Designing a Microprocessor Controlled Heater Fan for a Fireplace

Mohammad Fotouhi, Ali Eydgahi, Robert McCulley

University of Maryland Eastern Shore
Princess Anne, MD 21853

Abstract

This Paper describes the details of an undergraduate design project completed as part of the final senior design class for the Electrical Engineering Technology Program at University of Maryland Eastern Shore. The objective of this project was to use a PIC 16C622 microprocessor to control the speed of a fan depending on the temperature sensed just below the mantel of a fireplace. The PIC 16C622 was chosen because of its ease of programming, low cost, and compatibility with other components in the projects circuit. The student designed a complete circuit and developed a program to control the fan speed for a fireplace.

Introduction

The undergraduate major of Electronic Engineering Technology at the University of Maryland Eastern Shore requires each student to complete a design course[1-5]. The speed control of a fan dependent on temperature was one of the projects offered in this course.

The justification of this project was the need for this product in the consumer electronics industry. There is no product currently on the market that can be integrated into a consumer fireplace that will vary the fan speed according to the temperature below the mantel of the fireplace. When a fireplace is initially loaded with wood and started, the temperature below the mantel is at room temperature and there is no need to run the fan. As the wood begins to burn, the temperature below the mantel rises from room temperature (70 degrees F) to as high as 160 degrees F. At this point, over 70% of the heat is wasted behind the firebox and absorbed into the wall. This is the condition that a fan is needed to push the heat out of the firebox, away from the wall, and into the house. Likewise, when the fire burns out, the temperature begins to drop and there is no need to continue to run the fan after the temperature drops back to 70 degrees. It was therefore decided to develop a product that will control the fan to begin running at 50% speed at 90 degrees F. and ramp up to 100 % speed when the temperature reaches 130 degrees. The fan will continue to run at 100% as long as the temperature maintains 130 degrees or greater. This will insure that the heat generated by the fireplace will be pushed into the living space of the house where it is needed.

The Design Project

A conceptual block diagram was developed with the components needed to accomplish the project as shown in figure 1.

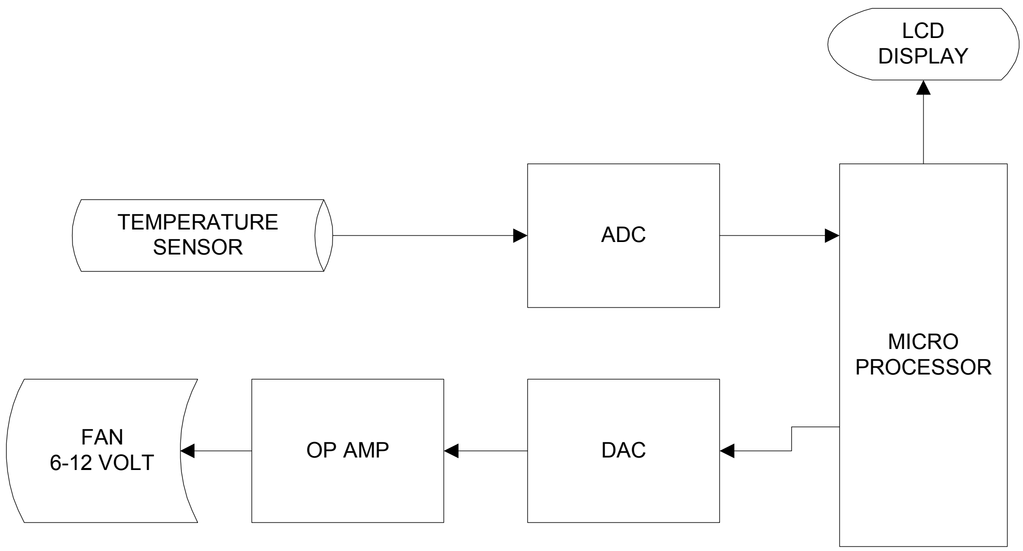

Fig. 1: Conceptual Block Diagram of the Design

I. The Fan

The first item to be considered was what device should be used to move the air. The two choices were a fan or a blower. Most products on the market today, use blowers that range from 50 to 120 Cubic Feet per Minute (CFM) to move air from the firebox. For this project, we used a 12^V fan due to the availability and cost. The fan specification is as follows:

- 12 Volts DC
- Voltage Range 6-13.8
- 3000 RPM
- 0.32 Amps
- 85 CFM
- Noise 48 dB Maximum
- Size 5"x 5" x .5"

II. The Temperature Senor

A LM34 temperature sensor was chosen due to its low cost and performance. The sensor gives an output of 0.01 volts per degree F. Therefore, the output from the LM34 at 100 degrees F. is 1 volt. The temperature range of interest for this project is 60-188 degrees F., which produces an output of 0.6 to 1.9 volts.

III. The Analog to Digital Converter

The ADC device chosen was the ADC8030. This device has an 8-bit resolution, which allows the temperature range to be broken up into 256 increments. The temperature range of 60 to 188 yields a maximum input of 1.9 volts which when divided by 256 increments gives a resolution of 0.00745 volts/ bit. This will be ample for the project. The reference voltage was therefore set up at 1.9 volts. The data sent out of the ADC is a single line of serial data to be fed into the microprocessor.

IV. The Digital to Analog Converter

The DAC used was the National Semiconductor DAC0832 which is a CMOS/Si-Cr 8-bit multiplying DAC which takes the 8-bit code from the PIC microprocessor and converts this into an analog voltage output. The output of the DAC will be 2.1^V to 5.51^V, which has to be amplified by an op amp to provide up to 12.7 volts at 0.32 amps to the fan. It should be noted that the output could be 0 but we don't turn on the system until we see 90 degrees, which is then 2.1^V at the DAC and then amplified to 6^V.

V. Transistor and Regulator

An ECG33L NPN transistor was used to switch the higher current of up to 1 amp to run the fan at a maximum voltage of 13 volts. A 5^V regulator LM7805 was also used to feed the chips that operate at V_{cc} of 5 volts.

VI. The Microprocessor

The PIC 16C622 was chosen for this project. The PIC family breaks up into three main groups as follow:

- 12-bit instruction core (16C5X, 12C5XX, 12CE5XX)
- 14-bit instruction core (16C55X, *16C62X*, 16C6X, 16C7X, 16C71X, 16C8X, 16F8X, 16F87X, 12C6XX, 16C9XX, 14C000)
- 16-bit instruction core (17C4X, 17C7XX)

All three groups share the same core set of RISC instructions with additional instructions available on the 14 and 16 bit cores. This means that assembly code written for the 12-bit family can be easily upgraded to work on a 14 or 16-bit core part. This is one of the great advantages to the PIC. All instructions, except *branch* and *go to* instructions, execute within 1 clock cycle (crystal freq. / 4), which makes it easy to check the execution timing. The ease of programming requires only 33 instructions to be learned. The followings are the PCB instruction set:

- *ASM..ENDASM - Insert assembly language code section.*
- *BRANCH - Computed GOTO (equivalent to ON..GOTO).*
- *BUTTON - Debounce and auto-repeat input on specified pin.*
- *CALL - Call assembly language subroutine.*
- *EEPROM - Define initial contents of on-chip EEPROM.*
- *END - Stop execution and enter low power mode.*
- *FOR..NEXT - Repeatedly execute statement(s).*
- *GOSUB - Call BASIC subroutine at specified label.*
- *GOTO - Continue execution at specified label.*
- *HIGH - Make pin output high.*
- *I2CIN - Read bytes from I2C device.*
- *I2COUT - Send bytes to I2C device.*
- *IF..THEN - GOTO if specified condition is true.*
- *INPUT - Make pin an input.*
- *LET - Assign result of an expression to a variable.*
- *LOOKDOWN - Search table for value.*
- *LOOKUP - Fetch value from table.*
- *LOW - Make pin output low.*
- *NAP - Power down processor for short period of time.*
- *OUTPUT - Make pin an output.*
- *PAUSE - Delay (1mSec resolution).*
- *PEEK - Read byte from register.*
- *POKE - Write byte to register.*
- *POT - Read potentiometer on specified pin.*
- *PULSIN - Measure pulse width (10us resolution).*
- *PULSOUT - Generate pulse (10us resolution).*
- *PWM - Output pulse width modulateaad pulse train to pin.*
- *RANDOM - Generate pseudo-random number.*
- *READ - Read byte from on-chip EEPROM.*
- *RETURN - Continue execution at statement following last executed GOSUB.*

- *REVERSE - Make output pin an input or an input pin an output.*
- *SERIN - Asynchronous serial input (8N1).*
- *SEROUT - Asynchronous serial output (8N1).*
- *SLEEP - Power down processor for a period of time (1 Sec resolution).*
- *SOUND - Generate tone or white-noise on specified pin.*
- *TOGGLE - Make pin output and toggle state.*
- *WRITE - Write byte to on-chip EEPROM.*

Figure 2 shows the pin layout of the PIC 16C622.

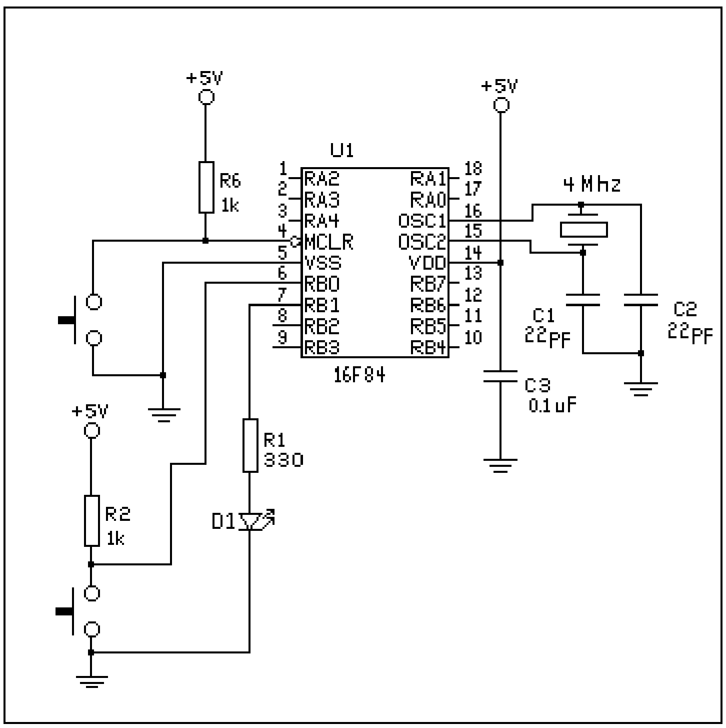

Fig. 2: The PIC 16C622 Pin Layout

VII. The Operational Amplifier

Because the input to the fan must be either 0 volts, or 6 to12.7 volts DC, an op amp will have to be used to increase the voltage from the DAC. The operational amplifier chosen was the LM741CN which is a general purpose operational amplifier which will amplify the signal from the DAC by 2.76 times using the 10^K ohm and 8.5^K ohm resisters as shown in figure 3.

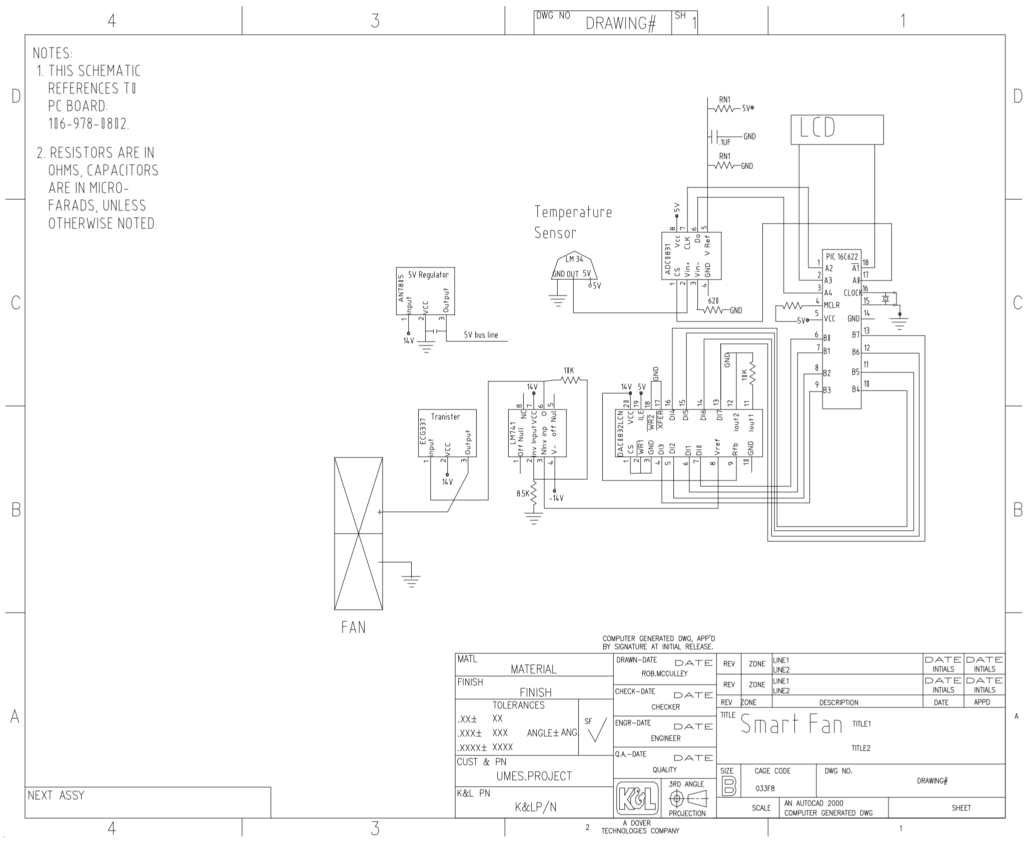

Fig. 3: Schematic of the Design

VIII. Software Development

The following program controls the fan speed depending on the possibility of one of the following conditions:

- Temperature less than 60 degrees F. Do nothing. Program does not initiate. No display.
- Temperature between 60 degrees F. and 90 degrees F. Show temperature on display but do nothing else.
- Temperature less than 90 degrees F and 130 degrees F. Turn on Fan at 50% speed (6volts) at 90 degrees and ramp up to full speed (12 volts) at 130 degrees. Display temperature while doing this.
- Temperature greater than 130 degrees but less than 188. Continue to run fan at full speed and display temperature.

- Temperature greater than 188 degrees. Continue fan at full speed but display goes blank.

```
CMCON = 7
DEFINE  OSC     4
TRISA = 0
TRISB = 0
TRISA.4 = 1

HIGH PORTA.0
LOW PORTA.1
LOW PORTA.3
LOW PORTB

TEMP    VAR     BYTE
CNT     VAR     BYTE
CNTT    VAR     BYTE
NUM     VAR     BYTE
KNT     VAR     BYTE
KNTT    VAR     BYTE
SEG     VAR     BYTE
OUT     VAR     BIT[7]
LCD     VAR     BYTE[10]
CTL     VAR     WORD
DUMCTL      VAR   WORD
OCTL  VAR   WORD
OTEMP   VAR     WORD
DUMTEMP     VAR   WORD
OUTTEMP     VAR   WORD
DISPL VAR     BYTE

LCD[0] = %10111111
LCD[1] = %10000110
LCD[2] = %11011011
LCD[3] = %11001111
LCD[4] = %11100110
LCD[5] = %11101101
LCD[6] = %11111101
LCD[7] = %10000111
LCD[8] = %11111111
LCD[9] = %11100111

READTEMP:
PAUSE 1
LOW PORTA.0
PAUSE 1
SHIFTIN PORTA.4,PORTA.2,2,[TEMP\9]
HIGH PORTA.0

CALCULAT:

DUMTEMP = (TEMP*500)
OTEMP = DIV32 1000
OUTTEMP = OTEMP + 60
```

```
IF OUTTEMP < 60 THEN UNDER
IF OUTTEMP < 90 THEN MID
IF (OUTTEMP >= 90) AND (OUTTEMP <= 130) THEN CALC
IF OUTTEMP > 130 THEN OVER

UNDER:
DISPL = 0
CTL = 0
GOTO OUTCTL

MID:
DISPL = OUTTEMP
CTL = 0
GOTO OUTCTL

CALC:
DISPL = OUTTEMP
DUMCTL = (3175*OUTTEMP)
OCTL = DIV32 1000
CTL = OCTL - 158
GOTO OUTCTL

OVER:
DISPL = 0
CTL = 255
GOTO OUTCTL

OUTCTL:

PORTB = CTL

DISPTEMP:

FOR CNTT = 3 TO 1  STEP -1
  CNT = CNTT - 1
  NUM = DISPL DIG CNT
  SEG = LCD[NUM]
  OUT[0] = SEG.0
  OUT[1] = SEG.1
  OUT[2] = SEG.2
  OUT[3] = SEG.3
  OUT[4] = SEG.4
  OUT[5] = SEG.5
  OUT[6] = SEG.6
    FOR KNTT = 8 TO 2 STEP -1
    KNT = KNTT - 2
    HIGH PORTA.1        'CLOCK'
    PAUSEUS 5
    PORTA.3 = OUT[KNT]  'DATA'
    PAUSEUS 5
    LOW PORTA.1
    NEXT KNTT
  NEXT CNTT

PAUSE 500
GOTO READTEMP
```

Conclusion

Controlling the heater fan speed with the microprocessor was achieved. The program allowed the fan to run at various speeds depending on the 5 conditions mentioned earlier. Since most recent fireplaces are made for fans, there should be no concern about carbon monoxide issues with this smart fan design. The air blows behind the fire box and never is combined with the burning chamber. The target cost for this project was $20.00 with the market value of $50.00. Future improvements to this project would be to:

- Design a PCB to secure components and fan on one carrier.
- Optimize Fan/Blower component for cost and performance.
- Design on board power supply at +/- 15 volts.
- Investigate operator interface option and various set points.

This project provided much experience in controlling a motor (fan) through the design and fabrication of microprocessors and the supporting components. The student learned how to transfer an idea from concept, to schematic, and then to implementation of an actual working device that is not only innovating, but provides a useful product to be used in society. The microprocessor controlled heater fan will provide a product to society that not only is convenient to the consumer, but will also conserve energy and provide a safer environment for those who use fireplaces as a source of heat.

References

1. Fotouhi, M., Eydgahi, A., and Schneider, M.V.; *Designing an EMS Emergency Beacon,* Computers in Education Journal, Vol. 14, No 2, April/June, 2004.
2. Fotouhi, M., Eydgahi, A., and Herz, D.A.; *Using C++ Language to Design a Robot Controller,* Proceedings of ASEE Mid Atlantic Section Fall Conference, Baltimore, MD, October 24-25, 2003. CD-ROM
3. Fotouhi, M., Eydgahi, A., and Malaby, T.*; Design of a Rain-Based Speed Controller for Automobile Windshield Wiper,* Proceedings of ASEE Annual Conference, Montreal, CANADA, June 16–19, 2002. CD-ROM
4. Fotouhi, M. and Eydgahi, A.*; Using a PLC Trainer to Control a Utility Cart,* Proceedings of ASEE Annual Conference, Albuquerque, NM, June 24–27, 2001. CD-ROM
5. Fotouhi, M., Eydgahi, A., and Wagner, J.; *Microprocessor Controlled Milling Machine: A Student Project*, Proceedings of ASEE Annual Conference, St. Louis, MO, June 18 –21, 2000. CD-ROM
6. Wray, W.C., Greenfield, J.D., and Bannatyne, R.T.; “Using Microprocessors and Microcomputers: The Motorola family,” Fourth edition, Prentice Hall, N.J., 1998.

MOHAMMAD FOTOUHI
Dr. Fotouhi is a Professor of electrical engineering technology at University of Maryland Eastern Shore. He received his Ph.D. in power System Engineering from University of Missouti-Rolla, M.S. from Oklahoma State University and B.S. from Tehran Polytechnic College. He has been conducting a practical research on the growth and characterization of the dilute magnetic semiconductor since 1985. He is a member of Eta Kappa Nu Honor Society. He was chairman of Student and Industry Relation and Host Committee member of IEEE Conference on Power Systems Computer Application in 1991. He also was chairman of Student Relation and Host Committee member of the IEEE Power Society Winter Meeting in 1996.

ALI EYDGAHI
Dr. Eydgahi is a Professor of electrical engineering at University of Maryland Eastern Shore. He received his Ph.D. and M.S. in Electrical and Computer Engineering from Wayne State University. Since 1986 and prior to joining University of Maryland Eastern Shore he has been with the State University of New York, University of Tehran, Wayne County Community College, and Oakland University. Dr. Eydgahi is recipient of the Dow Outstanding Young Faculty Award from American Society for Engineering Education in 1990, and the Silver Medal for outstanding contribution from International Conference on Automation in 1995. He is the ASEE Campus Representative at UMES and has served as a regional and chapter chairman of IEEE and SME in New York. He also has served as a session chair and a member of scientific and international committees for many international conferences. He has published more than ninety papers in refereed international and national journals and conference proceedings.

ROBERT MC-CULLEY
Mr. McCulley graduated from University of Maryland Eastern Shore with a bachelor of science degree in Mechanical Engineering Technology in 1999, and Electrical Engineering Technology in 2002. He also graduated with a bachelor of science degree in Information Systems Management from University of Maryland Baltimore County in 1984. Mr. McCulley is a manufacturing engineer for K&L Microwave in Salisbury, MD which manufactures Radio Frequency Filters for the telecommunications and military markets.