Session ____

Meeting the Needs of Industry: Development of a Microcontroller Course for Mechanical Engineers

William G. Culbreth
University of Nevada, Las Vegas

Abstract

A course was developed at the UNLV to address the needs of local industry. The course was designed to acquaint mechanical engineering students with the principles and use of microcontrollers for measurement and control.

Hastings' Chariots, a Las Vegas-based rocket company, expressed a need for mechanical engineers with knowledge in integrating sensors with microcontrollers used to control microgravity experiments. A biomedical company in Las Vegas also indicated the need for graduates trained in the use of this same microcontroller. The microcontroller employed by both companies was also used on a research project at the university in assessing radionuclide concentrations in pipes. This system is shown in figure 1.

Although mechanical engineering students have coursework in measurements, controls, and Fortran or C programming, they have little familiarity with microcontrollers. In response to the needs of our research program and local industry in Las Vegas, the mechanical engineering department developed a microcontroller course based on the BASIC Stamp II manufactured by Parallax, Inc. In lieu of a textbook, students in the course purchased a microcontroller kit that contained an extensive manual, PC connection cable, and a processor with motherboard. The course included instruction in BASIC programming, data acquisition, sensors, and computer communication. Numerical methods useful for analyzing data acquired by the microcontroller were also covered.

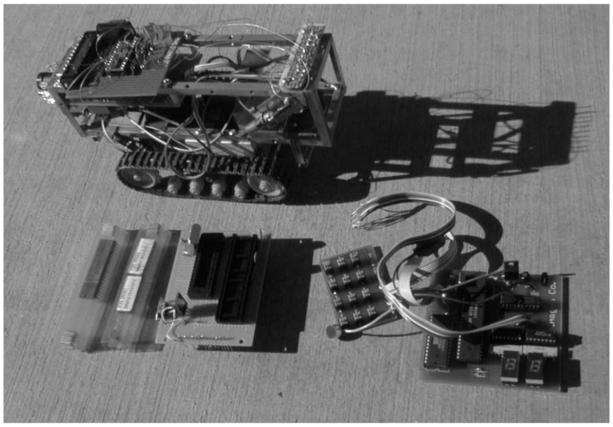


Figure 1. Microcontroller used in Autonomous Device for Radionuclide Assessment in Pipes

Students completed two design projects in the course. Each student designed and constructed a weather station that integrated several sensors with the BASIC

Stamp2. Data was transferred to a PC for analysis and display of the results. This initial project was followed up by individual projects that included the design of robotic prosthetic devices, a temperature-controlled cooling fan, and a robot to investigate the interior of pipes. The graduates of the course have used their experience to gain employment and to complete senior design projects using microcontrollers.

Mechanical engineering undergraduates are typically required to take computer programming, electrical circuitry, and mechanical measurements as a part of their curriculum. There is an increasing demand for mechanical engineers to work with microcontrollers for automated data acquisition and control. A microcontroller applications course was developed at the University of Nevada, Las Vegas, to meet this need.

I. Introduction

Engineering curricula change with the needs of local and regional industry. During the past several years, the mechanical engineering department at the University of Nevada, Las Vegas has been contacted by local companies and government agencies to assist in the development of products that require the integration of sensors and actuators with microcontrollers. These projects are often ideal for senior design, however, undergraduate mechanical engineering students often have difficulty in working with these devices.

To help our students better meet the needs of local industry, a mechanical engineering course was developed to teach students to use microcontrollers, to integrate sensors and actuators with these devices, and to connect to larger computers for communication with the user. The course was taught in the spring semester of 2000 to a class of undergraduates and master's students.

In addition to this course in microcontrollers, computer technology is integrated into a number of our undergraduate mechanical engineering courses. The introductory engineering course has a companion laboratory course[1] that introduces Microsoft Office, Mathsoft Mathcad, and Autodesk Autocad. They are also required to take a computer science course in Fortran or C++. A junior-level mechanical measurements course[2,3] introduces sensors and computer-aided data acquisition. Students taking the microcontroller course received instruction in these prerequisite courses.

II. Course Objectives

The objectives of the mirocontroller course for mechanical engineering students focused on several areas:

- Instruction in microcontroller programming
- Review of sensors and circuits
- Communication between microcontrollers, sensors, and host computers
- Computational analysis of data

An outline of the course content is included in Appendix A.

The course was intended to be “hands on” and several microcontroller systems were considered. Mechanical engineering students take a course in Fortran or C++ at the sophomore level, but have no experience in assembly language. We sought a system that was inexpensive and easy to program in a higher-level language.

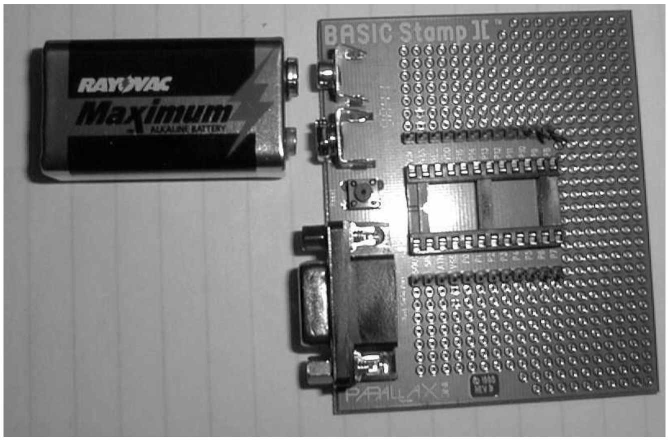


Figure 2. Microcontroller Board, Battery, and Serial Port Connection

The Parallax, Inc. BASIC stamp[4] series used an integer version of BASIC and a complete system was available to the students for roughly the price of a standard textbook. We chose the BASIC Stamp II due to its enhanced instruction set and moderate price. The system included a small motherboard, as shown in figure 2, a PIC processor with the BASIC interpreter, a well-written manual with examples, a serial cable, and a disk containing a PC-based editing program. The students ordered their microcontroller systems over the internet and received them within the first two weeks of the course.

The BASIC Stamp II contained a serial interface for connection to a PC, 16 input/output lines, and 4 kilobytes of RAM used for program and data storage. Measuring 60 mm by 70 mm, the boards were powered by a single 9 volt battery. Connection to a standard personal computer turned out to be very easy.

Individual Weather Station Projects

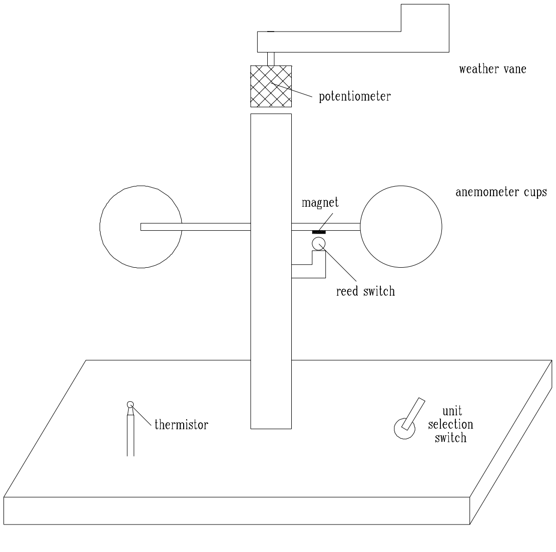


Figure 3. Weather Station Development

The first class project using the Stamp II involved the design and construction of a weather station. Each student was issued a thermistor, a multiturn, wirewound potentiometer, a single pole, single throw switch, a reed switch, and a small magnet. These items were assembled by the students into weather stations of their own design as outlined in figure 3.

The thermistor provided ambient

temperature through the microcontroller's RCTIME command as shown in figure 4. To read the sensor, the relevant pin is raised HIGH to 5 volts for several milliseconds. During the execution of the RCTIME command, this pin is brought to ground and the number of milliseconds required to dissipate the voltage from capacitor, C, through the thermistor is returned as a variable in BASIC. The discharge time for the thermistor is given by:

$$\tau = 1/RC$$

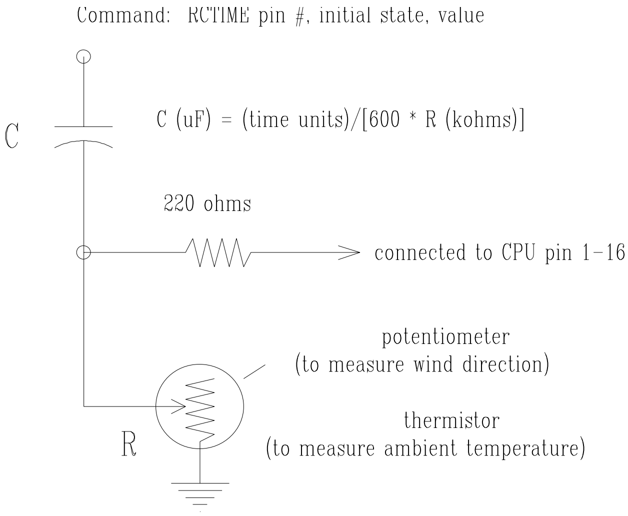

Command: RCTIME pin #, initial state, value
C
C (uF) = (time units)/[600 * R (kohms)]
220 ohms
connected to CPU pin 1-16
potentiometer
(to measure wind direction)
thermistor
(to measure ambient temperature)
R

Figure 4. Interface for Potentiometer or Thermistor Input

The students were required to calibrate their thermistor by plotting the resistance versus temperature as shown in figure 5. A capacitor was then chosen to yield the maximum allowable count (32,768) at a temperature of 0°C.

The wind direction was measured with a multiturn potentiometer through a similar circuit and command. A wirewound potentiometer was used since they require very little torque to turn.

MEG 495.2
Microcontrollers

Performance of Thermistor K-788
Fenwal Electronics, Inc.

T (F)	uncertainty	R (kohm)	uncertainty
73	0.1	1541	0.1
94.1	0.1	910	0.1
150	0.1	240.8	0.1
185	0.1	114.7	0.1
220	0.1	58.42	0.01
250	0.1	34.23	0.01

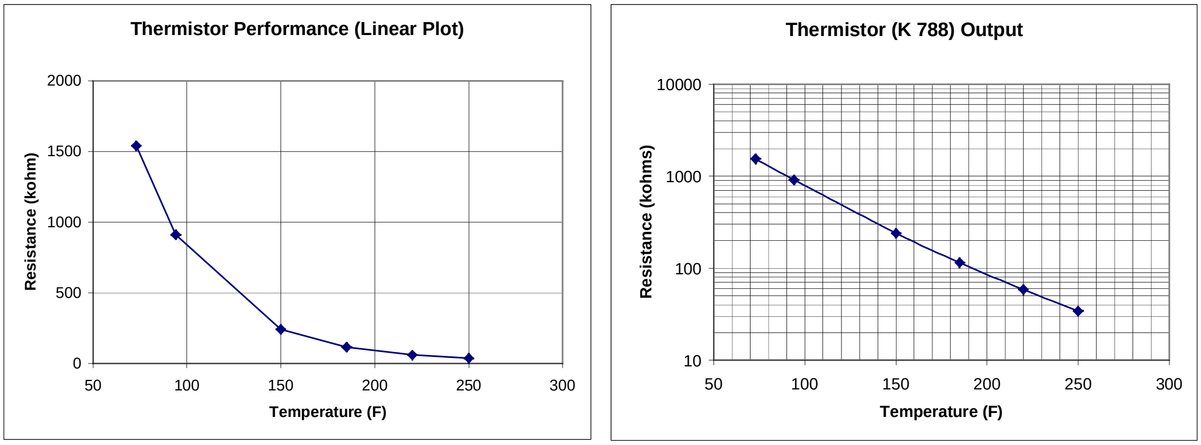


Figure 5. Thermistor Calibration

Wind speed was measured through a reed switch attached to a stationary platform beneath the anemometer. A magnet attached to the rotating cups alternatively opened and closed the reed switch at a rate proportional to the wind speed. The Stamp II's COUNT command easily provided the number of counts within a fixed period of time. Several students calibrated their anemometers by using their automobiles. They recorded the anemometer output signal at several vehicle speeds to create a first-order calibration curve. An example of a student weather station is shown in figure 6.

The Stamp II does not have its own display device, although inexpensive LCD displays are available. The students were required to connect their microcontrollers to a PC for data analysis and presentation. A switch located on the weather station was used to tell the PC, through the microcontroller, whether to display the results in MKS or English units. The students developed a PC program in Microsoft QuickBasic[5] to display the wind velocity, direction, and temperature in appropriate units. A simple QuickBasic program that allows the Stamp II to communicate with a standard PC is included in Appendix B.

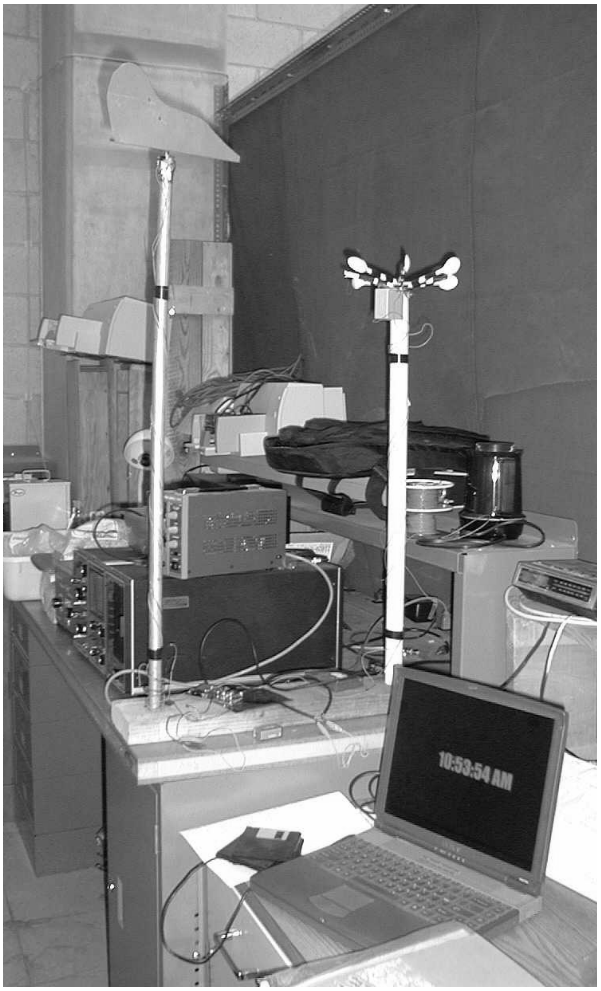


Figure 6 Student Weather Station (Vane, Anemometer, Thermistor, and Laptop Connection)

The weather station project exercised many of the STAMP II commands and provided practical instruction in connecting microcontrollers to PC's and assorted sensors.

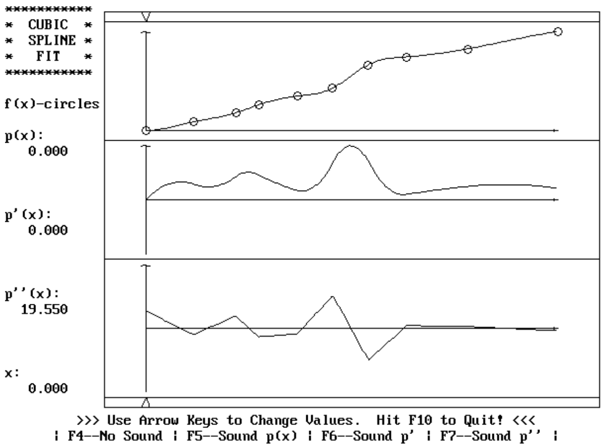
Individual projects were next assigned based on the desires of the students. Each student completed their own design project, although the small class size allowed for a good exchange of ideas between students.

III. Numerical Methods

Instruction on the usage of microcontrollers concluded with a discussion of mathematical methods for the analysis of data. This included least squares curvefitting to help in the calibration of sensors, cubic spline analysis to obtain derivatives or integrals of measured values, and Fourier analysis to assess periodicity in sensor readings.

Several computational tools were introduced through this course to analyze the data acquired by the microcontrollers.

- Statistical properties of data.
- Least squares curvefitting.



```
************
*  CUBIC   *
*  SPLINE  *
*   FIT    *
************

f(x)-circles

p(x):
   0.000

p'(x):
   0.000

p''(x):
  19.550

x:
   0.000
      >>> Use Arrow Keys to Change Values.  Hit F10 to Quit! <<<
  ¦ F4--No Sound ¦ F5--Sound p(x) ¦ F6--Sound p' ¦ F7--Sound p'' ¦
```

Figure 7. Cubic Spline Analysis of Acquired Data

- Cubic spline fitting of data.
- Fourier analysis.

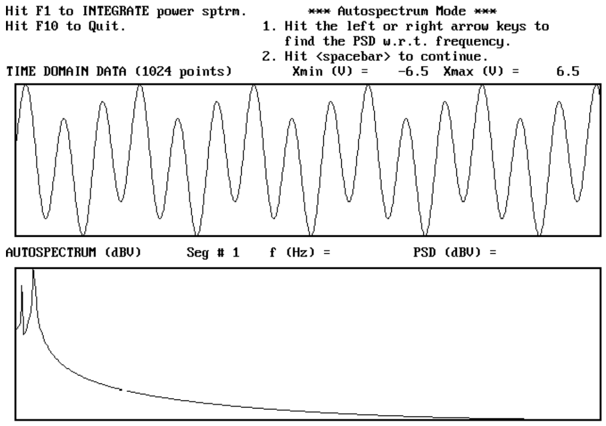
These concepts were presented in conjunction with the work on their weather stations. A discussion of statistical analysis, for example, accompanied our work on acquiring temperatures from the thermocouple and average wind speed from the anemometer. Homework assignments were given that required the students to write a program to read a data file from disk, compute the mean and standard deviation of temperature, and present the results on the screen in engineering units.

The thermistor, anemometer, and wind direction meter all required calibration. The least squares method was discussed to demonstrate that tabulated calibration data could be reduced to simple functions in a computer program. Homework assignments were used to encourage the students to write programs in BASIC to curvefit calibration data using linear and exponential fits. These programs were then used by each student to calibrate their own sensors for their projects.

The microcontrollers could also be used to acquire data from strain gages or accelerometers. To accommodate these sensors, cubic spline fitting was explained. The students were given a cubic spline program written in QBASIC to help with homework and laboratory exercises. Sample output is shown in figure 7.

For a sensor output x(t), cubic splines provide continuous functions for x'(t) and x''(t). Strain gages, for example, can be used to measure velocity and acceleration through differentiation of the time-dependent strain gage output. Numerical integration of the functions produced by a cubic spline fit can yield velocity and displacement from and accelerometer signal.

The students were given a Fourier analysis program (figure 8) written in QuickBasic to convert time-dependent signals to frequency domain. A large data set obtained from a hot wire anemometer was used in a laboratory exercise to demonstrate the use of Fourier analysis.



```
Hit F1 to INTEGRATE power sptrm.         *** Autospectrum Mode ***
Hit F10 to Quit.                     1. Hit the left or right arrow keys to
                                        find the PSD w.r.t. frequency.
                                     2. Hit <spacebar> to continue.
TIME DOMAIN DATA (1024 points)          Xmin (V) =    -6.5  Xmax (V) =     6.5

AUTOSPECTRUM (dBV)       Seg # 1    f (Hz) =           PSD (dBV) =
```

Figure 8. Fourier Analysis of Acquired Data

IV. Results

The students completed individual projects based on the BASIC Stamp II micrcontroller and their individual interests. A list of these projects includes:

- Pipe crawler using a “slinkee” and solenoids for radionuclide assessment in pipes.
- Design of a prosthetic, actuated foot using shape memory alloy actuators.
- Design of a cooling system for electronics with a stepped fan speed based on temperature.
- Design of a prosthetic, actuated knee.
- Shape memory alloy actuation of a prosthetic finger.
- Sound-following system for an entertainment engineering device.

V. Conclusions

Mechanical engineering students benefited by an introduction to the use of microcontrollers with project-based instruction. The course removed some of the anxiety that these students expressed with regards to computer programming. The BASIC language still provides a simple interface to complex programming. Although all students had a formal course in Fortran, they readily acquired programming skill in the BASIC language.

The students in the course most enjoyed working on their individual weather station projects. Each student had a very different design ranging from devices the size of a tissue box to 1 meter high towers. The topics of computer programming and computational analysis were embraced by the students as they applied these techniques to their own designs. There was quite a bit of camaraderie and shared information throughout the project design and construction.

Bibliography

1. Culbreth, W. G., and Rivera, M., Application Programs in the Engineering Curricula: The Freshman Experience, Ninth ICMCM, Berkeley, CA, (1993).
2. Culbreth, W. G., Computer-Aided Data Acquisition and Instruction, Proceedings of the 1984 ASEE-PSW Conference on Computer Aided Processes in Instruction and Research, Tempe, Arizona, pp. 23-37, (1984).
3. Culbreth, W. G., Computer-Aided Data Acquisition in Mechanical Engineering, Proceedings of the 1985 ASME International Computers in Engineering Conference, Boston Mass., v 2, pp. 167-172, (1985).
4. Parallax, Inc., BASIC Stamp Programming Manual, Version 1.9, Rocklin, California.
5. Culbreth, W., Programming in BASIC QuickBasic for Engineers, University of Nevada, Las Vegas, January, (2000).

WILLIAM CULBRETH
William Culbreth is an Associate Professor of Mechanical Engineering at the University of Nevada, Las Vegas. He received his B.S. degree in Physics from the California State Polytechnic University, and his M.S. in Nuclear Engineering and Ph.D. in Mechanical Engineering from the University of California, Santa Barbara in 1981.

Appendix A

A. Introduction
   1. How are microcontrollers (µC) used?
   2. Overview of data acquisition
   3. Overview of µC control.
   4. Ordering information for the BASIC Stamp.
   5. Microsoft QuickBasic.

B. Data Acquisition
   1. Measuring counts per second
   2. Digital to analog converters
   3. Analog to digital converters

C. Principles of Sensors
   1. Simple sensor – the wire
   2. Wheatstone bridge and voltage dividers
   3. Switches and potentiometers as input devices
   4. Operational amplifiers
   5. Power supplies
   6. Sensors for temperature, pressure, and acceleration

D. Actuators
   1. Standard Output Devices
   2. Solenoids
   3. Motors
      a. DC motors
      b. Stepper motors
   4. Shape memory alloy actuators

E. Principles of Microcontrollers
   1. Microcontroller architecture
   2. Microcontroller architecture
      a. Central processing unit (CPU)
      b. Clock
      c. Address and memory bus
      d. Memory
      e. Communication
      f. RS-232
      g. TTL interfaces
   3. Input/Output Pins
   4. Connection to switches
   5. Programming in Integer BASIC
   6. Connection to output – LED
   7. Connection to DC Motor
   8. Input from a sensor with feedback

F. Advanced Programming
   1. Least squares method
   2. Cubic spline fitting
   3. Fourier analysis

G. Individual Projects

Appendix B

```
’-----------------------------------------------------------------------
’
’  Program:  Serial
’
’  Purpose:  Communicate with a BASIC Stamp through a
’            serial interface.  This program is a template
’            that you can expand upon to make your own
’            program.  The variable a$ contains the text
’            read from the BASIC Stamp.
’
’-----------------------------------------------------------------------

 CLS
 OPEN "COM1:9600,N,8,1,asc,cs,ds,cd" FOR INPUT AS #1

 COM(1) ON
 ON COM(1) GOSUB Com1Handler

 DO
 LOOP WHILE INKEY$ = ""
 COM(1) OFF
 CLS
 END



Com1Handler:
  LINE INPUT #1, a$
  PRINT a$

 'This is where you would add your program to compute ‘the temperature, wind direction, and
 wind speed.

  RETURN
```