AC 2008-1513: THE UBIQUITOUS MICROCONTROLLER IN MECHANICAL ENGINEERING: MEASUREMENT SYSTEMS

Michael Holden, California Maritime Academy

Michael Holden teaches in the department of Mechanical Engineering at the California State University Maritime Academy.

The Ubiquitous Microcontroller in Mechanical Engineering: Measurement Systems

Introduction

This paper will describe a project aimed at integrating microcontrollers in several classes throughout the mechanical engineering curriculum at the California State University Maritime Academy (CMA). The goal is to give our students useful skills that will help distinguish them from other job candidates. Microcontroller technology is new enough that recent graduates can successfully compete with more senior engineers who have never learned to design with microcontrollers.

Microcontrollers are becoming ubiquitous in many modern products and machinery, due to their ability to perform complex electronic functions for low cost, and understanding how to use these systems is a valuable skill set for any engineer. While some projects will require an electrical engineer to implement the microcontroller design, having the ability to design simple microcontroller systems will give a mechanical engineer the ability to be responsible for the entire design of many mechatronic devices, as inexpensive microcontrollers replace discrete electronic component designs. For example, a microcontroller may be used to read an analog sensor and control an output display or actuator, a simple task ideally suited to an inexpensive microcontroller, and one that can be implemented with only basic microcontroller experience.

The goal of the project is to design microcontroller hardware, software, and courseware that will be used in several classes in the ME curriculum, so that the students gain familiarity with common microcontroller systems and applications without taking a special elective. The hardware design must be useful for classes and laboratories including programming, electronic circuits, measurement systems, control systems and mechatronics. Finding a common platform to use in many classes allows the instruction time devoted to microcontrollers to be distributed so that the new topic can be added without cutting significantly into the existing curriculum.

Rather than taking one microcontroller class near their senior year, the students will be exposed to microcontrollers as early as their freshman or sophomore years, and will gain experience with the same hardware in several classes. There are several advantages to this approach compared to adding a standalone technical elective to the curriculum.

The primary advantage is that students learn the material early in their education and have a developed skill set ready to apply to capstone design projects. There is not much prerequisite knowledge needed to learn microcontrollers (such as calculus or dynamics). Assuming that microcontroller programming (programming in C) will be taught as part of the microcontroller curriculum, only basic computer skills are needed from the students. Most incoming students have the knowledge to get started in microcontrollers.

Another advantage is that the students learn the skills without adding classes to the curriculum. At CMA, as in most engineering programs, the student course load is at a maximum and to add material requires removing other material. There simply isn't room in the curriculum for another class, so either electives must be traded or the microcontroller skills must be taught in existing classes.

The disadvantage to the integrated approach is that more faculty must "buy in" to the program and be able to teach microcontroller applications, and that some material in each course will be squeezed out in order to teach microcontrollers. This paper will measure student perceptions of the approach taken in order to measure whether the advantages outweigh the disadvantages.

Method

As a starting point, an inexpensive microcontroller system is being developed with the following specifications:

- Cost less than $75 per lab station, and less than $25 per student unit.
- Be used to teach programming in a traditional programming class
- Be used for laboratories in an electronic circuits course
- Be used as a measurement device (DAQ) with Labview as well as a standalone data logger in a measurement systems course.
- Be the cornerstone of mechatronics laboratory robotics work.

There are many systems that could satisfy these requirements. For example, the Basic Stamp [1] is a popular microcontroller for educational uses. Simple systems like the Atmel Butterfly [2], GumStix [3] computers, and the Brainstem controller [4] would all work, although some can be more costly than specified. But it is so easy to design a new microcontroller system that a custom circuit board is also an attractive solution. A circuit board to support a microcontroller does not need many components and is inexpensive and easy to assemble.

A microcontroller experiment board that was inspirational to the author is that developed by Hugh Jack and Jeff Roberts at Grand Valley State University [5]. Shown in Figure 1, it is USB powered and brings the bulk of the IO pins to a useful header, while maintaining a form factor that is compact for many creative applications.

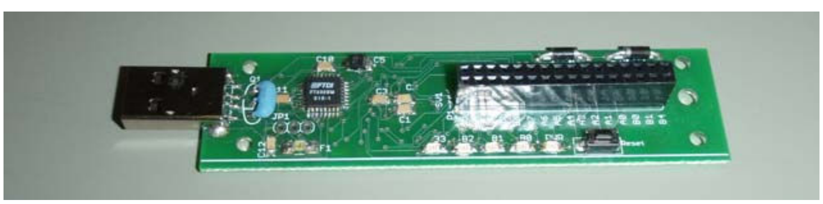

Figure 1: Grand Valley State University Microcontroller Board

The author has also created similar designs, such as the one in Figure 2. This was an autopilot board for a GPS-guided model car [6] that contains a serial interface for a GPS and a user interface consisting of an LCD display and some buttons.

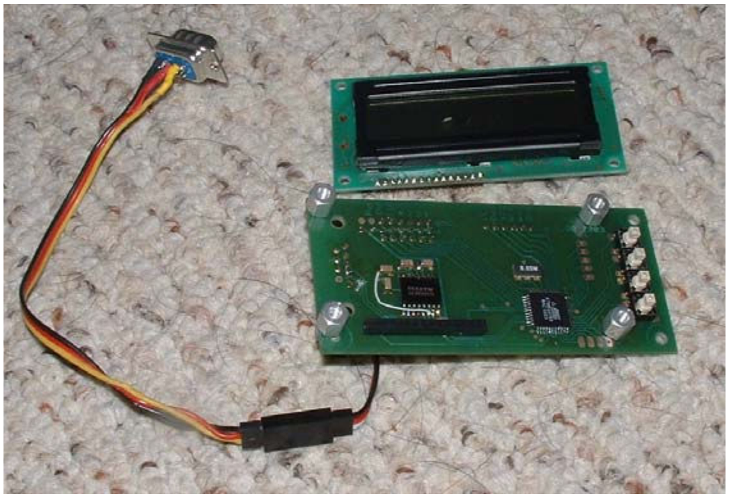

Figure 2: Microcontroller Autopilot Developed by the Author

This paper describes the first attempt at integrating microcontrollers in a class devoted to other topics. As an initial step, it was decided to use a very crude form of microcontroller board. For the results given in this paper, the microcontroller system was assembled using a proto-board. The circuit is simple enough that it is not very difficult to build on a proto-board, and this assembly technique shows the students exactly how they could put a system together. Figure 3 shows the proto-board system. Students who build this system in lab will not need any assistance to assemble their own units after they graduate or for a senior project. The drawbacks to a proto-board setup is that the system is somewhat fragile and bulkier than if a dedicated board was designed and soldered together.

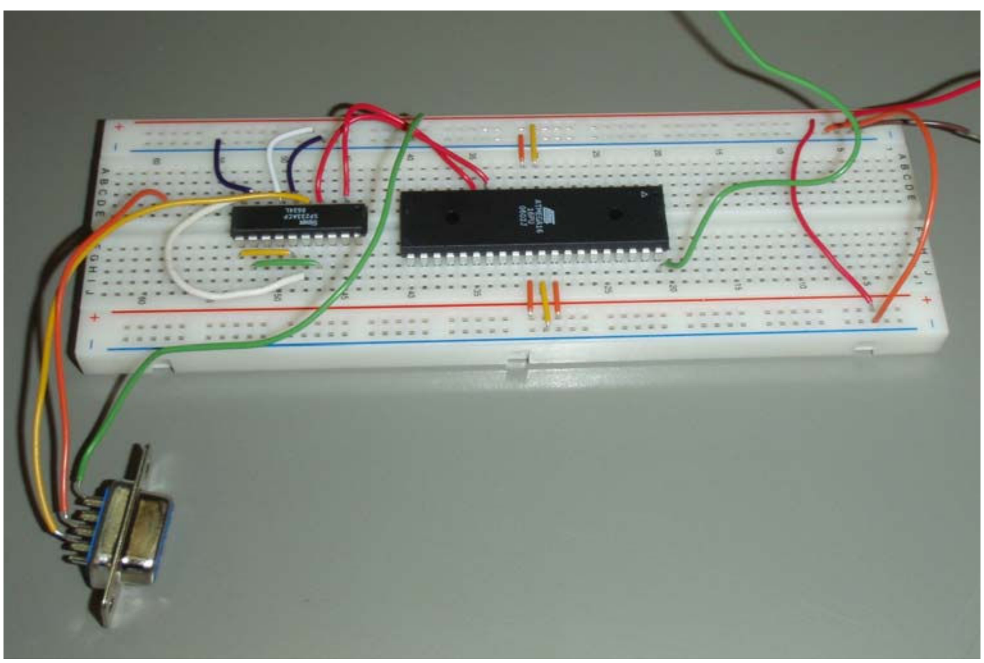

Figure 3: Protoboard Microcontroller DAQ

The system is comprised of an Atmel Mega 16 microcontroller, with a MAX-233 RS-232 converter chip (or a 2N2222 Transistor for the most simple version), and a DB-9 serial connector. In the laboratory 5V power supplies are used to power the chip, a USB programmer is used to program the chips, and a serial port extension cable is used to connect to the PCs in the lab. A schematic of the system is shown in Figure 5. The schematic shown uses a 2N2222 transistor as an inverter instead of the MAX-233 chip; this does not meet the RS-232 standard but nearly always works for serial transmission. The system as used is able to read 8 analog inputs (0 to 5V with 10 bit precision) and send the data over the serial port to Labview, thereby creating an inexpensive DAQ device.

The microcontroller software is very simple. The main infinite loop is shown in Figure 4: it consists of just 10 lines of code. Each of the 8 A/D channels is sampled in turn, the reading is converted to Volts, and the measurement is printed to the serial port (the Codevision compiler [7] maps the standard i/o device to the UART).

```

chan = 0; // make sure we are reading channel 0
while (1)
   {
      adint = read_adc(chan);              // read the A/D channel
      mVolts = ((float)adint)*5000/1023; // convert 10 bit A/D to Volts
      printf("%d,",mVolts);                // print Volts to serial port
      chan++;                              // next channel
      if (chan>7) { // wrap around channel index
        chan = 0;
        printf("\n\r"); // end line after all 8 channels
        delay_ms(250);  // not so fast
        ii++;           // counter for debugging purposes
      }
   };
```

Figure 4: DAQ Microcontroller Code

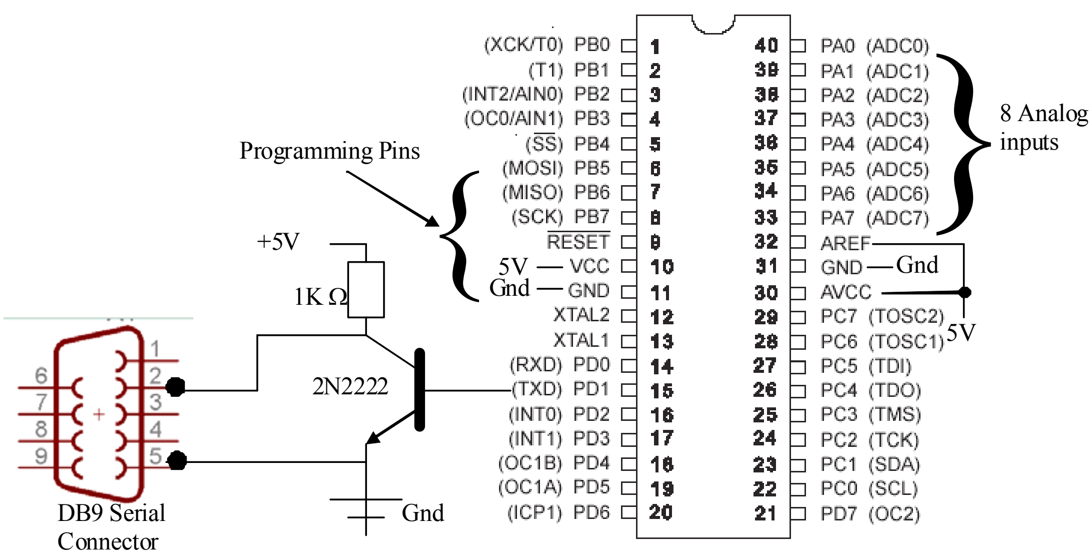

Figure 5: Microcontroller DAQ Schematic

Measurement Systems Class

The Measurement Systems course at the California State University Maritime Academy includes a laboratory to give the students hands-on experience with sensors, signal-conditioning circuits, and data acquisition systems. The lab has traditionally used Labview [8] to acquire data on a PC. The National Instruments data acquisition devices range from internal cards with high speed and precision to inexpensive USB units.

The goal of adding microcontrollers to the Measurement Systems course was to create a data acquisition device of our own. This would allow the students to leverage their knowledge of Labview graphical interfaces while using the analog-digital converters and serial port on the microcontroller to measure data.

To gather the data in Labview, the Instrument I/O Assistant was used to read and decode the string from the serial port on the PC. The assistant makes parsing the data simple although it sometimes has a mind of its own. The block diagram for the Labview data acquisition process is shown in Figure 6. The data is displayed according to the desires and needs of the student; a simple front panel that plots one channel and displays the others is shown in Figure 7.

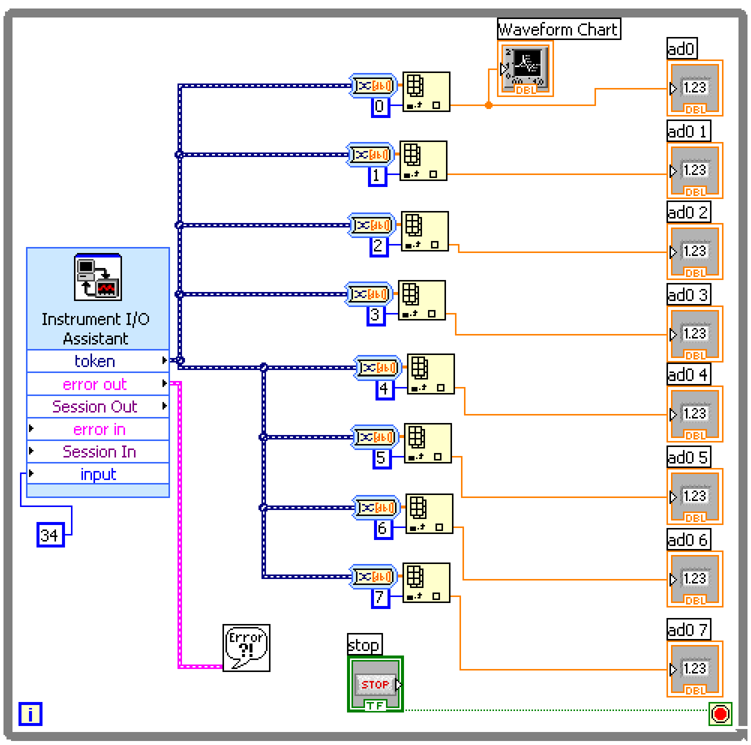

Figure 6: Labview DAQ Block Diagram

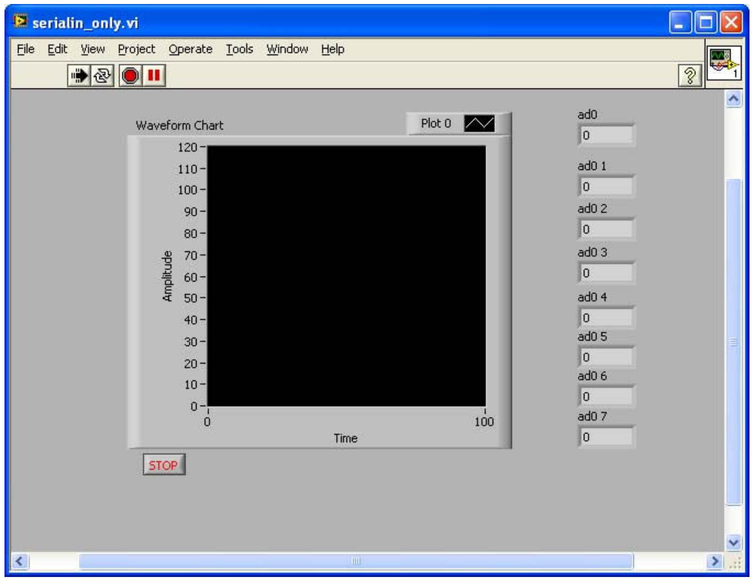

Figure 7: Labview DAQ Front Panel

This very basic system is a starting point for data acquisition using a microcontroller. The system could be modified to suit different projects by changing the displays, or scaling the data in the microcontroller to reflect the units of the sensor being measured. The microcontroller could also store the data as it is measured, and dump it into labview later, for self contained data logging. The freedom to modify the software to suit the project is one of the biggest advantages to using a microcontroller over a traditional DAQ unit.

The DAQ system compares quite favorably to the commercial units, especially considering the parts cost (without the breadboard) is less than $20. A printed circuit board for this design would be small and likely to cost less than $10, so the system cost would be reasonable if assembled into a dedicated device.

The data is logged with less precision (10 bits vs. 12 or 16 on the National Instruments DAQ devices), and the data is logged slower (4 Hz vs. up to 1000 Hz for the more expensive NI units), but the speed and accuracy are sufficient for many applications. For example, all but 2 of the 12 labs performed by the students in the Measurement Systems lab using NI devices could have been run using the microcontroller-based DAQ without modification; the exceptions need a fast sample rate for FFT analysis of sound signals.

The software to create the DAQ using an Atmel microcontroller as well as the Labview serial acquisition VI are available from the author's web site: www.holdentechnology.com.

<u>Course Details</u>
The introduction of microcontrollers did require that the format of the course change somewhat. Approximately 2 weeks of lecture time was devoted to introducing microcontrollers, in particular teaching the students to read C code, as the students learn Matlab in the required programming class. Enough of the elements of C were covered in lecture so that the students could hack existing code and understand the general structure of a C program for microcontrollers. The

remaining lecture time was devoted to the particulars of setting up and using microcontroller chips. Plenty of example code is made available to give students a starting point.

Three additional labs were added to the laboratory portion of the course. These labs consisted of an introduction to the compiler and simulator, where the students programmed the chips to blink an LED; a demo program that exercised the various on-chip peripherals such as the A/D and UART; and finally the DAQ application described in detail above.

<u>Assessment</u>
The students in the labs were asked to complete a survey to assess their perceptions of the value of the microcontroller portion of the class. The questions attempted to measure the value of the material to the students, the amount of time that should be devoted to microcontrollers in a Measurement Systems class, and the skills the students hope to use after they graduate. The students were asked to rate their level of agreement or disagreement with statements on the survey.

The first statement, "Microcontroller data acquisition seems appropriate in a measurement systems class", shows that the students clearly believe the material is relevant and fits with the other material in the class.

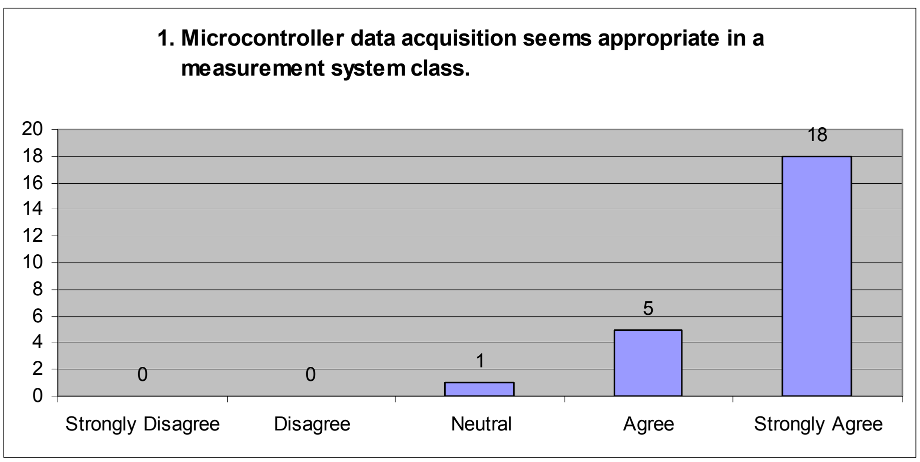

Figure 8: Assesment 1

The second statement, "I would rather have spent more class time on sensors and signal analysis and less time on microcontrollers and programming", received a mixed response, with a negative majority. This is interpreted as supporting the addition of the material in the class, but it is worthwhile to note that some students would have preferred less time spent on the new material.

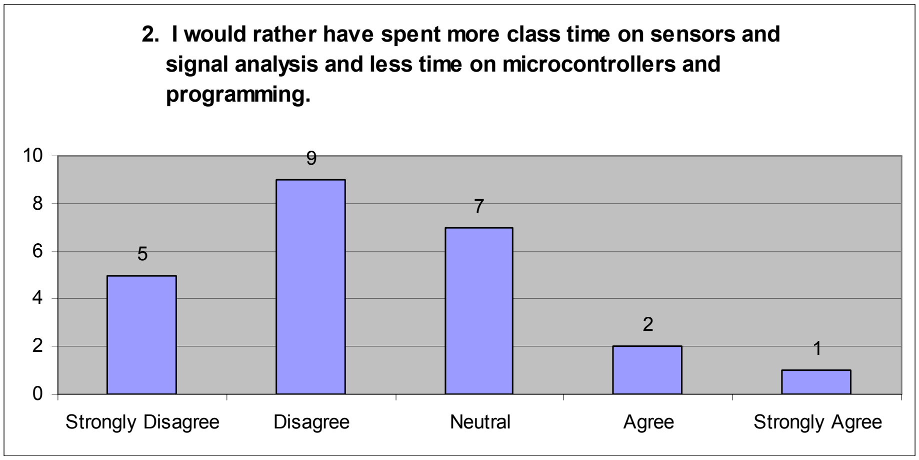

Figure 9: Assessment 2

The third statement, "More time should have been spent explaining how microcontrollers work" reinforces the conclusion of the previous statement. Most of the students would have liked more time spent on microcontrollers, with a few who disagree.

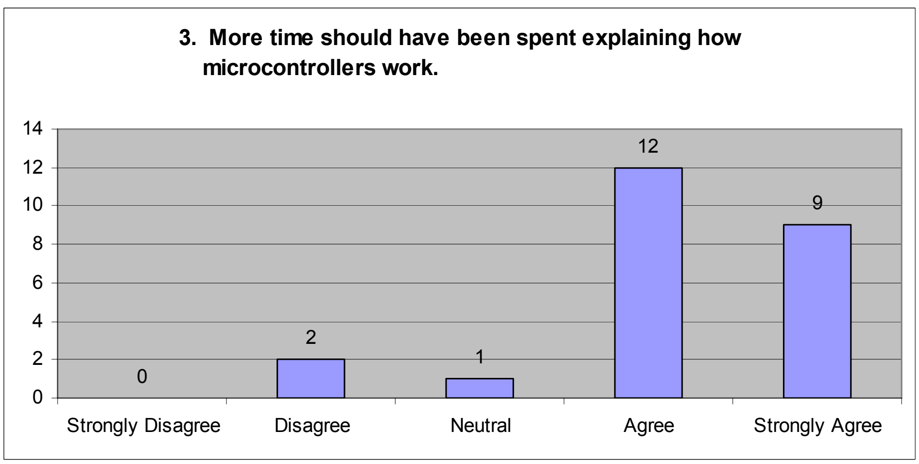

Figure 10: Assessment 3

The fourth statement, "I hope to use the microcontroller skills from this class in my career", received all positive or neutral responses. This shows that the students see how the microcontroller skills could help their careers, and that they would be agreeable to working in this industry.

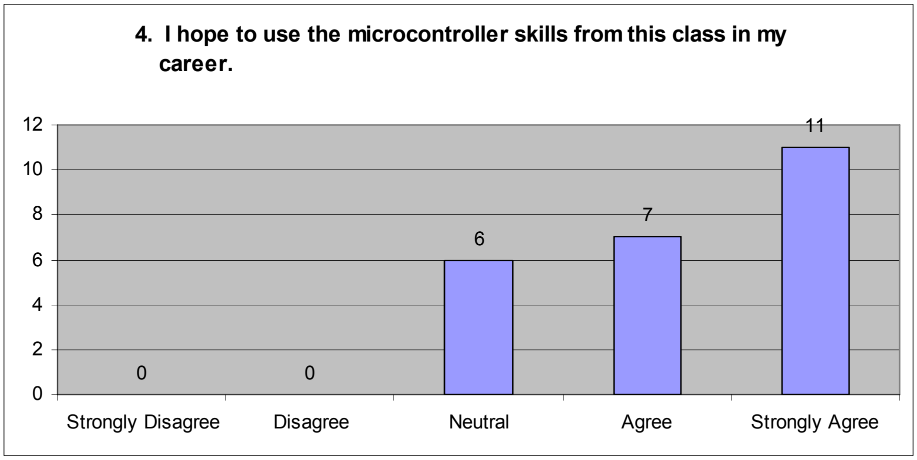

Figure 11: Assessment 4

The fifth statement, "I hope to use my labview skills in my career", had a less positive response; although the majority are in agreement, there are a few dissenters who disliked labview.

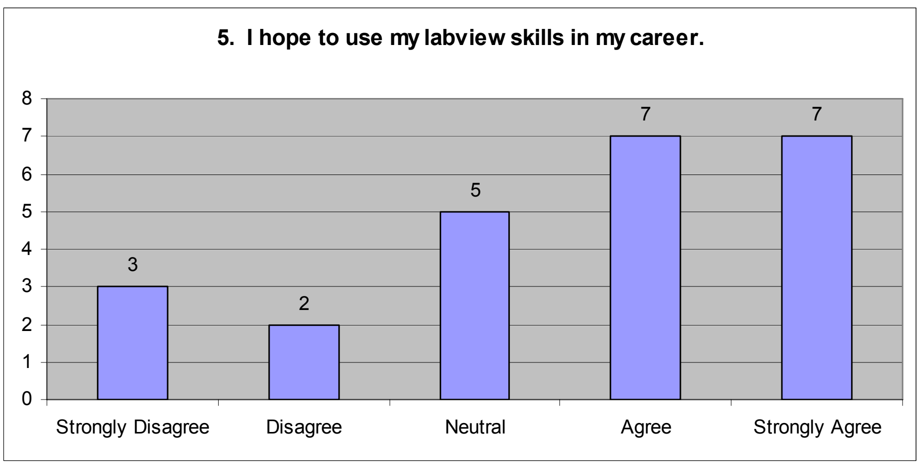

Figure 12: Assessment 5

<u>Conclusions</u>

Incorporating microcontrollers with a Measurement Systems class was successful. The students valued the material and it fit nicely within the traditional curriculum. The class did have to be streamlined somewhat to fit the microcontroller material, but overall I feel, as the instructor, that the added material is more worthwhile than the lost material, and the surveys show that the students agree.

The DAQ application is a natural one for teaching microcontrollers, as it combines a simple program with the peripherals on the chip to create a compelling and useful example. The students who completed the microcontroller portion of the class were in general excited about using them in other projects such as their senior capstone project.

The approach of adding short microcontroller modules to existing lower-division classes would seem to be a good one, as it allows the students to gain useful skills without taking extra units.

With the success of this class, plans are being made to incorporate microcontroller topics in the Computer Programming as well as the Circuit Analysis courses. The hardware will evolve during this work, ultimately to a custom circuit board, to be assembled and owned by the students themselves.

Future papers are planned to document the use of the system in these classes, and the hardware and software designs that support them.

References

1. www.parallax.com
2. http://www.atmel.com/dyn/products/tools_card.asp?tool_id=3146
3. www.gumstix.com
4. www.acroname.com
5. Barakat, Nael; Jack, Hugh; "A Student Owned Microcontroller Board" American Society of Engineering Education general conference, June 2006
6. Holden, M. "Low-Cost Autonomous Vehicles Using Just GPS", Computers in Education Journal, American Society of Engineering, Vol XV, No. 3, July-September 2005
7. www.hpinfotech.ro
8. www.ni.com