MONITORING TEMPERATURE USING A MICROCONTROLLER WITH A HIGH-LEVEL LANGUAGE

Dogan Ibrahim
e-mail: dogan@dircon.co.uk, Near East University, PK670, Lefkosa, TRNC

Key words: Temperature monitoring, microcontroller based design

ABSTRACT

Temperature monitoring and control are very important in most modern process control applications. This paper describes the design of a microcontroller based temperature monitoring system with a digital display. The paper shows the simplicity of designing such a system using a high-level programming language instead of the usual assembly language.

I. INTRODUCTION

Temperature monitoring and control are vital in many industrial processes. Accurate control of temperature is essential nearly in all chemical processes. In some applications, an accuracy of around 5-10ºC may be acceptable. There are also some industrial applications which require better than ±1ºC accuracy.

Temperature is measured using temperature sensors. These sensors come in many different forms and a number of techniques have evolved for the measurement of temperature. There are new forms of sensors that require no contacts with the medium whose temperature is to be sensed. The majority of sensors still require to touch the solid, liquid, or the gas whose temperature is to be measured. Four technologies are currently in use: thermocouples [1] (TC), thermistors [2,3], resistance temperature detectors [4,5] (RTD), and integrated circuit (IC) sensors [6].

Most temperature sensors produce an analog output voltage which is proportional to the temperature. In some sensors (e,g, thermocouples) this voltage is in the microvolt range and special techniques are employed to measure this voltage and hence the temperature correctly. Analog sensors require analog-to-digital (A/D) converters so that the output voltage can be converted into digital form, suitable for interfacing to a digital computer.

Some temperature sensors (e.g. some IC sensors) produce digital outputs and these sensors can directly be connected to microprocessors or microcontrollers without the need for A/D converters.

Microcontrollers are widely used in many commercial and industrial applications. They are traditionally programmed using the assembly language of the target microcontroller. High-level programs are now available for most popular microcontrollers and this makes the program development and maintenance a much easier task.

This paper is about the measurement of temperature using an analog temperature sensor and a serial A/D converter circuit. A low-cost PIC microcontroller is used with a digital display. The microcontroller is programmed using a high-level programming language. The advantages of using a high-level language instead of the conventional assembly language are discussed in the paper.

II. THE HARDWARE

It was decided to use a low-cost microcontroller and a PIC 16F84 [7] type 8-bit microcontroller was chosen. This microcontroller has the following features:

- 1K flash program memory
- 36 bytes of data RAM
- 64 bytes of erasable EEPROM
- Four interrupt sources
- 13 input/output sources
- Watchdog timer
- Power saving sleep mode

PIC 16F84 contains a flash program memory which can easily be programmed using a suitable programmer device. There is no need to erase the memory with an UV light. This makes the development and testing an easy and a relatively quick task.

The block diagram of the temperature measurement circuit is shown in Fig. 1. The complete circuit diagram is shown in detail in Fig. 2. Temperature is sensed using a LM35DZ [6] type analog sensor. LM35DZ is a simple but accurate 3-pin temperature sensor IC. Pin 1 of the device is connected to the power supply (+5V), pin 3 is connected to the ground. Pin 2 is the output and this output provides a voltage which is directly proportional to the ambient temperature. The device can measure temperature from 2ºC up to 100ºC (some types can measure a wider range) and the output voltage to temperature relationship is 10 mV/ºC. For example, at 20ºC the output voltage is 200 mV. The output of the temperature sensor IC is connected to a ADC0831 [6] type serial A/D converter. The reason for using a serial converter instead of a parallel one was to minimize the pin usage. ADC0831 is basically an 8-pin device and interfaces to a microcontroller via 2 pins: CLK and DO. CLK is the clock pin and data is output from the DO pin

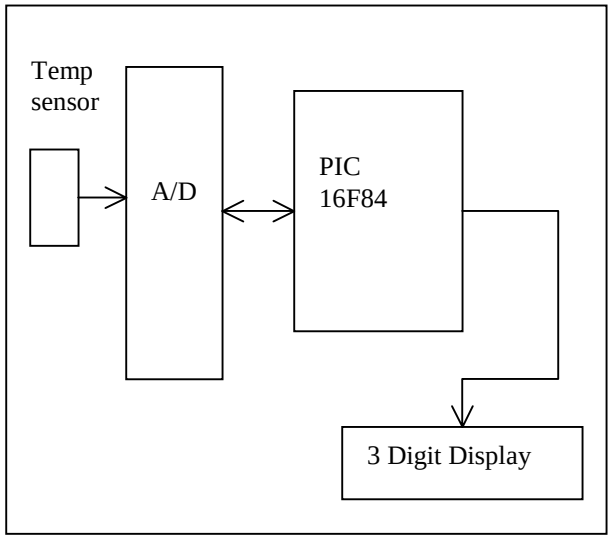

Figure 1. Block diagram of the system

when the CLK pin is pulsed. CS is the chip-select input and this pin should be LOW to start a conversion. Vref is the A/D converter reference pin and it should be connected to a +5V supply. Vin is where the analog input signal is connected to.

The output of the A/D converter is connected to bit 6 of Port B (RB6) of the microcontroller. The microcontroller is driven from a 4 MHz crystal. It receives the temperature in digital form, processes this data and then displays the temperature on 3 TIL311 type alphanumeric displays. The data inputs of all the displays are connected to port pins RB0-RB3 of the microcontroller. The LATCH inputs of each display digit is controlled independently from pins RA0-RA2 of the microcontroller.

III. THE SOFTWARE

The operation of the temperature measurement system is very simple. Analog temperature is sensed by the temperature sensor. This temperature is then converted into digital format by the A/D converter. The microcontroller receives this data and displays the temperature on 3 digital displays. The above process is repeated every second.

PIC Basic Pro [8] language is used for the program development. This is a very powerful PIC microcontroller high-level programming language, developed by the Micro Engineering Labs, Inc. [8] This language is similar to the standard BASIC language but it has advanced features which makes it an ideal tool to develop microcontroller based applications. High-level code is written using a standard program editor on a Personal Computer (PC). The code is then compiled on the PC and is downloaded to the target PIC microcontroller. PIC 16F84 microcontroller has built-in flash program memory and is programmed using a low-cost programming device.

The software algorithm of the temperature measurement system is described below as a pseudocode:

```
BEGIN
        Initialize variables
        DO FOREVER
                Start A/D conversion
                Wait for conversion complete
                Display temperature
                Delay for 1 second
        ENDDO
END
```

The complete program listing is given in Fig. 3. Display latches latch_msd, latch_middle, and l;atch_lsd are assigned to Port A pins 0, 1 and 2 respectively. ADC0831 pins CLK, DO and CS are then assigned to Port B pins. Variables first, second, and third store the MSD, middle, and the LSD digits of the display. The main program starts with label loop. Here, the A/D conversion is started by lowering the CS input. The SHIFTIN command is then used to send out clock pulses and then read serial data. Once the A/D

conversion is complete, the program calls to subroutine DISPLAY_TEMPERATURE to display the temperature on the digital displays. After a second delay, the program jumps back to label loop to repeat the process.

```
‘The program reads the analogue temperature,
‘converts it to digital form and then displays on
‘digital displays every second.
'
Include "MODEDEFS.BAS"
'Define the display latches
latch_msd       var PortA.0       'MS digit latch
latch_middle    var PortA.1       'Middle latch
latch_lsd       var PortA.2       'LSD digit latch
'
'Define ADC0831 connections
CLK             var PORTB.7       'Port RB7
DO              var PORTB.6       'Port RB6
CS              var PORTB.5       'Port RB5
'
'Define display registers
first           var byte          'MSD digit
second          var byte          'Middle digit
third           var byte          'LSD digit
'
'Define other variables
new_data        var byte          'A/D data
temp            var byte
secs            con 1000          ‘Seconds delay
'
'START OF MAIN PROGRAM LOOP
'
TRISA = 0
TRISB = %01000000

loop:
      CS = 0
         'Start conversion
      SHIFTIN DO,CLK,MSBPOST,[new_data\9]
      CS = 1
         'Stop conversion
      GOSUB    DISPLAY_TEMPERATURE
         'Display data
      PAUSE secs
         'Delay a second
      GOTO loop
         'Go back and repeat

' This subroutine displays the data on 3 TIL311
‘ type alphanumeric displays.
‘
DISPLAY_TEMPERATURE:
      new_data = 2 * new_data
      first = new_data / 100
      temp = new_data // 100
      second = temp / 10
      third = temp // 10
      PORTB = first
      latch_msd = 0
      latch_msd = 1
      PORTB = second
      latch_middle = 0
      latch_middle = 1
      PORTB = third
      latch_lsd = 0
      latch_lsd = 1

      RETURN
END
```

Figure 3. High-level program listing of the temperature measurement system

IV. CONCLUSIONS

The paper has described the design of a microcontroller based temperature measurement system. The system senses the temperature and displays it on an LCD display every second.

The software has been developed using the PicBasic Pro programming language, which is a microcontroller specific version of the well known BASIC language. It is shown that the programming is a relatively easy task when a high-level language is used. The same project would have taken much longer time to develop and test if assembler language was to be used.

REFERENCES

1. “Guide to Thermocouple and Resistance Thermometry”, TC Ltd, PO Box 130, Uxbridge, UB8 2YS, England.
2. Potter, D. “Measuring Temperature With Thermistors – a Tutorial”, National Instruments Corp. App. Note. 065, 1996.
3. “Thermistor Applications”, Web site: www.wuntronic.de/sensors/therm_appl
4. “Measuring Temperature With RTDs - a Tutorial”, National Instruments Inc., Appl. Note. 046, Nov. 1996.
5. “RTD Sensors”, Web site: www.thermometricscorp.com
6. National Semiconductor Inc., Web site: www.national.com
7. Microchip technology Inc., Web site: www.microchip.com
8. “Pic Basic Pro Compiler”, MicroEngineering Labs Inc. Web site: www.melabs.com

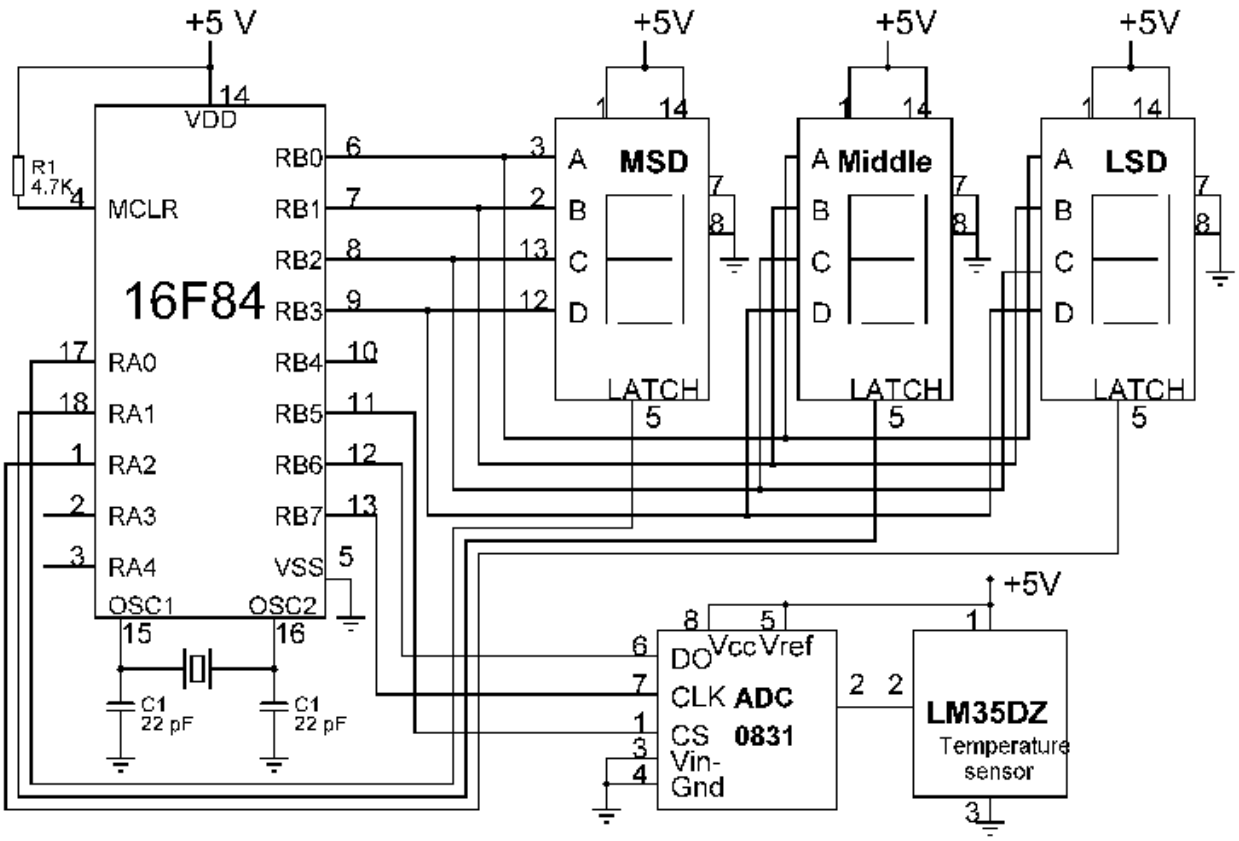

Fig 2. Circuit diagram of the temperature measurement system