (LED) and plastic optical fiber to the arm RCX as a go/ no-go signal. The RCX uses the LEGO light sensor as the input device for optical communication with the circuit. This part of the project motivates the related topics of electronic circuits, integrated circuits, fiber optics, and coding. As part of the design, students are provided schematics and a tutorial on breadboarding, and asked to design the values of resistance and capacitance that will cause the data to shift at a frequency of about 1 Hz. Also, the duty cycle should be sufficient to be able to see the LED blinking at the output of the clock circuit.

The second component is the X-Y table, which, due to its relative complexity and limited time resources, is provided to the students along with its control NQC program. Students must understand the commands to which it can respond, and how to initiate and receive communications from the table. The table, based on a microscope stage design, has racks and gears on two levels, with one motor driving the top platen and two motors driving the bottom platen. The table control commands are encoded in a Huffman scheme, thus providing the opportunity to discuss and practice efficient codes for data transmission. In fact, the Huffman code is useful in this context, since an inefficient code might cause the scanning of the table to be needlessly long, *i.e.* up to several minutes. The table can respond to the following bi-phase input codes provided to the students:

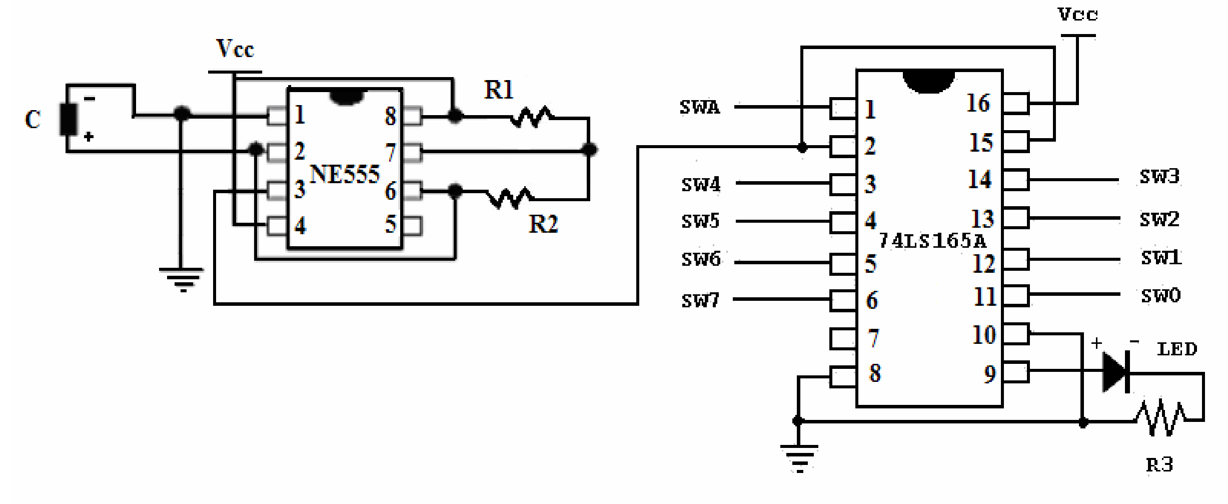

Figure 2: Schematic for electrical circuit used to initiate operation of the Millipede model. The 555 timer is designed by the students to oscillate with a period of about 1 Hz. This clock drives the parallel in/serial out shift register (SR). Data into the SR forms a bi-phase code that is shifted out to the LED whose light is transmitted to a LEGO light sensor on the Millipede Arm via an optical fiber. The circuit and bank of input switches can be seen in Fig. 1. The arm NQC code detects the correct code and starts the Millipede operation.

Table 1: Huffman Codes

Huffman Code	Description
0	Move the table in the 'y' direction by one bit (indexRow)
10	Move the table in the 'x' direction by one bit (indexColumn)
110	Rewind the table to the first bit in a column (rewindColumn)
1110	Rewind the table to the first bit in a row (rewindRow)
11110	Calibrate the table to the first column (calibrateColumn)
11111	Calibrate the table to the first row (calibrateRow)

The third component, designed, built, and programmed by the students, is the arm. The purpose of the arm is to interrogate each bit position to determine its height, and therefore its digital value. The arm RCX runs the control code for the entire system, and is responsible for receiving optical communication from the circuit and sending commands to the table. The data scanning algorithm is a major component of the arm code. The arm design is constrained to comprise LEGO construction parts, two light sensors (one each to communicate with the circuit and the table), a rotation sensor, a touch sensor, and a motor. Students are guided to a design in which the touch sensor detects contact with the table LEGO stacks, and the rotation sensor measures the motion of the arm or tip, and therefore the height of the stacks. The RCX datalog keeps track of the position and height of each data bit, and after the scan is uploaded to a provided MATLAB routine that plots a map of the digital data (Fig. 3). The map is part of the demonstration grade.

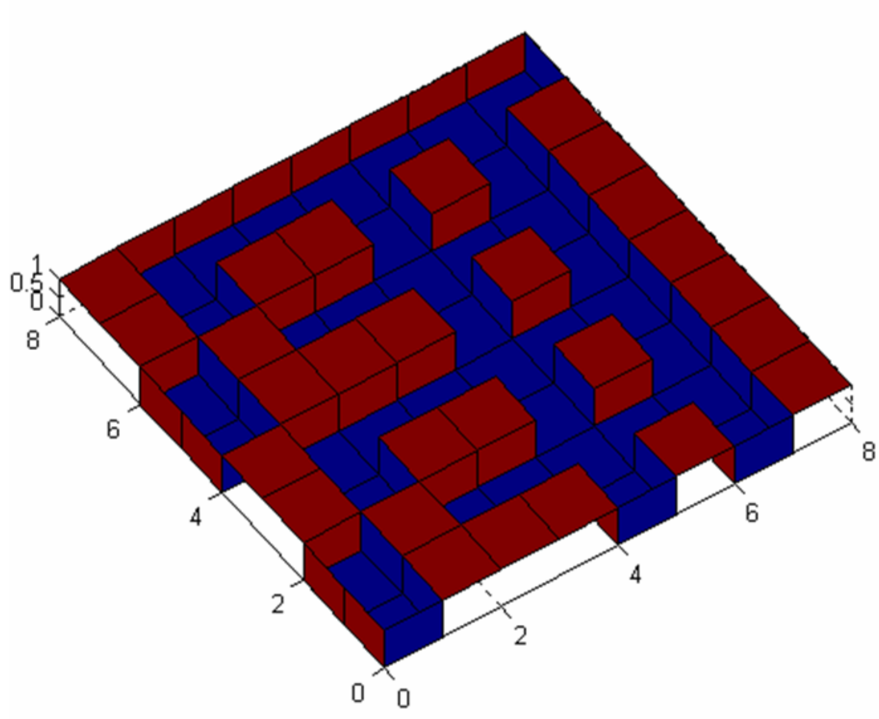

Figure 3: Data map representing the LEGO block stacks on the X-Y table

Students saved the rotation sensor values to the RCX data log and downloaded it to a MATLAB routine provided to them that generated the map. This map was used as verification of proper operation of the models during the final demonstrations.

3.2 Learning Centers

Students are guided through the entire design process by materials presented in lectures, homeworks and LCs. The five LC activities are to:

Week 1:

1. Become familiar with the RCX computer interface
2. Become familiar with the RCX brick functions
3. Learn how to control motors from the RCX
4. Learn how to interface sensors to the RCX
5. Learn simple variable and function routines in NQC
6. Learn about the RCX datalog

Week 2:

1. Be introduced to the Millipede table and its optical encoder mechanism
2. Practice using functions in NQC
3. Learn to work in teams to reach a design consensus
4. Experience the interplay of various types of gears to develop a simple mechanical system
5. Learn to optimize a system under various practical constraints

Week 3:

1. Understand the difference between series and parallel circuits
2. Learn basic use of a digital multimeter
3. Learn how to construct a simple circuit
4. Understand the concept of RC time constants
5. Become aware of some applications of simple timer circuits

Week 4:
1. Become familiar with the digital trainer
2. Solidify concepts in bi-phase coding
3. Learn how to construct a shift register circuit

Week 5:
1. Gain practice in basic programming
2. Work in teams to complete a project
3. Gain practice in implementing bi-phase coding

Week 6: Demonstration of final designs

4. SUMMARY OF STUDENT FEEDBACK

At the conclusion of the project, students were asked during class to provide feedback by completing a formal written survey that covered the entire semester. Also, by using handheld electronic units, they responded to questions on various activities specifically related to the learning module. These results, along with the results from previous surveys, will be used as part of the assessment process for the module and the course and will aid in the identification of improvements. This section summarizes student responses.

Students were asked to identify the project activities liked the most and the least. Student responses to these questions were collected by electronic hand-held units. These data were summarized and presented to the students. A summary of the responses is shown in the following table:

Table 2: Student Responses - Activities

Student activity	**Which activity did you like the**	
	Most	**Least**
Building the arm	38%	5%
Wiring circuits – electronics	16%	13%
Programming the arm	14%	49%
Putting it all together	10%	11%
I liked them all/I didn't like any of them	10%	9%

It is observed that the response percentages were not equally distributed among the response choices, with building the arm identified as the most-liked activity and programming the arm identified as the least-liked activity. Nearly half the class identified programming the arm as the least-liked activity with nearly the same percentages of students identifying wiring the circuit and putting it all together as the next least-liked activities.

When asked in the survey to evaluate the module on the amount of learning, level of effort and interest level, the students responded as shown in the following table.

Table 3: Student Responses - Learning

	Amount of Learning	Amount of Effort	Interest Level
Very low	4%	0%	8%
Low	6%	3%	11%
Medium	20%	12%	20%
High	39%	30%	31%
Very high	31%	55%	30%

The student responses provide evidence of a successful first deployment for the module and that no major changes to the module are required. However, student responses suggest opportunities to further improve the module.

REFERENCES

Brockman, Jay B., Fuja, Thomas E., and Batill, Stephen M. (2002). A Multidisciplinary Course Sequence for First-Year Engineering Students. *2002 American Society for Engineering Education Annual Conference and Exposition*, **2253**.

Centeno, Grisselle, Clayton, LaNetra, Otero, Luis Daniel, and Zekri, Souheil (2004). Innovative Modules To Introduce Advanced Science And Engineering Concepts. *34th ASEE/IEEE Frontiers in Education Conference*, **F3D**, 12-17.

Crone, Wendy C., Carpick, Robert W., Lux, Kenneth W., and Johnson, Buck D. (2003) A Course in Micro- and Nanoscale Mechanics. *American Society for Engineering Education Annual Conference & Exposition*, **XX68**.

Lakhtakia, Akhlesh (2006) Priming Pre-University Education for Nanotechnology. *Current Science*. **90**, (1), 37-40.

McWilliams, Leo H., Silliman, Stephen E., and Pieronek, Catherine (2004). Modifications to a Freshman Engineering Course Based on Student Feedback. *2004 American Society for Engineering Education Annual Conference & Exposition*, **3353**.

Pieronek, Catherine, McWilliams, Leo H., Silliman, Stephen E., and Uhran, John J. (2005). The Value of a Comprehensive, Integrated Assessment Program for Improving the First-Year Engineering Experience. *35th ASEE/IEEE Frontiers in Education Conference*, **T3H**, 1-6.

Uddin, Mahbub and Chowdhury, A. Raj (2001). Integration of Nanotechnology into the Undergraduate Engineering Curriculum. *International Conference on Engineering Education*, **8B2**, 6-9.

Uhran, John J. Jr., Batill, Stephen M., and Silliman, Stephen E. (2005). A Curriculum Renovation Revisited. *35th ASEE/IEEE Frontiers Education Conference*, **F1D**, 10-15.

Vettiger, P., Despont M., Drechsler, U., Durig, U., Haberle, W., Lutwyche, M. I., Rothuizen, H. E., Stutz, R., Widmer, R., and Binnig, G. K. (2000). The "Millipede" – More than one thousand tips for future AFM data storage. *IBM Journal of Research Development*, **44**, (3), 323-340.