

## **Construction and Application of a Computer Based Interface Card**

Michael Combs  
Telescope Operations Engineer  
m.combs@morehead-st.edu  
Morehead State University  
Morehead, Kentucky

Ahmad Zargari, Ph.D., CSIT  
Associate Professor  
a.zargar@morehead-st.edu  
Morehead State University  
Morehead, Kentucky

### **Abstract**

Automated control of manufacturing systems and research tasks are becoming more demanding in today's competitive market. Computer interface and control applications are common practice not exceptions in the workplace. The knowledge of how to manage processes with computer control can be developed by working with basic tools of microprocessor interfacing. An interface card placed in a microcomputer can provide practitioners with the skills they need to be competitive. The Intel 82C55A programmable peripheral interface IC can be used to interface external hardware with a microcomputer.

The interface card being built and used at Morehead State University is based on the Intel 82C55A programmable peripheral I/O device. With minimal support circuitry the 82C55A can be interfaced with a microcomputer. The 82C55A contains three 8-bit ports providing 24 lines of digital I/O. Several circuits can be connected to the 82C55A making it more versatile. Counter/timers, analog to digital converters, digital to analog converters, sensors, and several other circuits can be connected to the interface card.

The card is used to control and monitor different processes through hardware and software interfaces. Software used to program the interface card includes BASIC, Visual Basic, Visual C++, and Labview. The software interfaces include simple to complex routines depending on the application. Construction and application of the Intel 82C55A interface card help provide an individual with the knowledge and skills of computer based control applications.

### **Introduction**

Microcomputers have developed into a powerful and versatile tool in today's industry and research labs. Microcomputers are being used to control complex tasks and processes. It has become necessary to have a basic understanding of how these control processes work in order to be competitive.

In order to control a process by a microcomputer an interface must exist from the microcomputer to the hardware being controlled. An interface can span from a simple to complex design.

Usually the more complex interface designs are more versatile and difficult to troubleshoot. In order to acquire an understanding of the theory and application of interfacing to a microcomputer a more basic design may be used. A popular and low cost interfacing component is the Intel 82C55A programmable peripheral interface. Any compatible TTL I/O device may be interfaced to the microprocessor using the 82C55A (Brey, 1997).

## Intel 82C55A

The Intel 82C55A programmable peripheral interface component is used to interface hardware to the microprocessor of a microcomputer. The 82C55A contains three 8-bit parallel ports for a total of 24 lines of digital I/O. This allows for one to several devices to be connected to the 82C55A. The 8-bit ports are configured by sending a control word to the control register (Rafiquzzaman, 1995). Figure 1 is a diagram describing the individual bits in the control register.

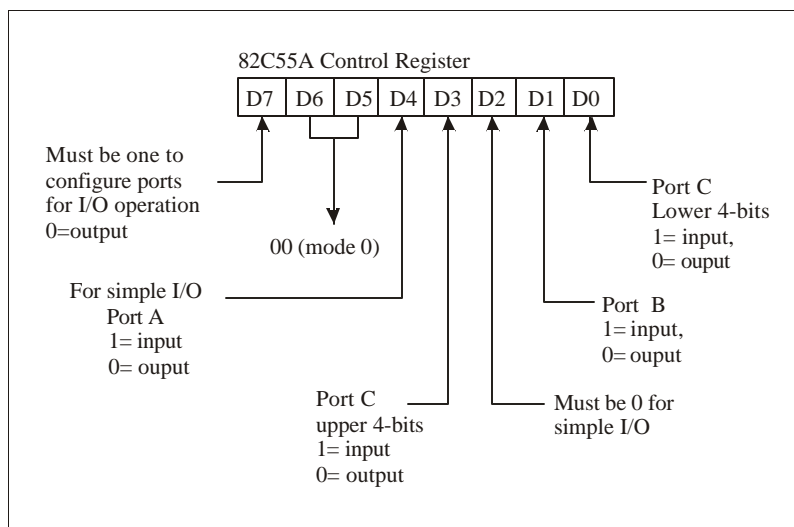


Figure 1. Diagram of the 82C55A control register bits (Rafiquzzaman, 1995).

The 82C55A can be configured to operate in three different modes. In mode 0 the three ports are configured for basic I/O operation, two handshaking I/O ports for mode 1, and a bidirectional I/O port with five handshaking signals for mode 2 (Uffenbeck, 1998). Throughout the paper the 82C55A will be configured for basic I/O operations in mode 0.

## Interface Card

The interface card is based on Intel's 8255A programmable peripheral interface (PPI) IC. The 8255A contains three 8-bit ports providing 24 lines of digital I/O. Support circuitry for the 8255A consists of a buffer and address decoder circuit. The buffer circuit buffers the signals between the 8255A and the motherboard, and the address decoder circuit assigns the 8255A an address on the address bus. The interface card is inserted into an open ISA slot in the personal computer. Through the interface card external circuitry may be connected to a personal computer. Figure 2 is a diagram of the schematic for the interface card.

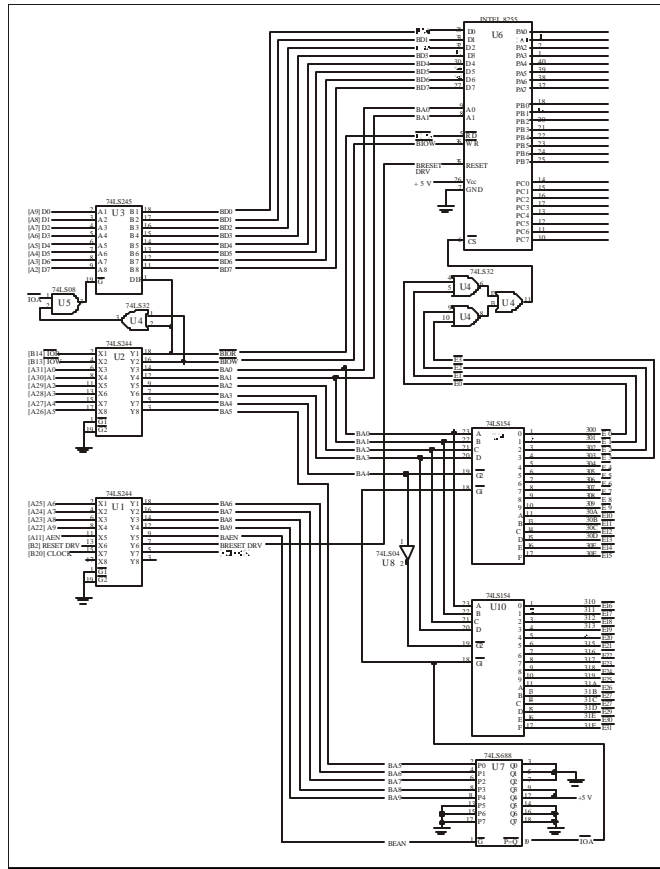


Figure 2. Schematic of the 8255A interface card.

The interface card is constructed using the wire wrap technique. This allows for easy troubleshooting if so needed. Components are placed on the ISA prototype board in a logical order to allow for neat wiring and ease of troubleshooting. A 25 pin connector is used to make external connections to the interface card. Figures 3 and 4 show the top and bottom views of the finished 82C55A interface card.

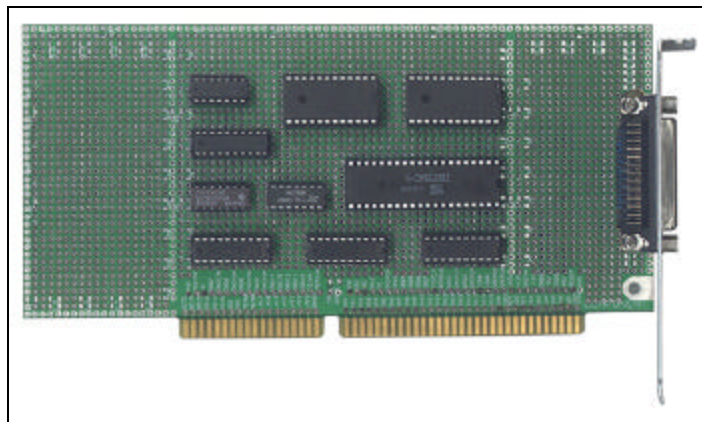


Figure 3. Top View of the finished 8255A Interface Card.

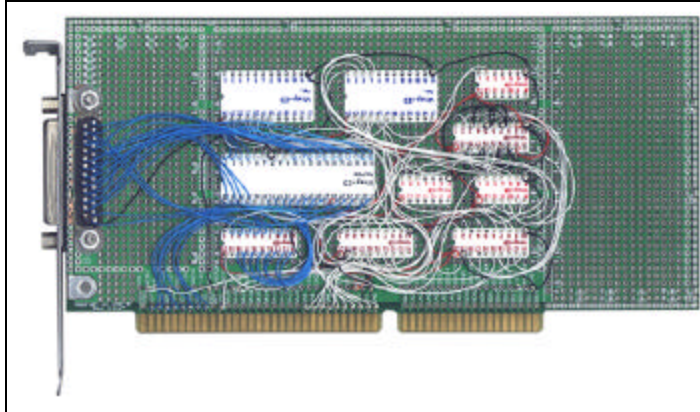


Figure 4. Bottom view of the finished 8255A Interface Card.

### Testing the Interface Card

Software routines are programmed to control the functionality of the interface card. BASIC, Visual Basic, and Visual C++ have been used to program the interface card. Programs in the National Instruments programming language Labview are also being developed. The following is a sample program to test the interface card once construction is completed.

```

100 REM 8255 PPI Set Ports A,B,C to Output
101 REM USES PORT B TO DISPLAY COUNT FROM 0 TO 255
110 BASEADDR = 768: REM 300H
120 PORTB = BASEADDR + 1
130 CNTRL = BASEADDR + 3
140 OUT CNTRL, 128
150 FOR SIGNAL = 0 TO 255
160 OUT PORTB, SIGNAL: PRINT "DECIMAL = "; SIGNAL
170 FOR DELAY = 1 TO 2000: NEXT DELAY
180 NEXT SIGNAL
999 END
  
```

The program sets all three ports of the 8255A as output. A simple circuit of eight LEDs are constructed to test the circuit. The program uses Port B to count from 0 to 255. If the card is functioning properly the LEDs will light sequentially in binary until 255 is reached and all LEDs are on. The 82C55A requires a control word be sent first to configure the ports for proper operation. Once this is done the proper commands or data can be sent to the individual ports of the 82C55A.

### Applications

Once the interface card has been tested and proven to work properly other circuitry may be added to the card. Depending on the application, simple to complex circuitry may be required. A complex example may be the control of an assembly line process. A simple example may be the individual circuit to control a motor in the assembly line process. Circuits may be connected in

two different ways. Circuits may be added directly to the 82C55A or due to the design of the address decoder circuit they may be added directly to the interface card. Figure 5 is the address decoder portion of the interface card.

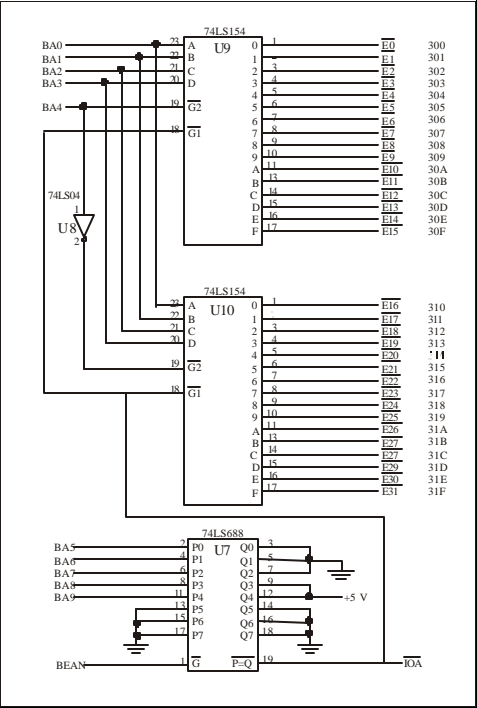


Figure 5. Address decoder portion of the interface card.

The address decoder allows addresses 300 to 31F to be used. The 82C55A uses address 300 to 303. Address 300 is assigned to port A, 301 to port B, 302 to port C, and 303 to the control register. Addresses 304 to 31F are open for future components to be connected.

Several different devices may be added to the interface card, examples are analog to digital converters, digital to analog converters, timers/counters, relays, various sensors, keyboards, displays, DC motors, and stepper motors. Depending on which device is to be added would determine how it would be connected to the card. For instance a keyboard or control of a motor would be added to the 82C55A portion of the interface card. An 82C54 programmable interval timer would be added to the interface card address decoder circuitry and function separately from the 82C55A. These are examples of the versatility and power of the interface card design.

**Stepper Motor Control**

A basic interface example is the control of a stepper motor system. A stepper motor is essentially a digital motor. It is controlled to move in discrete steps to rotate 360° (Bateson, 1993; Brey, 1997). The stepper motor may be controlled by using a motor driver also called a translator. The translator uses a digital TTL signal input to control the stepper motor.

The translator being used in this application is the SLO-SYN Micro Series packaged 430-PT translator manufactured by Superior Electric. The 430-PT is based on solid state electronics. The translator allows for 3.5A max current per phase. The translator has three connector sections, power input, motor connection, and signal I/O connector. Power input is used to supply 110V source for the translator. The motor connection interfaces to the stepper motor. The signal I/O connector is where the control lines from the 82C55A interface card will be connected (Oh, 1997).

The motor being used with the translator is the model number M093-FD07 SLO-SYN stepper motor manufactured by Superior Electric. The stepper motor has a current rating of 3.5 amps, and a step resolution of 200 steps for a complete rotation. Figure 6 is a picture of the SLO-SYN 430-PT translator. Figure 7 is a picture of the SLO-SYN stepper motor.



Figure 6. SLO-SYN 430-PT translator manufactured by Superior Electric.

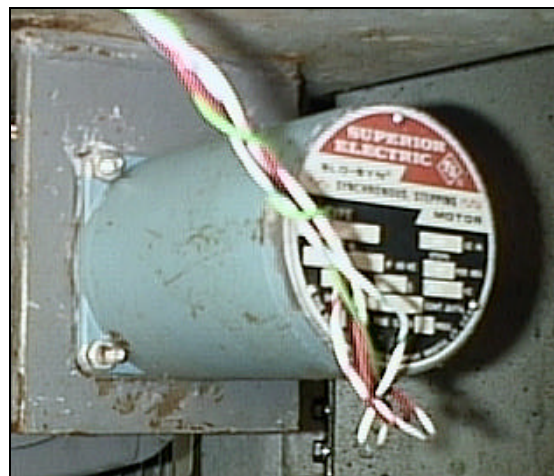


Figure7. Model M093-FD07 SLO-SYN

motor manufactured by Superior Electric.

### Interface Card and Translator Connections

Connections for the stepper motor control circuit are simple. The 82C55A interface card and stepper motor connect to the translator. Figure 8 is a block diagram of the stepper motor control circuit.

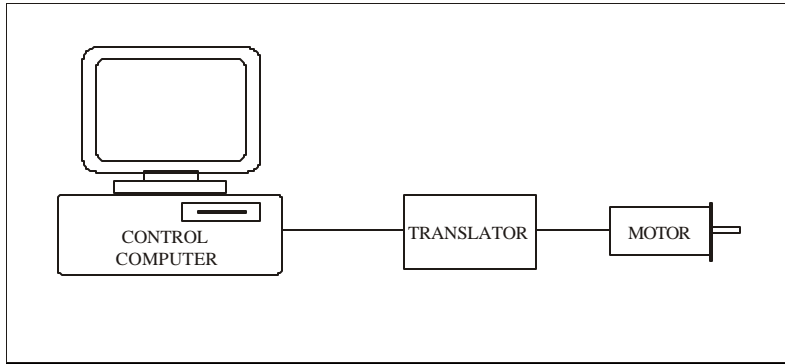


Figure 8. Block diagram for stepper motor control circuit.

Two control lines plus ground are needed for connection from the 82C55A interface card to the translator motor driver. Lines PA0 and PA1 are used from port A of the interface card. PA0 is connected to the direction control input of the translator. The PA1 line is connected to the pulse-in line to control the speed of the stepper motor by generating a variable pulse train. Table 1 lists the connector pins for the signal I/O connector on the translator.

Pin	Assignment
1	All Windings Off
2	CW/CCW
3	Pulse In
4	Not Used
5	Not Used
6	Not Used
7	Vo
8	Not Used
9	Half/Full
10	Opto Supply Out
11	Opto Supply In
12	Vo
13	Not Used
14	Not Used
15	Not Used

Table 1. Pin assignments for the signal I/O connector of the SLO-SYN 430-PT translator.

The translator has a few options for control, the translator may be configured for half or full step resolution. It may also be wired for turning the power off to the windings to allow for the motor to move freely. For this application the stepper motor will move in full steps and the all windings off is not wired. Pins 10 and 11 are wired together on the translator for it to use its internal opto isolation. This helps to protect the control lines of the translator. Figure 9 is a connection diagram from the 82C55A to the SLO-SYN translator.

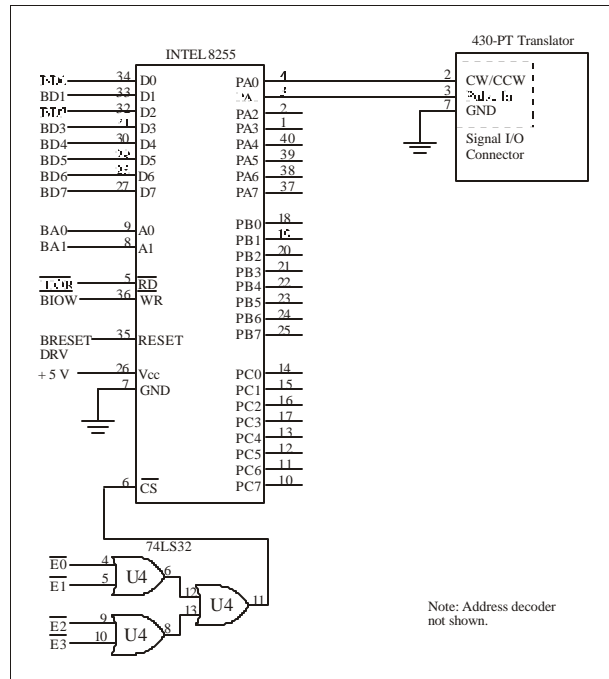


Figure 9. Connection diagram of the interface card to the SLO-SYN 430-PT Translator.

### Translator and Motor Connections

The translator contains a twist-lock circular female AMP connector for motor connections. Table 2 lists the connections pins and their assignments.

PIN	Assignment
1	M4
2	M1
3	No Connection
4	Ground
5	No Connection
6	M5
7	No Connection
8	M3

Table 2. Pin assignments for the motor connector of the SLO-SYN 430-PT translator.



Using the proper cable for the translator and motor connections is essential. Superior Electric suggest a shielded, twisted pair cable (Superior Electric, 1986). Figure 10 is a connection diagram from the translator to the stepper motor.

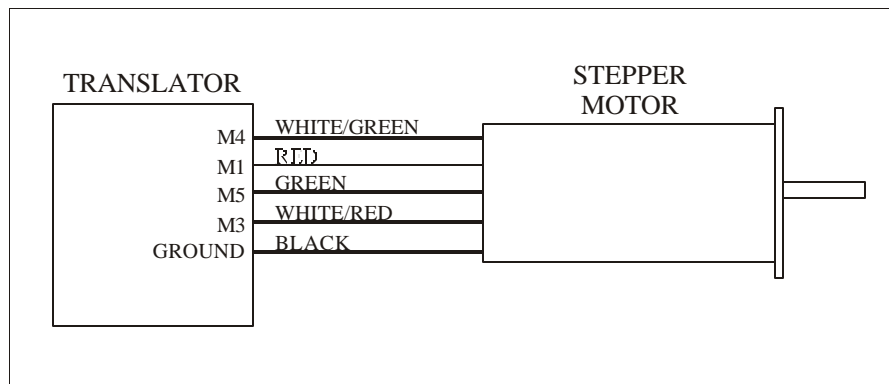


Figure 10. Connection diagram for the 430-PT translator and M093-FD07 SLO-SYN stepper motor.

Once all hardware connections are complete the software programming phase of the project may begin. Simple software routines may be programmed quickly for testing.

### Programming

The 82C55A is simple to program due to it containing only two internal command registers (Brey, 1997). These registers are used to configure the 82C55A for operation by the user. The following three programs are used to test the stepper motor interface application to insure proper wiring and configuration. All of the following programs all ports are set as output and port A is used to generate a pulse train to drive the stepper motor. The first program generates a pulse train set by the user by entering a delay value. The duty cycle is set at 50%.

```

100 REM 8255 PPI Set Ports A,B,C to Output
101 REM USES PORT A AS A PULSE OUT
110 BASEADDR = 768: REM 300H
120 PORTB = BASEADDR + 1
130 CNTRL = BASEADDR + 3
140 OUT CNTRL, 128
145 PRINT "TYPE IN A DELAY LOOP AMOUNT"
146 INPUT D
150 SIGNAL1 = 0
155 SIGNAL2 = 1
160 OUT BASEADDR, SIGNAL1
170 FOR DELAY = 1 TO D: NEXT DELAY
180 OUT BASEADDR, SIGNAL2
185 FOR DELAY = 1 TO D: NEXT DELAY
190 GOTO 160

```

The second program generates a pulse train and user defines the duty cycle by entering two different delay values for the high and low states.

```
100 REM 8255 PPI Set Ports A,B,C to Output
101 REM USES PORT A AS A PULSE OUT
110 BASEADDR = 768: REM 300H
120 PORTB = BASEADDR + 1
130 CNTRL = BASEADDR + 3
140 OUT CNTRL, 128
145 PRINT "TYPE IN AMOUNT FOR LOW DELAY"
146 INPUT L
147 PRINT "TYPE IN AMOUNT FOR HIGH DELAY"
148 INPUT H
150 SGNAL1 = 0
155 SGNAL2 = 1
160 OUT BASEADDR, SGNAL1
170 FOR DELAY = 1 TO L: NEXT DELAY
180 OUT BASEADDR, SGNAL2
185 FOR DELAY = 1 TO H: NEXT DELAY
190 GOTO 160
```

The third program generates a ramping pulse train The duty cycle is set by the user by entering a delay value for the high and low states as was in the second program .

```
100 REM 8255 PPI Set Ports A,B,C to Output
101 REM USES PORT A AS A PULSE OUT
110 BASEADDR = 768: REM 300H
120 PORTB = BASEADDR + 1
130 CNTRL = BASEADDR + 3
131 SGNAL1 = 0
132 SGNAL2 = 1
140 OUT CNTRL, 128
146 D = 5 + D
160 OUT BASEADDR, SGNAL1
170 FOR DELAY = 1 TO D: NEXT DELAY
180 OUT BASEADDR, SGNAL2
185 FOR DELAY = 1 TO D: NEXT DELAY
190 IF D < 700 THEN GOTO 146
195 IF D = 700 THEN GOTO 160
```

These three testing programs will help insure proper operation of the stepper motor control circuit. Once testing is complete more elaborate programs may be written depending on the application. The programs may be written in different languages also. Factors of the application will determine what is needed.

## Future Applications

The interface card allows for many different applications, and has the potential to control most anything with the proper support circuitry. Possible future interface projects could include interfacing a keyboard to the 82C55A. Problems interfacing a keyboard include detecting a single key stroke from a switch matrix, debouncing to keep from detecting several key strokes from a single key stroke, and assigning values to the individual keys (Uffenbeck, 1998). An 8259A programmable interrupt controller could be added to the interface card. The 8259A could be used to monitor input switches to control corresponding events (Goody, 1993). The 82C54 programmable interval timer can be used to generate pulse trains, time events or used as a counter (Triebel, 1998).

## References

- Bateson, R. N. (1993). Introduction to control system technology (4<sup>th</sup> Ed). New York, NY: Macmillan.
- Brey, B. B. (1997). The Intel microprocessors 8086/8088, 80186, 80486, Pentium, and Pentium Pro processor: Architecture, programming, and interfacing (4<sup>th</sup> Ed). Upper Saddle River, NJ: Prentice Hall.
- Goody, R. W. (1993). Intel microprocessors: Hardware, software and applications. New York, NY: Glencoe.
- Oh, P. (1997). 8255 interface card applications manual. Philadelphia, PA: Boondog Automation.
- Rafiquzzaman, M. (1995). Microprocessors and microcomputer-based system design (2<sup>nd</sup> Ed). New York, NY: CRC Press.
- Superior Electric (1986). Instructions for SLO-SYN Micro Series packaged translators types 230-PT and 430-PT. Bristol, CT: Superior Electric.
- Triebel, W. A. (1998). The 80386, 80486, and Pentium processor: Hardware, software, and interfacing. Upper Saddle River, NJ: Prentice Hall.
- Uffenbeck, J. (1998). The 80X86 family: Design, programming, and interfacing (2<sup>nd</sup> Ed). Upper Saddle River, NJ: Prentice Hall.

## Biography

Michael Combs is the Telescope Operations Engineer of the Astrophysics Laboratory at Morehead State University. He has been with the Astrophysics Laboratory since 1996, starting as an undergraduate and is now employed full time. He develops and maintains the systems of the Morehead Radio Telescope, which is a 13 meter Nike Hercules antenna converted for radio astronomy work. Michael Combs also teaches electronics courses half time in the IET Department.

Ahmad Zargari, Ph.D., CSIT. Research interests in Technology Graduate Programs, Job Market for IT, Alumni and Employers Feedback, Microprocessors, Total Quality Management, Statistical Process Control (SPC) and Design of Experiments (DOE). A senior member of Society of Manufacturing Engineers, member of Epsilon Pi Tau, National Association of Industrial and Technical Teacher Educators, International Technology Education Association (ITEA) and member of the American Society for Engineering Education (ASEE).