A New Course to Teach Microcontrollers and Embedded Networking to Biosystems and Agricultural Engineers

Matthew J. Darr, *Ohio State University*
Timothy S. Stombaugh, *University of Kentucky*
Scott A. Shearer, *University of Kentucky*
Richard S. Gates, *University of Kentucky*

Int. J. Engng Ed. Vol. 23, No. 4, pp. 716–722, 2007
Printed in Great Britain.
0949-149X/91 $3.00+0.00

A New Course to Teach Microcontrollers and Embedded Networking to Biosystems and Agricultural Engineers*

MATTHEW J. DARR
Research Associate, The Ohio State University Department of Food, Agricultural, and Biological Engineering, Columbus, OH 43210, USA. E-mail: darr.27@osu.edu

TIMOTHY S. STOMBAUGH
University of Kentucky Department of Biosystems and Agricultural Engineering, Lexington KY 40506, USA. E-mail: tss@uky.edu

SCOTT A. SHEARER
University of Kentucky Department of Biosystems and Agricultural Engineering. E-mail: scott.shearer@uky.edu

RICHARD S. GATES
University of Kentucky Department of Biosystems and Agricultural Engineering. E-mail: ates@bae.uky.edu

One of the fundamental skills required of biosystems and agricultural engineers is an ability to interact with systems that affect the production and processing of biological materials. This involves monitoring and controlling parameters within complex biological systems. Also, there is often a need to link multiple systems over a network to allow control and feedback data to be shared at several points. These monitoring systems and more sophisticated embedded networks are enhancing the ability of biosystems and agricultural engineers to solve problems by facilitating real-time data collection and enabling control actions. The use of microcontrollers in industry applications is growing steadily each year. Currently, worldwide microcontroller unit sales are increasing at a rate of 13% to 17% annually. This growth is reflected in biosystems and agricultural engineering by an increased use of microcontrollers in all aspects of the profession. Food and bioprocessing groups are utilizing microcontroller resources to improve the accuracy and efficiency of process control machines. Microcontrollers are also increasingly used for environmental control, where they have been implemented into distributed control systems. Applications are increasing in all areas of specialization where cost effective and precise control is required. In order to fulfill the departmental mission to provide students with the highest quality and most diverse learning experience possible, the University of Kentucky Department of Biosystems and Agricultural Engineering has developed a course to introduce the basic operation and industrial use of microcontrollers and embedded network systems. The fundamental goals of the course were to teach students the basic operating principles of microcontrollers and communication protocols, and to introduce practical microcontroller uses in industry. Laboratory assignments were tailored toward applications in biosystems and agricultural engineering including analog data acquisition, environmental monitoring, and process control. Principle concepts of motor and valve control were also discussed as they relate to real-world control applications.

Keywords: microcontroller; instrumentation; data acquisition; embedded systems; controller area networks

INTRODUCTION

The objective of this manuscript is to disseminate new teaching topics aimed at emerging technologies within Biosystems and Agricultural Engineering. Technology has drastically impacted control, data acquisition, and processing systems that are routinely used to solve problems within this field of engineering. This manuscript defines the blueprint which has been used to educate undergraduate and graduate students in advanced microcontroller technology, and can be used as a starting point for institutions of higher learning which choose to teach such technologies.

INTRODUCTION TO MICROCONTROLLER TECHNOLOGY

Advanced electronic and automated systems are no longer dedicated to specialized robotics or factory automation, as they have found their way

* Accepted 20 June 2007.

into multiple focus areas of Biosystems and Agricultural Engineering. Farm machinery automation has provided numerous advancements in data collection and control in production agriculture [1]. Many of the systems used to implement "precision agriculture" methodologies are based on inexpensive and dedicated microcontroller systems. Microcontroller based control and data acquisition systems are also growing rapidly in soil and water based applications such as soil moisture monitoring [2], plant environment control [3], and in animal environment monitoring [4].

Many reasons exist why microcontroller use is growing in Biosystems and Agricultural Engineering areas. The size and cost of microcontrollers provides simple, cost effective implementation into agricultural systems. An average sale price of an 8-bit microcontroller which will provide a basic level of instrumentation and control was $3.84 in 2002. The use of microcontrollers in other areas of engineering has also benefited agriculture by providing sophisticated software compilers and readily available sample applications. While the average household in the U.S. still does not own a computer, there are an average of 271 microcontrollers per household. These are found in cell phones, dishwashers, televisions, alarm clocks, and clothes irons among others [5].

The fundamental operation of microcontroller systems can be broken into three distinct levels.

1. Microcontroller architecture and instruction set
2. Internal peripherals and interrupt handling
3. Communication with external functionality and embedded networks.

The chip architecture defines the physical design of the microcontroller [6]. This will include the layout of memory and instruction execution paths inherent to a specific brand of microcontroller. The 8.2 billion microcontrollers shipped worldwide in 2002 were distributed mainly by a handful of companies [5]. Each company uses a slightly varied architecture which employs specialized functionality to improve performance and/or enhance marketability. Fortunately, once a single architecture is mastered it becomes much easier to move between competitive products.

Instruction sets also vary among competitive microcontroller developers [6]. The instruction set is a defined list of executable functions that the microcontroller can process. More advanced functions are developed by building structured routines from the basic instruction set. The most basic implementation of instruction set functions is through either machine code or assembly language programming. While these languages provide an efficient means of programming, they are better suited to a trained assembly language engineer than to an agricultural engineer simply trying to improve the functionality of a given system. Fortunately, software compilers have been developed that allow biosystems and agricultural engineers to create operating code in a more user friendly development environment such as BASIC or C++ programming [7].

Internal peripherals are used to provide expanded functionality to basic microcontrollers [6]. The most common internal peripheral used in biosystems and agricultural applications is the analog to digital converter which allows sampling of analog sensors within a system. Another commonly used peripheral is the pulse width modulation (PWM) engine which provides a dedicated pulse wave output with controllable frequency and duty cycle. PWM control signals are commonly used in variable speed motor control applications and can be used to create an analog output voltage when the output is properly filtered [7]. Timer peripherals are commonly used to create precise timing loops for data sampling intervals, control routine timing, or activity time stamping. The final commonly used peripheral is the counter which is mainly employed to determine the frequency of an input signal.

One major limitation of microcontrollers, as with most computer systems, is that the internal processor can only perform one function at a time. There is an inherent limitation to multi-tasking due to the limited processor capabilities. To overcome this problem, hardware interrupts and interrupt handling is utilized [6]. Once configured, hardware interrupts will execute specific tasks and only utilize processor time briefly after triggered events [8].

The final aspect of microcontroller operation relating to biosystems and agricultural engineering is an understanding of why and how multiple microcontrollers could be linked together to maximize system performance. As stated previously, microcontrollers are typically dedicated to single tasks and cannot handle substantial multitasking. Often specialize peripherals are required that can perform operations not inherent to the host microcontroller or can perform operations at a higher level of accuracy. This is typical of analog to digital conversions. Most 8-bit microcontrollers contain analog to digital converters with a resolution range of 8—12 bits. This may be unsatisfactory for certain systems. Low cost analog to digital converter peripherals are available to provide enhanced levels of resolution and accuracy [9]. An alternative to this master-slave communication is master-master communication where multiple microcontrollers communicate directly between one another. This is commonly seen in agricultural systems where distributed control and/or data collection are employed [10].

DESCRIPTION OF COURSE OBJECTIVES AND ORGANIZATION

The focus of BAE-599 was to develop the following list of competencies:

1. Understand the basic architecture of microcontrollers

2. Develop assembly language programs for microcontrollers
3. Use software compilers (high level languages) and understand their advantages and disadvantages
4. Understand the importance of interrupt handling in microcontrollers and be able to implement interrupts appropriately within programs
5. Configure and operate all microcontroller internal peripherals including analog to digital converters, pulse width modulators, counter/timers, frequency analyzers, and communication ports
6. Understand the mathematical computation limitations inherent to 8 bit, 16 bit and 32 bit microcontrollers
7. Communicate between multiple networked microcontrollers via RS232 and CAN 2.0 B communication protocols
8. Interface microcontrollers with external peripherals including level shifters, transceivers, analog to digital conversion chips, LCD screens, and off chip memory
9. Design and implement a process control system using microcontrollers
10. Understand the ISO 11783 (CAN 2.0B), DIN 9684 (LBS), and SAE J1939 (CAN 2.0B) communication protocols

The course objectives fall into three distinct education modules. Module I focused on the fundamental operation of microcontrollers. This module covered course objectives one, two, three, and four. Educational Module II focused on microcontroller functionality, specific internal peripherals, and mathematical computation capabilities and limitations. This module covered objectives five and six. Module III was perhaps the most useful education section to biosystems and agricultural engineers as it provided skills which enabled students to solve real engineering problems via microcontroller applications. It focused on the transfer of data between master microcontrollers, slave peripherals, and other master microcontrollers. This module covered course objectives seven, eight, nine, and ten.

The PIC18F458 from Microchip® (Chandler, AZ) was selected as the course microcontroller. This 8-bit device provided 32 Kbytes of program memory along with 1500 bytes of RAM. The PIC18F458 was driven by a 20 MHz oscillator and was able to execute instructions in a time of 0.2 μsec. It also contained many of the peripheral devices desired including:

- 8 channels of 10 bit analog to digital conversion
- USART driver for RS232 and SPI communication
- Internal CAN protocol engine
- Hardware pulse width modulation driver
- 4 hardware timers
- Multiple interrupt capabilities

LEARNING MODULE I MICROCONTROLLER ARCHITECTURE

Typical students in biosystems and agricultural engineering had little if any experience with microcontrollers or computer architecture during an undergraduate program. Students must quickly learn basic elements of computer processor architecture to effectively understand the operating characteristics of microcontrollers. The architecture section of the course was broken into five topic areas:

1. Microcontroller data bus architecture
2. Memory types and allocation
3. Instruction set and hardware stack management
4. Assembly language programming
5. Compiler programming using PIC Basic Pro®

The data bus architecture described the individual components which make up microcontrollers. The concept of a data bus system in which multiple peripherals interface to a central processing unit was discussed in detail. Considerable time was spent with each component including the microprocessor, data stack, memory address bus, memory data bus, instruction decoding block, program counter, accumulator, and the arithmetic logic unit.

Discussion of memory types and allocation begins the process of understanding how data was transferred through a microcontroller (Figure 1). FLASH program memory was explained as the destination for user code. Students responded positively to this discussion as they began to see the architecture as a physical reality rather than simply a schematic on the chalk board. RAM memory was discussed in detail as it is generally split into two separate sections, general purpose program memory and control registers. A common analogy was made between control registers and light switches, as they both act as logic devices to turn on or off circuits. Again, students responded well when concepts were related to real world objects or ideas. EEPROM memory was introduced in its role as a low volume, non-volatile memory bank. Traditional biosystems and agricultural engineering applications of EEPROM were mentioned, which generally revolved around utilization of calibration parameters that required sporadic adjustment.

The microcontroller instruction set and assembly language programming was taught at a simplistic level to gain a baseline understanding of the instruction execution process. A single assembly level lab was used to demonstrate the nature of simple operations such as a pin toggle and program pause. After completion, assembly language was abandoned and programming was done using a PIC BASIC PRO® compiler from Microengineering Labs, Inc. (Colorado Springs, CO). This compiler was preferred over comparable C++ compilers due to the simple nature of the

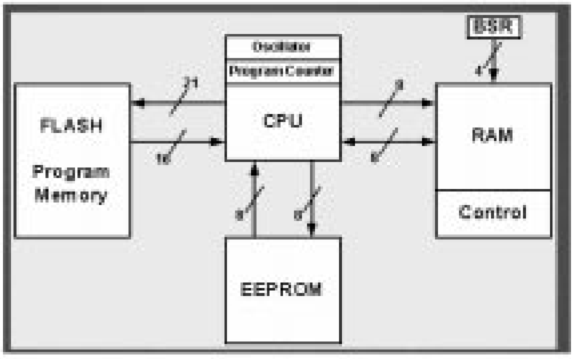

Fig. 1. PIC18F458 memory map

BASIC language which allowed students with no programming experience to quickly learn the syntax. In general, the predefined microcontroller commands inherent to the PIC BASIC PRO were not used. The compiler was simply used as a method to address registers and perform mathematical functions. This was beneficial to the students because they were not able to rely heavily on subroutines that they did not design themselves.

LEARNING MODULE II—PERIPHERAL INTERFACING AND EXPANDED FUNCTIONALITY

Module II focused on configuration and implementation of on-chip peripherals for added functionality. Five separate peripheral types were discussed:

1. Digital input and output capacities
2. Analog to digital conversion
3. Pulse width modulation
4. Multiple applications for timers
5. Hardware interrupts

Digital signal input and output (I/O) functionality was discussed as it related to control systems. Emphasis was placed on understanding the electrical limitations of each I/O pin such as max current sinking and sourcing capacity. Laboratory tests using load resistors were also completed to verify the current sourcing capacity of each I/O pin and of the entire chip. The use of pull-up and pull-down resistors was also discussed as they related to digital input signals.

The basic concept of analog to digital conversion (ADC) as well as implementation techniques in microcontrollers was discussed in detail during lecture. Although the process of signal digitalization was covered, more focus was placed on the resulting value which was returned in units of counts. The authors had noted in previous instrumentation classes that computer based data acquisition systems automated the conversion process from counts to voltage and left the student without a true understanding of the process. At the completion of the ADC introduction students were well versed in the terminology and methodology of analog conversions including, resolution, full scale range, and quantization error.

When configuring the ADC peripheral module, special attention was paid to each register associated with sampling time. The maximum sampling frequency was calculated directly from the timing associated with each step in the conversion process. Limitations of the ADC were discussed based on the calculated maximum sampling frequency. An in-class demonstration showed how Nyquist's sampling frequency was related to the maximum sampling frequency of the microcontroller.

The PIC 18F458 contained four timer modules that could be configured for multiple applications. The most widely used application for timers was to create specialized timing loops for pulse counting, designating sampling intervals, or determine the frequency of external signals. Frequency analyzing is a commonly used practice in biosystems and agricultural engineering to determine the speed of a rotating shaft instrumented with a sensor designed to output a given number of pulses per rotation. It is also commonly used in stream flow monitoring where a turbine monitoring device provides a pulse output corresponding to the flow velocity of the stream. Class examples showed how a single timer module could be setup as the acquisition timing element and a second timer module could be configured as an external pulse counter. By counting pulses for a given time period, the frequency (pulses per second) was automatically known. Students experimented with this type of signal processing by utilizing function generators as the sensor input in a laboratory setting. They experimented with the functionality of the acquisition unit and found that the limiting factors were update rate and timer register size depending on the magnitude of the input frequency.

For high frequency signals, the sampling duration must be minimized to ensure that the 16-bit input counter did not overflow. The opposite was true for extremely slow signals. For slow signals an external trigger was used to interrupt the processor when a rising or falling edge signal was found. Immediately on the rising edge, the timer was initiated. Once the falling edge was reached, the timer was stopped and the appropriate on time was directly calculated from the resultant value in the timer register. This provided an accurate measurement of frequency for square wave signals at a relatively low frequency. Timer modules are also used to drive other functionality within the microcontroller.

Pulse width modulation (PWM) is a common application used in multiple areas of biosystems and agricultural engineering. In hydraulic control systems, PWM control is used to rapidly cycle high frequency valves and thus control the flow rate through the system. PWM is also used in servo valve applications as a means to supply adjustable analog current signal. During configuration, timer registers are utilized to define the duty cycle and frequency of the PWM signal.

Hardware interrupts have been discussed briefly and their inclusion in the course provided students with the experience of developing embedded systems similar to those used in industry. Interrupts allow for event-driven programming, which means that portions of logic execute only when initiated by an external signal or internal background event. External interrupts were discussed at length during lecture to convey the benefits of event-driven programming. Take for example a system where a single push button switch event from the user will instigate a series of controlled events. If no interrupts were employed, then the processor must execute a tight program loop to watch for a change in the external signal from the push button switch. This is inefficient, because nothing is being accomplished while waiting for a change in signal. If interrupts were utilized, the program can continue to execute other objectives such as digital signal processing, feedback communication, or simply go into sleep mode to reduce power consumption. When a change in the external signal occurs, the processor will jump to the appropriate location and execute the desired code. Several peripheral operations can be configured as interrupt driven including:

- Timers overflow
- Timers capture/compared occurred
- External signal edges
- External state changes
- Parallel slave port read/write
- Analog to digital conversion complete
- USART transmit/receive complete
- Comparator input change
- EEPROM write operation change
- System low voltage detection
- CAN bus error
- CAN bus activity wake-up
- CAN bus transmit/receive complete

LEARNING MODULE III— COMMUNICATION PROTOCOLS AND APPLICATIONS

The final learning module covered an area with great application to biosystems and agricultural engineering. Phase one of this module explored interfaces to other microelectronics that could expand system functionality above the original chip capabilities. This dealt with peripheral functions, but now the peripheral engine was located outside of the main microcontroller and electronic signals were communicated between the dual chips. The second phase of the module dealt with linking multiple microcontrollers together over a distributed network system.

Several proprietary communication protocols have been developed to link microcontrollers to external peripherals. Serial Peripheral Interface (SPI), initially developed by Motorola, is widely used for this for this purpose and has become a standard language between multiple manufacturers. Inter-Intercomputer Communications (I2C) is also widely used for this type of communication, but was not discussed in class due to time restraints. Examples of peripherals that communicate via SPI include high resolution analog to digital converters, input channel multi-plexors, temperature sensors, pressure sensors, and external nonvolatile memory. Two examples were used in lecture and in the laboratory to demonstrate the functionality of SPI peripherals.

A simplified approach to SPI communication was granted by allowing students to utilize compiler commands for SPI. This simplified the code requirements of the protocol and instead focused the emphasis on how SPI might be incorporated to improve an embedded design. The first SPI example was an EEPROM memory chip 25AA320 from Microchip®. This chip provided 4000 bytes of external memory which was nonvolatile in nature and could be used for data logging or calibration information. The laboratory task for each student was to sample an analog voltage at a rate of one hertz and store the result into EEPROM for future recovery. An interrupt driven external trigger routine was used to designate when the data should be recovered from the EEPROM device. Upon edge trigger change, all data that had previously been previously written was downloaded to a PC via an RS232 connection.

The second SPI example demonstrated the use of a simple electronic sensor that could greatly reduce instrumentation headaches. A MAX6675 cold junction compensated K-type thermocouple and digital converter chip, produced by Maxim® (Sunnyvale, CA), was used to sample high temperature air. Two major problems exist when trying to incorporate thermocouples into an embedded design. First, the output signal is of very small magnitude and requires amplification circuitry. Secondly, a cold junction compensation circuit must be incorporated to determine the actual temperature. Both limitations were overcome with the MAX6675, which completed cold junction compensation and digitization all within its 8 pin DIP footprint. The students were able to "clock out" the current temperature from the MAX6675 via an SPI connection with a resolution of 0.25°C and a maximum temperature of 1025°C.

Success concerning SPI communication was slightly less than previous sections as students had difficulty adjusting their mindsets from internal hardware peripherals to external peripherals. The MAX6675 was an excellent peripheral selection to cover in lab as it demonstrated the improved functionality that can be gained with SPI peripherals.

RS232 communication was discussed briefly in an effort to conserve lecture time and because each student in the course had previous experience with RS232. Again, compiler software implementation was used rather than programming each hardware register individually. RS232 was used in two fashions during the course. The first was interfacing with a serial LCD screen. The LCD screen, BPI-216 from Scott Edwards Electronics Inc. (Sierra Vista, AZ), received simple commands from the microcontroller via an RS232 connection. The LCD screen was used during each laboratory as a means to display register and variable values. This provided a simple avenue for troubleshooting project code and for providing feedback to the student.

A second implementation of RS232 was for reading position data from a Global Positioning Satellites (GPS) receiver. Students used a Garmin (Kansas City, MO) GPS76 receivers to simulate a GPS positions within the laboratory. They then programmed their microcontroller to read the incoming GPS string and displayed pertinent data back to the LCD screen.

The final aspect of communication was master-master communication between several microcontrollers. This section of the course was particularly focused on recent advancements in networking of agricultural vehicles and implements. The CAN Bus, which is becoming standard equipment on much of today's agricultural equipment, is a computer network designed specifically for mobile equipment. Through the use of a two wire resistive network, the CAN Bus enables serial data to be communicated between multiple points on a vehicle. Every aspect of CAN communication was covered in lecture including message prioritization, arbitration, bus timing, time quanta, identifiers, transmit/receive buffers, bus error analysis, and CAN interrupt handling. No compiler software was available for CAN implementation, so students set hardware registers individually to implement the protocol.

An end of the semester project was assigned to test CAN Bus networks and to evaluate the students ability to solve a problem while working in separate groups. This was modeled after industry where several working groups will collaborate on a final system. The project topic was an HVAC system for a new age office complex. The system was to be fully automated and communicate over a CAN Bus. The follow list describes the required working groups:

1. Main power node to turn system on or off
2. Data acquisition node within the office area to report current temperature and set-point temperature
3. Data acquisition node to report external air temperature
4. Air conditioning unit node to heat/cool external fresh air and recirculate inside air
5. Fan controller node to implement variable speed fan drive based on current and set-point temperature data
6. Feedback display node to show current fan speed as well as all temperature readings

The students successfully implemented the CAN Bus network and control system within a single three hour lab period. The design worked very well and demonstrated the expandable nature of CAN communication as well as the ability to efficiently share information among multiple end points.

COURSE WEBSITE

A website for the course was developed and posted at http://www.bae.uky.edu/Instruction/CourseDescriptions.htm. The website contains links to each lecture used in the course in a PowerPoint® format. Course handouts are also available that correspond to each of the lecture presentations. Links are available to data sheets and manuals used as references throughout the course. The website was designed with an open architecture so that other educational institutions may freely borrow material for advancing the educational experience within their own programs.

COURSE EVALUATION DATA

The following is a summary of student feedback after completing the course:

Sixty percent of the students agreed and 40% strongly agreed that this course enabled them to analyze problems from a different viewpoint. Similar results were obtained when considering the students' response to their ability to analyze and evaluate problems in a better manner, as 42.9% agreed and 57.1% strongly agreed. Twenty-nine percent agreed and 71.4% strongly agreed that this course increased there ability to solve problems and has provided them with an application-based tool that will increase their future engineering abilities. Fifty-seven percent agreed and 28.6% strongly agreed that this course stimulated them to read further into these developing subject matters and maintain connected with emerging technologies.

SUMMARY

Electronic systems have continued to grow in popularity as a means for engineers to quickly and accurately monitor and control systems. Specifically, embedded microcontrollers have the ability to provide extremely simple and cost effective solutions to many areas of Biosystems and Agricultural Engineering. Microcontrollers and Controller Area Network Applications for Biosystems Engineering, BAE 599, at the University of Kentucky Department of Biosystems and Agricultural Engineering was developed to provide a background in microcontroller architecture and introduce various applications for embedded microcontroller systems. The basic instruction set for microcontroller operation was discussed to demonstrate the functionality and operational characteristics of an 8-bit Microchip® PIC18F458. Individual peripherals such as digital I/O, analog to digital converters, pulse width modulators, counters, and timers were discussed in detail. Laboratory exercises emphasized peripheral configurations discussed in lecture and provided real world application examples. Several communication protocols were discussed including Serial Peripheral Interface (SPI), RS232, and Controller Area Network (CAN). Students responded very well to the hands on teaching approach and communicated positive feedback towards the topics and organization of laboratory assignments. As a group, the students felt they were better prepared to solve industry problems and were pleased with the opportunity to learn about embedded microcontroller systems.

REFERENCES

1. Reid, J. F., Q. Zhang, N. Noguchi, and M. Dickson. 2000. Agricultural automatic guidance research in North America. *Computers and Electronics in Agriculture* **25**, 155–167.
2. Valente, A., J. B. Cunha, J. H. Correia, and C. Couto. 2002. A Silicon Probe with Integrated Microelectronics for Soil Moisture Measurements. In: *Proc. of the World Congress of Computer in Agriculture and Natural Resources*, 440–446. Iguacu Falls, Brazil, 13–15 March.
3. Brown, D., and R. E. Lacey. 2002. A distributed control system for low pressure plant growth chambers. ASAE Paper No. 023078. St. Joseph, MI.: ASAE.
4. Agouridis, C. T., T. S. Stombaugh, S. R. Workman, B. K. Koostra, D. R. Edwards. 2003. Examination of GPS Collar Capabilities and Limitations for Tracking Animal Movement in Grazed Watershed Studies. ASAE Paper No. 032001. St. Joseph, MI.: ASAE.
5. Micrologic Research. 2002. Overview of the Microcontroller Market. Phoenix, AZ.
6. Peakman, J. B. 1998. *Design with PIC Microcontrollers.* Upper Saddle River, NJ: Prentice-Hall, Inc.
7. Predko, M. 2001. *Programming and Customizing PICmicro Microcontrollers..* New York, NY: McGraw-Hill.
8. Microchip. 2003. PIC18FXX8 Datasheet. Chandler, AZ.
9. Dabney, R. 2002. 15-Kilogram Scale Using the TC520. Microchip Application Note #AN780. Chandler, AZ.
10. Darr, M. J. 2004. Development and Evaluation of a Controller Area Network Based Autonomous Vehicle. MS Thesis. Lexington, KY: University of Kentucky, Department of Biosystems and Agricultural Engineering.

Mr. Matthew Darr received a MS degree from the University of Kentucky in Biosystems and Agricultural Engineering. He is currently a Research Associate at The Ohio State University and works in the Precision Agricultural and Air Quality fields.

Dr. Timothy Stombaugh received a PhD from the University of Illinois in Agricultural Engineering. His is currently an Assistant Professor at the University of Kentucky and works in the Precision Agricultural field.

Dr. Scott Shearer received a PhD from The Ohio State University in Agricultural Engineering. He is currently a Professor at the University of Kentucky and works in the Precision Agriculture field.

Dr. Richard Gates received a PhD from Cornell University in Biological Engineering. He is currently Professor and Head at the University of Kentucky Department of Biosystems and Agricultural Engineering. He works in the Air Quality and Animal Environment fields.