INTERFACING AN 8085-BASED MICROCONTROLLER: A PRACTICAL APPROACH TO DEVELOPING COMPUTER APPLICATION SKILLS

Jian-ren Li, Ph.D.

WESTERN KENTUCKY UNIVERSITY

Ahmad Zagari, Ph.D., CSIT

MOREHEAD STATE UNIVERSITY

ABSTRACT

In the Electronics Technology course – EE 445, students were directed to construct, troubleshoot, and interface an 8085-based microcontroller in order to control electromechanical circuits. This microprocessor, using Intel 8085A processor, was assembled from components and was interfaced to take full control of a stepper motor's motion. In addition, the microprocessor's application to generate and measure waveforms was examined. This manuscript attempts to describe the basic architecture of a microcontroller system, and examine its interfacing techniques as well as its applications in providing communication to the outside world.

I. INTRODUCTION

The knowledge of computers and their applications has become a fundamental technical skills required of electronics engineering technology students. In order to provide students with an opportunity to develop microprocessor application skills, this course is focused on teaching programming and application of Motorola 68HC11-A8 microprocessor. Although students who have learned one type of microcontrollers in class can adapt without too much difficulty to other types, this usually takes a considerable time and effort and they may not afford the practice and learning of the function of other types of processors during their academic years of university study. On the other hand, the competitive labor market may require them to demonstrate skills on major types of microcontrollers such as Intel-based processors immediately after their graduation. In order to help students to develop a crosslinking skills so that they are able to work on at least two types of microcontrollers, i.e., Motorola and Intel, a senior project was integrated in the Computer Electronics course – EET 445. In this course, while a student is provided with the knowledge of Motorola microcontroller instruction set and interfacing in classroom and laboratory setting, he or she builds an 8085A-based microcomputer system as a major project from components.

II. THE 8085A-BASED COMPUTER SYSTEM

The 8085A microprocessor is a 40-pin semiconductor integrated circuit (IC) device. In the 8085A microprocessor, in addition to execution of the instruction set, the functions of clock generation, system bus control and interrupt priority selection are contained. The 8085A transfers data on an 8-bit bi-directional 3-state bus (AD0-AD7) which is time multiplexed so as to also transmit the eight low-order address bits. The 8085A CPU generates control signals that can be used to select appropriate external devices and functions to perform READ and WRITE operations and also to select memory or I/O ports. The 8085A can address up to 256 different I/O locations. These addresses have the same numerical value (00-FF) as the first 256 memory addresses. They are distinguished by means of the IO/M output from the CPU.

The 8085A is provided with internal 8-bit and 16-bit registers. General-purpose registers B, C, D, E, H, L may be used as six 8-bit registers or as three 16-bit registers, interchangeably, depending on the instruction being performed. H-L pair functions as a data pointer to reference memory addresses that are either the sources or the destinations in a number of instructions. A smaller number of instructions can use B-C or D-E for indirect addressing. The flag register contains five one-bit flags. These are carry, auxiliary carry, sign, zero and parity.

Figure 0 is a simplified functional block diagram of the 8085A-based computer system. The 8085A functions as the central processor unit (CPU). The memory is in two parts: the read-only memory (ROM) and the random access memory (RAM). The ROM device is the Intel 2816 integrated circuit which has a capacity of 2K bytes of memory, This memory is electronically erasable and data may be written to it at a slow rate by the CPU or manually by data and control switches. The RAM device is the Intel 8156 integrated circuit, which contains 256 bytes of memory. The 8156 also contains the input/output (I/O) section. This section consists of three ports: A, B and C. Ports A and B are 8-bit in length and port C is 6-bit long. The computer's built-in keyboard consisting of 16 data and 8 function keys can be used to input data to the I/O section. Outputs from the I/O section are directed to two 7-segment displays. In order to provide the ability for the computer to interface with an external device, the output of one of the two 8-bit I/O ports (Port B) is also directed to the 8 pins of a 16-pin integrated circuit connector. Attached to the address-data bus are also the address-data bus logic indicators and the data/control switches. The address-data bus logic indicators consist of 8 drive circuits each to drive a light-emitting diode (LED) to indicate jointly the logic state of the address-data bus. The data/control switches are used to write to or read from the ROM, RAM or I/O ports.

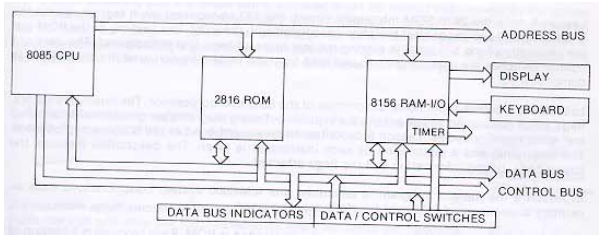


Figure 0. Block Diagram

The students constructed the 8085A-based computer system in a stepwise manner. In the following, we follow the construction sequence performed by the students.

III. CONSTRUCTION OF THE 8085A-BASED COMPUTER SYSTEM

A. Construction of the Power Distribution System

Student built the 8085A-based computer system on a pre-designed circuit board. The first step is to implement the power distribution system on to the circuit board. This was done by soldering the necessary capacitors and a voltage regulator on to the circuit board. Figure 1 is a simplified schematic of the computer system's power distribution system. Unregulated DC voltage in the range of 12-13V is brought in through connector S3 to the power ON-OFF switch SW16. Switch SW16 applies this voltage to filter capacitors C3and C14 and to voltage regulator IC7. The 5V output of the regulator is filtered by capacitors C4 through C12. Capacitor C4 is a 100 micro farad capacitor to filter out low frequency noise such as 60 Hz and capacitors C5 through C12 filter out any high frequency noise which may be generated by the rapid switch operation. After completion of this part of the circuit, students are directed to identify the correct voltage at various voltage-feeding points on the circuit board.

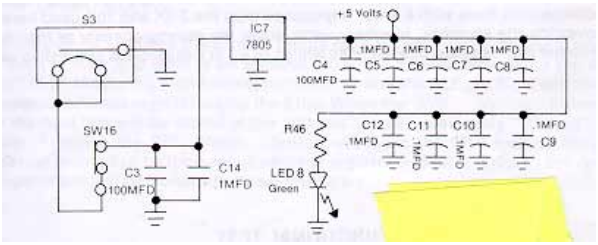
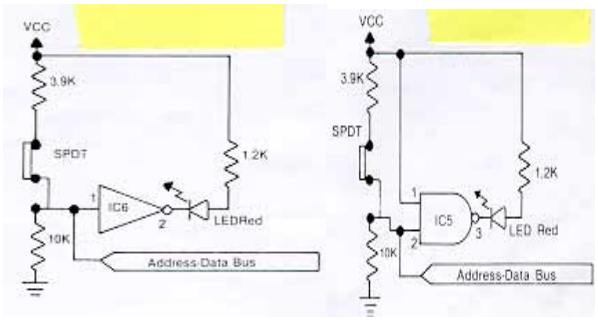


Figure 1. Power Distribution System

B. Implementing of the Address-Data Indicator circuit

To construct this part of the circuit, 8 SPDT switches and 8 LED have to be installed onto the circuit board. Figure 2A is the schematic diagram of address-data bus indicator circuit. The 74HC04 Hex inverter consisting of 6 independent inverter circuits each driving one LED logic level indicator. When a logic 1 is present on the data bus, a voltage of 3.5V is applied to the input of the inverter. This causes the output of the inverter to be a logic 0 corresponding to a voltage level of 0.2V from the ground. This causes a current of 2.5mA to flow through the corresponding LED. The condition of the LED to be lit is used to indicate a logic 1 level on the data line. When a 0 is present on the data bus, the input to the inverter circuit is 0.2V. The output of the inverter is 5V and no current can flow through the LED connected to this output. The LED remains dark indicating a 0 is present on the data line. The hex inverter thus provides a method of indicating the logic states on the address-data lines AD0-AD5. Since a NAND gate will be used later to make a de-bounce circuit, Two of the four NAND gates in a 74HC00 is used to drive the remaining 2 logic level indicators which are LED6 and LED7. The schematic diagram is shown in Figure 2B.



A B

Figure 2 (A. & B). Address-Data Bus Indicator

C. Implementing RAM

The Intel 8156 programmable interface adapter (PIA) consists both RAM and I/O on a single chip. Introducing this device onto the circuit board at this stage is to implement RAM. A 40-pin socket is first soldered onto the circuit board and the 8156 is inserted into the socket. Figure 3 shows the switch networks necessary for manual operation of the 8156 memory circuit. The IO/M switch is used to select the RAM or the input-output ports. For selecting RAM it should be in M position (switch down). A brief description of the functions of the lines each operated by a button (dimple switch) is given as follows:

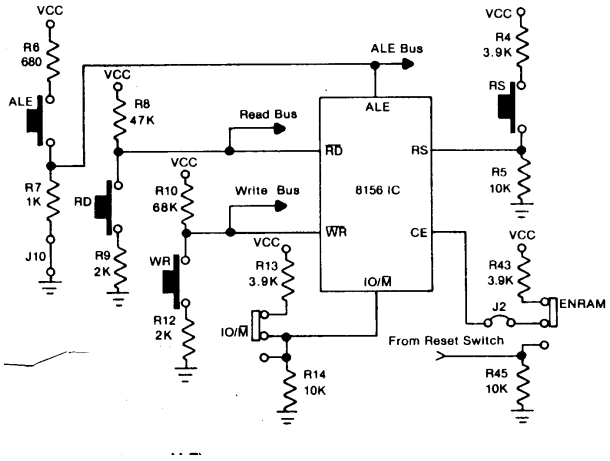


Figure 3. 8156 Switch Network

Address Latch Enable (ALE)

The ALE line is normally held near ground potential (0V). When the ALE button is pressed, the line is brought to a high voltage by the voltage divider R6 and R7. The logic 1 condition of the ALE pin of the 8156 enable the RAM to store the data on the address-data bus as an address.

Read from Memory (RD)

The RD pin on the 8156 is held at a high voltage by resistor R8. When the RD button is pressed, the logic level on the RD pin becomes 0. This enables a read operation from the RAM. The data in the memory location at the latched address is reproduced on the 8 data lines. The IO/M Switch must be in the M position in order to read from the RAM.

Write to Memory (WR)

When WR key is pressed, data on the data bus are written into the latched memory address. The IO/M Switch also must be in the M position in order to write to the memory.

Reset (RS)

This key is used to reset the 8156.

D. Implementing Input/Output and Display

When the IO/M switch is in the IO position, the 8156 passes the data on the data bus to input/output ports. Before data can be transmitted or received through the ports, the 8156 must be configured in terms of its input/output ports. Instructions are given to the 8156 by setting bit patterns in the Command Status Register. This register is located at address 0000 0000. Figure 4A shows how the Command Status Register operates.

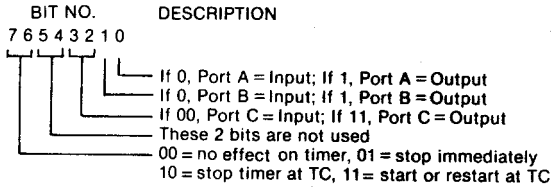


Figure 4A. Command Status Register

For example, if the binary pattern 1000 1110 is written to address location 0000 0000 when the IO/M switch is in the IO position, the 8156 will function as follows:

- The timer will stop when terminal count is reached
- Port A will be input port
- Both Port B and C will be output ports

After the ports are set, data are transmitted by first going to the address of the desired port. Next, data are output from the port pins and placed on the data bus if the port is configured as an output port. If the port is configured as an input port, data are input into the port pins. The address of each port is listed in Figure 4B.

ADDRESS	NAME
7654 3210	
XXXX X000	Command Register
XXXX X001	Port A
XXXX X010	Port B
XXXX X011	Port C

X-Indicates don't care, can be 1 or 0

Figure 4B.. Ports Addresses

In our construction, Both Display1 and Display2 are connected to Port B as illustrated in Figure 4C. However, they are time multiplexed. The multiplexing is controlled by bit 0 of Port C. To light any segment of the display, a logic 0 must be placed in the corresponding bit of Port B.

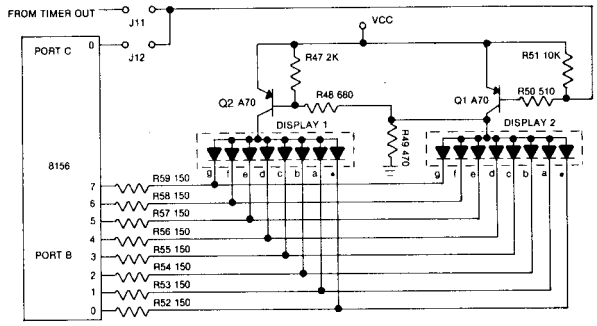
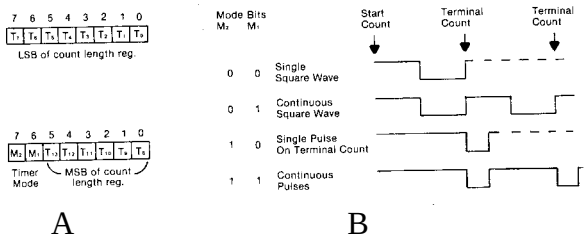


Figure 4C.. Display 1 & 2.

E. Implementing the Timer

The 8156 timer circuit is a 14-bit programmable counter. The counter is programmed by setting the two bytes of the Count Length Register as shown in Figure 5A. the Count Length Register bits T0 through T13 are counted down by TIMER IN pulses. When the terminal count is reached, the cycle is complete and a pulse or square wave is present at the TIMER OUT pin of the 8156 integrated circuit. Bits M1 and M2 of the Count Length Register specify the timer mode as shown in Figure 5B. The Count Length Register may be set to any value from 002H to 3FFFH (H stands for hexadecimal number).



A B

Figure 5 A & B. The Count Length Register

The LSB (least significant Byte) of the Count Length Register is at I/O address 04H. The MSB (most significant byte) of the Count Length Register is at I/O address 05H.

The start and stop of the counter is controlled by bits 6 and 7 (TM1 and TM2) of the Command Status Register.

If the timer is not running when the Count length Register is loaded, it will remain in not running condition until a new start command is issued. If the timer is running when a new mode or count length is loaded into the Count Length Register, the timer will continue to run with the old mode or count length until a new start command is issued through the Command Status Register.

To clock the timer a low-to-high transition must be applied to the TIMER IN pin of the 8156. De-bounce is accomplished by tying the TIM switch to a flip-flop circuit as shown in Figure 5C. Both pins 12 and 13 of integrated circuit 74HC00 must be high to make pin 11 go low. With the TIM switch in the down position, a low is applied to pin 13causing pin 11 hence pin 10 to be high. Pin 9 is biased high by resister 39. With both pin 9 and 10 high, pin 8 will be low. This applies a low to pin 12 and the TIMER IN through jumper J3. The low on pin 12 keeps pin 11 high even if the low is removed from pin 13. The flip-flop is now in the logic low state.

Write Enable (WE)

This line serves the same purpose as the **WR** line of the 8156 to allow data to be write into 2816. Figure 6 is the schematic diagram of ROM This line serves the same purpose as the **RD** line of 8156 allowing data to be read from 2816.

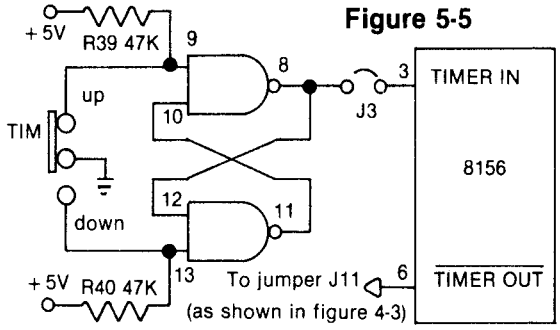


Figure 5C. Debounce Flip-Flop

To clock the timer, the TIM switch is moved to the up position. This breaks the ground connection to pin 13 causing it to be biased high by resistor R40. Next, pin 9 is connected to ground causing pins 8 and 12 to go high. Since pin 13 is high, pin 11 hence pin 10 goes low which forces pin 8 to stay high even if pin 9 goes high and low several times due to switch de-bounce. The flip-fop is now in the logic high state and the low-to-high transition has been sent to the TIMER IN pin of the 8156 to clock the TIMER.

F. Implementing ROM

As mentioned earlier, a 2816 EEPRM is used as the ROM. This is a 24-pin device. The write operation for this device requires 10 ms. The 2816 contains 2K memory locations, each capable of storing 1 byte data. The total memory capacity of 2816 is 16K.

Address Lines (A0-A10)

These 11 address lines tell the 2816 which of the 2K memory locations is to be written to or read from.

Input/output Lines (I/O0- I/O7)

During a write operation, data on these lines are stored in the memory location specified by the address on the address lines A0-A10. During a read operation, the data specified by the address lines are placed on the I/O lines.

Chip Enable (CE)

Output Enable (OE)
This line serves the same purpose as the **RD** line of 8156 allowing data to be read from 2816.

Write Enable (WE)
This line serves the same purpose as the **WR** line of the 8156 to aloow data to be written into 2816.

Figure 6 is the schematic diagram of ROM circuit. The 74HCT573 is an Octal-D type transparent latch which consist of 8 independent latches controlled by a control input at pin C. When C is high, the latches are transparent. A high on any of the inputs, D1-D8, will result in a high on the corresponding Q output. A low on a D input will likewise result in a low on the corresponding Q output. When C goes low, the data that was present in the latch is retained on the Q outputs regardless of any of the D input. The ALE bus is connected to the C input of the 74HCT573 integrated circuit. When the ALE key is pressed, the ALE bus goes high and the LSB of the address on the address-data bus is passed through the transparent latches to the address inputs A0-A7 of the 2816. When the ALE button is released, the address is retained at the A0-A7 inputs. The address-data bus is then free to pass data to or from the 2816 I/O pins. Address lines A8, A9 and A10 are connected to the appropriate pins of the 2816.

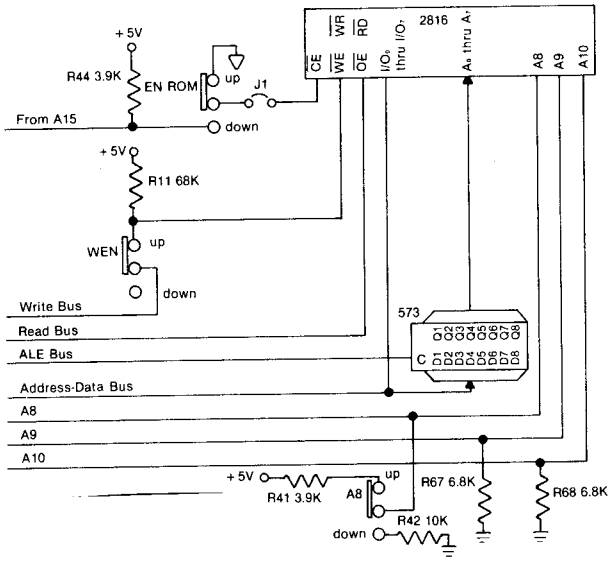


Figure 6. ROM Circuit

The Read bus is connected to the OE pin and is used to gate data from the 2816 onto the data bus. The WE input to the 2816 is tied to the top of the WEN switch. With the WEN switch in the up position, The WE input is connected to the write bus as shown in Figure. The WR button can then be used to write data into the 2816 memory. Pushing the WR button ties the WE input to a low voltage (logic 0) through the resistor network R10, R11 and R12. Releasing the WR button ties the WE input to a high voltage (logic 1) through resistors R10 and R11. If the WEN switch is in the down position, The WR button is disconnected from the WE input. The WE input is tied to high voltage (logic 1) through resistor R11 and writing is inhibited. To prevent accidental erasure of the contents in 2816, the WEN switch should always be in down position when power is turned ON or OFF.

The center of the Enable ROM (ENROM) switch is connected to the CE input of the 2816. With the switch in the up position, the CE is tied to a low voltage (logic 0). The 2816 will then respond to orders on the WE and OE lines. With the switch in the down position, the CE input is tied to a high voltage (logic level 1) through resistor R44 and the 2816 is inhibited from responding to orders on the WE and OE lines.

G. Implementing 8085A Microprocessor

The next thing is to install the 8085A chip onto the circuit board. This is done easily by first soldering a 40-pin socket onto the circuit board then inserting 8085A into this 40-pin socket. The internal structure of the 8085A has been described in the previous paragraph and will not be repeated here.

H. Implementing the Monitor Program

The monitor program resides in ROM locations 0000-0003 and 0400-012D. RAM locations 80DC-80FF are also used. All codes of the monitor program are entered manually via

data/control switches with the address-data bus logic indicators as a visual aid. This a long and tedious process. However, students gain efficiency by practice. Further, they build up better understanding of the machine language programming. Once all codes are entered without error, the monitor program is ready for running. When the RESET switch is down, the 8085A comes up running the monitor program whenever the power is on. With the monitor program running, the keyboard is capable of entering data or new programs into memory. Any byte in memory may also be sent to the display. By using the GO button on the keyboard, control can be transferred to an application program.

There are three modes of operation, Data (DA), Address Low (AL), and Address High (AH). The active mode is determined by the code in the MODE L byte. The MODE L byte is set to the low order byte of the address of DDA which is FC for DA mode, to the low order byte of the address of DAL which is FD for AL mode, and to the low order byte of the address of DAH which is FE for AH mode. The MODE H byte always contains 80 which is the high order address byte of DDA, DAL and DAH.

Data Mode

In data (DA) mode, the MODE H and MODE L bytes contain the address of the DDA byte which is at 80FC. The content of DDA is displayed in Display1 and Display2. The display operation is performed in the following way: When one of the keyboard data keys (0-F) is pressed, the data representing the key (a nible) is entered into the low order digit of DDA and displayed on Display2. When a new key is pressed, the low order digit of DDA is shifted into high order and displayed on Display1, the new data becomes the low order digit displayed on Display2. No decimal point is lit indicating DDA byte is displayed.

Address Low Mode

In AL mode, the mode H and mode L bytes contain the address, 80FC, of the DAL byte. Data from keyboard is now entered into DAL in the same manner as into DDA. The display is the same except that the decimal point of Display2 is lit indicating that the DAL byte is being used.

Address High Mode

In AH mode, the MODE H and MODE L bytes contain the address, 80FE, of the DAH byte. Operation in the AH mode the same as in AL mode with the decimal point of Display1 being lit indicating DAH byte is being displayed. When the function key STORE (ST) on the keyboard is pressed, the data in DDA is stored at the address specified by DAH and DAL. This address is automatically incremented by 1 internally with the monitor program running in DA mode displaying the data just stored.

When the function key Read (RD) on the keyboard is pressed, the content in the address specified by DAH and DAL is read and placed in the DDA byte. The address is then incremented by 1 and the monitor is in the DA mode to display the content read. When the function key Go (GO) on the keyboard is pressed, control is transferred to the instruction whose address is specified by DAH and DAL.

IV. INTERFACING WITH THE STEPPER MOTOR

The control of the motion of a stepper motor is by sending a 4-bit digital code sequence from Port B to the bases of four transistors. The collector of each of the four transistors is connected to the low potential terminal of each of the four stator windings with the emitter connected to ground. The high potential terminals of the four stator windings are connected together and a 5V-power supply is connected to this point.

The program for controlling the motion of the stepper motor is listed below. For clockwise rotation, the logic sequence is 0A, 09, 05, 06. The sequence is then repeated again and again. For counterclockwise rotation the sequence is the reverse. A time delay is needed between the executions of two adjacent sequential codes. This is done by calling a time delay subroutine during the execution of the stepper motor motion control program.

Program for Controlling a Stepper Motor

Time Delay Routine

8100	LXI HL	21
8101		FF
8102		FF
8103	DCR L	2D
8104	JNZ	C2
8105		03
8106		81
8107	DCR H	25
8108	JNZ	C2
8109		03
810A		81
810B	RET	C9

Data for Sequential Operation of Stepper Motor

8110		0A
8111		09
8112		05
8113		06

Main Program for Stepper Motor control

811F		00
8120	LX1 HL	21
8121		10
8122		81
8123	MOV AM	7E
8124	OUT	D3

8125	Port B	82
8126	CALL	CD
8127	Delay	00
8128		81
8129	LX1 HL	21
812A		11
812B		81
812C	MOV AM	7E
812D	OUT	D3
812E	Port B	82
812F	CALL	CD
8130	Delay	00
8131		81
8132	LXI HL	21
8133		12
8134		81
8135	MOV AM	7E
8136	OUT	D3
8137	Port B	82
8138	CALL	CD
8139	Delay	00
813A		81
813B	LXI HL	21
813C		13
813D		81
813E	MOV AM	7E
813F	OUT	D3
8140	Port B	82
8141	CALL	CD
8142	Delay	00
8143		81
8144	JMP	C3
8144		1F
8145		81

V. FURTHER WORK

The control of the stepper motor only needs 4 bits from Port B. The full use of the 8-bit output capability can be made by connecting the input of a DAC circuit to Port B. By programming the 8085, an analog waveform of a user's design can be generated at the output of the DAC circuit.

VI. CONCLUSION

This project enables students to integrate classroom and laboratory knowledge with project-based learning. It has allowed students to develop advanced technical skills by cross-linking two sets of contemporary microprocessor technologies, i.e., Motorola and Intel, in a compact and

efficient way. The development of such competencies is essential in order to function successfully in today's competitive electronic job market.

Our preliminary assessment indicates that although the project is challenging, it is not overwhelming or beyond electronics technology students capabilities. It has helped students to develop the troubleshooting and problem solving skills that are required in real life situations. This project has helped to demystify the inner workings of microcontroller technology and has generated and enhanced the students' interest in further exploration of microprocessor technology. Upon completion of the project, the students demonstrate a sense of accomplishment, confidence, and self-satisfaction.

References

Rafiguzzaman, M . (1996). Microprocessor and microcomputer-based system design. (2nd Ed.). New York, NY: CRC Press.

Staugaard, A . C. Jr. (1998). 8-bit microprocessor Interfacing and Applications, Heath Company.

Micro-Master - MM 8000, 8085 Microprocessor-Basic Systems Course *Computer Theory, Construction, and Programming*, Elenco Electronics, 1996.

JIANREN LI

Jian-ren Li obtained his Ph.D. degree in Solid-State Physics from King's College, University of London, UK, in 1985. He has worked in industry for more than ten years. He is currently teaching digital electronics courses in the Department of Engineering Technology at Western Kentucky University, Bowling Green, Kentucky.

AHMAD ZARGARI

Dr. Ahmad Zargari is Associate Professor of Industrial Education and Technology at Morehead State University. He teaches graduate and undergraduate courses in Microprocessors Programming and Applications, Data Communication, and Networking Technology. He is a Member of ASEE, a Senior Member of Society of Manufacturing Engineers (SME), a Certified Industrial Technologist (CSIT). His research interest is in microprocessor technology, data communication and networking, and statistical quality control (SQC).