A MICROCONTROLLER BASED DIGITAL THERMOMETER WITH TIMER (DIGITHERMO)

A. Abayomi-Alli [1] E. J. Etuk [2] P.I. Ezomo[3] F. A. Izilein[4] [5]A.Y. Akingboye and K. B. Erameh[6]

[1] Federal University of Agriculture Abeokuta, Ogun State, Nigeria.
[2] Northumbria University, United Kingdom.
[3, 4, 5] Igbinedion University Okada, Edo State, Nigeria.
[6] University of Benin, Edo State, Nigeria.

Abstract

Using conventional thermometers for measuring temperature will require a separate instrument for measuring time such as stop clocks, ordinary watches, or digital timers. These thermometers are fragile; prone to measurement errors, contain hazardous material that can burn the skin, eyes, and respiratory tract if spilled.

A microcontroller based digital thermometer with timer (DigiThermo) was designed and constructed. The device employs the AT89C4051 CMOS microcontroller (MCU), interfaced with the CA3162 ADC and a 16 x 1 character LCD display. Temperature is measured with a precision IC linear temperature sensor (LM35D) and time is counted using the MCU's timer circuits. The circuit was assembled on a prototype board, tested, modified and finally assembled on a set of matrix boards, and cased in a portable, stylish plastic casing with the sensor attached to a 28.0 cm long probe.

Results during testing showed that the device displays time count in seconds and temperature in degrees Celsius. The device can be used in the chemistry and engineering laboratories as well as in industrial, agricultural and in other applications requiring simultaneous temperature/time measurements.

Keywords: DigiThermo, Firmware, Microcontroller, Programming, Temperature, Thermometer, Timer.

1. Introduction

Chemistry and Chemical Engineering Laboratories typically employ mercury-in-glass thermometers for measuring temperature and a separate instrument for measuring time such as stop clocks, ordinary watches, or digital timers. However, mercury-in-glass thermometers must be handled with extreme care as they are fragile, contain mercury which is a hazardous material that can cause burns to the skin, eyes, and respiratory tract if spilled from a broken thermometer. These thermometers can explode if mistakenly used in a reaction whose temperature exceeds its range. Again, reading errors, such as errors due to parallax can occur with the use of these thermometers thereby introducing errors to measurements made with them. A similar problem of

reading errors also occurs with time measurements depending on the instrument used. Not does the use of a separate timing device inconvenient for the user, it also introduces error in some reaction kinematics measurements. DigiThermo is designed to solve these problems by incorporating these two measuring devices in one. The use of precision temperature sensor and microcontroller to perform computations will eliminate errors thereby enhancing the device's accuracy, increase flexibility and programmability – meaning the product can easily be modified to measure temperature in degrees Fahrenheit, for instance, by changing the application program, rather than redesigning the electronic circuit. It would also eliminate the aforementioned hazards associated with the mercury-in-glass thermometer.

Electronic thermometers have been built which serve as alternatives to mercury-in-glass thermometers. These employ sensors whose electrical properties vary in some way with temperature change. These temperature sensors combined with signal conditioning elements, signal processing elements and data presentation / display elements form an electronic thermometer which could be analog or digital. Digital thermometers [1] are temperature-sensing instruments that are portable, have permanent probes, and a digital display. They are typically battery powered. DigiThermo is designed primarily for use in Chemical Engineering Laboratories and the Chemistry Laboratories for measurement of temperature within the range of 0-100˚C as well as measure time in 'seconds' unit, the standard unit of time used in scientific and engineering calculations. It aims to replace the conventional mercury-in-glass thermometer and other timing devices used in these laboratories.

The design and construction of the microcontroller based digital thermometer and timing device (DigiThermo) illustrates the use of 'C' language in the programming of embedded systems / microcontrollers, the use of dual

IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 2, No 3, March 2013
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org

slope converter, LCD interfacing, as well as digital filtering. Additionally, it shows that embedded computers can be used as integral components of electronic circuits to control and/or provide accurate and flexible alternatives to designs/devices that had hitherto been realized using discrete (logic) components, analog electrical/ electronic circuits, mechanical designs as well as reliance on physical properties (as in the case of the mercury-in-glass thermometer) to produce simpler, yet versatile and powerful devices most of which solve many of the problems associated with the already mentioned techniques.

2. Design Methodology

The design of the DigiThermo is divided into two: Hardware and Firmware (Software).

2.1 Hardware Design

The circuit of the DigiThermo is made up of the following units: power supply unit, sensing unit, processing unit, display unit. The power supply unit will not be discussed in this paper because it is considered basic.

2.1.1 Design of the Sensing Unit

The temperature sensor chosen for this device is an IC temperature sensor, the LM35DZ (TO-92 package) from the LM35D series of precision integrated circuit temperature sensors. Its output voltage is linearly proportional to the Celsius (Centigrade) temperature. It is chosen for its low output impedance, linear output, and precise inherent calibration that make interfacing very easy. Its features and specifications are detailed in [2]. However we consider the output of LM35D

$$V_{OUT} = T_A \text{x } 10\text{mV} \quad (1)$$

Where T_A is the ambient air / surface temperature.
Output range of LM35D
Temperature range: 0°C to +100°C
Voltage range: 0 x10 mV to +100 x 10 mV
0 mV to +100 mV

$$\text{Supply Voltage: } 4\text{V} \leq V_S \leq 30\text{V} \quad (2)$$

but our chosen supply voltage, V_{CC} = 5V. This satisfies the condition in Eq. (2).

2.1.2 Design of Processing Unit

The processing unit is made up of an ADC and a microcontroller. A resistor, R_1 and a capacitor, C_1 connected in series are used to provide a first-order low-pass anti-aliasing filter used as a front end hardware filter. Since the information coming from the IC temperature sensor is coded in the time domain, edge sharpness and noise removal is important for proper sampling as opposed to removal of frequencies above the Nyquist frequency which is the case for information represented in the frequency domain.

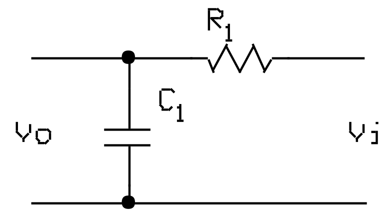

Figure 1: Anti-aliasing Filter Circuit

To provide for a quick step response [3], the value of C_1 is selected as 0.022 µF and that of R_1 as 100 KΩ.

$$\text{Time Constant, } T = RC \quad (3)$$

$T = 100{,}000 \text{ x } 0.022 \text{ x } 10^{-6} = 0.0022$ s = 2.2 ms.
The voltage across the capacitor rises to 96 % of input voltage in 3 time constants [4], or 3 x 2.2 ms = 6.6 ms which is appropriate for our application. Cut-off frequency of filter:

$$f_C = 1 / 2\pi RC \quad (4)$$

$f_C = 1/(2 \text{ x}\pi\text{x}100{,}000\text{x}0.022\text{x}10^{-6}) = 72.34315595$ Hz
This is approximately 72 Hz. The ADC chosen is the CA3162 chip by Intersil. It is a 3-digit DVM (digital voltmeter) that employs dual-slope integrator providing 10Hz sampling rate. Its features and datasheet [4] specifications are as follows: I^2L monolithic A/D converters, dual slope A/D conversion, 3 digit multiplexed BCD output, ultra stable internal band gap voltage reference, capable of reading 99mV below ground with single supply, differential input, internal timing - no external clock required, choice of low speed (4Hz) or high speed (96Hz) conversion rate, "Hold" inhibits conversion but maintains delay, and ability to indicate Over-range.

The ADC requires adjustments to improve its accuracy. From its datasheet, this is achieved with the aid of a potentiometer connected across pins 8 and 9 to provide ZERO adjustment and Variable resistor connected between pin 13 and ground for gain control / adjustment.

The potentiometer, VR_1, chosen for the zero adjustment is 50 KΩ as shown in the typical application circuits of the CA3162, whileVR_2 (used as a variable resistor) is 10 KΩ chosen for gain adjustment based on data obtained from the typical application section of the CA3162 datasheet.

The chosen value for the integrating capacitor, C2 is 330 nF (0.33µF). This is a commonly available value and it is just slightly higher than the 0.27 µF value suggested in the application circuits of the CA3162. The capacitor is a polyester type as emphasised in the datasheet.

The value chosen for C_3 is 0.1 µF as illustrated in the ADC datasheet. It is to be placed as close as possible to the CA3162E's ground and power pins. Each BCD output of the CA3162E is an open collector output requiring a pull-up resistor to interface it to CMOS logic. The value chosen for the pull-up resistors R_2, R_3, R_4, and R_5 is 20 KΩ each. Maximum current flows through each resistor when the BCD output is in the high state (negative logic according to the CA3162 datasheet), i.e. 0V. Maximum current through each resistor is:

$$V_{CC} / R_{pullup} = 5/20000 = 25mA$$

The power across the resistor is:

$$I^2R = 0.00025^2 \times 20000 = 1.25mW$$

Thus this current is safe for standard ¼-watt carbon resistors. Similar to the BCD outputs of the CA3162E, each Digit Select output pin (MSD, NSD, and LSD pins) requires pull-up resistors to interface to CMOS logic. The value chosen for each of the pull-up resistors R_6, R_7, R_8 is 50 KΩ. Maximum current flows through this resistor when the BCD output is in the high state, i.e. 0V. Maximum current through each resistor is:

$$V_{CC} / R_{pullup} = 5/50000 = 0.1mA$$

The power across the + resistor is:

$$I^2R = 0.00025^2 \times 20000 = 0.5mW$$

Thus this current is safe for standard ¼-watt carbon resistors. To ensure that the digit select signal received by the MCU emanates from the CA3162 (and not a low signal caused by fluctuation in power supply, etc), a capacitor is connected in series with the each digit select pull-up resistor. This keeps the input voltage to the MCU input pin at 5V even if the power source was disconnected momentarily until the digit select output pulls it low. A short time constant of 0.5 ms is chosen.
For C_4, C_5, and C_6 consider Eq. (3), Time Constant,
T = RC, Thus, C = T/R
$C = 0.0005 / 50000 = 1.0 \times 10^{-8}$ F $= 0.01 \times 10^{-6}$F = 0.01 µF
A ferrite bead, FB_1, is used to keep digital signal noise (originating from the ADC) from corrupting the analog signal path. This is because in the case of this design, the digital and analog supplies are the same. The ferrite bead chosen is the FAIR RITE 27430011112 suggested in the Intersil application note AN9214.2 [5], 'Using Intersil High Speed A/D Converter'. The datasheet [6] shows the specifications for the fair rite 27420011112. The ADC conversion rate chosen is the slow speed of 4 Hz. Pins 6 is connected to ground to enable this mode.

2.1.3 Design of the Microcontroller, MCU

The chosen microcontroller is the ATMEL AT89C4051-24PU 8 bit microcontroller with 4K Bytes flash with the following datasheet [7] specifications and features: Compatible with MCS-51™ products, 4K Bytes of reprogrammable flash memory with endurance of 1,000 Write/Erase cycles, 3.0V to 6V operating range, fully static operation from 0 Hz to 24 MHz, two-level program memory lock, 128 x 8-Bit internal RAM, 15 programmable I/O lines, two 16-Bit timer/counters, six Interrupt Sources, programmable serial UART channel, direct LED drive outputs, on-chip analog comparator, low power idle and power down modes, brown-out detection. The AT89C4051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 4K Bytes of Flash programmable and erasable read only memory (PEROM) having an Intel 8051 compatible architecture. It was chosen because it is a powerful microcomputer that provides a highly flexible and cost effective solution to many embedded control applications, has a small physical size suitable for the overall portable design of the , program code size, its two 16-bit timers, on-chip oscillator and clock circuitry, static logic for operation, as well as sufficient I/O ports enough for the intended embedded application.

The oscillator circuit is the heartbeat of the MCU and is crucial to its correct operation: proper functioning as well as timing calculations performed by the MCU. According to the AT89C4051, the components needed to generate an oscillator circuit are included on the MCU and by connecting a quartz crystal or ceramic resonator to the XTAL1 and XTAL2 pins plus two capacitors, an on-chip oscillator can be configured. The circuit for this is given in the datasheet.

Quartz Crystal, X1**,** is chosen for its stability. From the AT89C4051 datasheet, a machine cycle takes 12 oscillator or clock cycles. The frequency chosen for the design is 6 MHz. This is because using a factor of 12 (or a multiple of this) would simplify and make timing calculations accurate.
Microcontroller frequency: 6,000,000/12= 500,000 instructions/s (Hz).
Period of instruction: 1/f = 1/500,000 = 0.000002s =2 µs.

The value chosen for C_8 and C_9 is 30 pF as given in the AT89C4051 datasheet specifications.

A reset routine is run when the microcontroller starts up in order to place it into an appropriate state before it can begin executing the user program. A Logical one on the reset pin stops microcontroller's operating and erases the contents of most of its registers. By applying logical zero to this pin, the program starts execution from the beginning. In other words, a positive voltage pulse on this pin resets the microcontroller. In order to control the reset behaviour at start-up, an RC circuit was designed which would provide the required pulse on the MCU's reset pin. From the AT89C4051 datasheet, holding the RST pin high for two machine cycles while the oscillator is running resets the device and each machine cycle takes 12 oscillator or clock cycles. Using a 6 MHz clock, this would take 4μs, the minimum time required.

Pulse duration of 0.5s is chosen. The value chosen for C_7 [9] is 10μF.Since the capacitor is fully charged after 5 time constants (T) [3], the required time constant is (0.5/5) s = 0.1 s.
$R_9 = T/C = 0.1 / (10 \times 10^{-6}) = 10{,}000\ \Omega = 10\ K\Omega$.

2.1.4 Design of the Display Unit

The chosen display unit is PC 1607-A, a specialised LCD display only activated by microcontrollers. It is based on the popular Hitachi 44780 controller chip and displays all letters of the alphabet, Greek letters, punctuation marks, mathematic symbols, etc. It is a 16 x 1 (i.e. single line and 16 characters) display with each character made up of 5 x 8 dots. The PC 1607-A was chosen for its versatility – information can be displayed as desired and set that way – and because of its 4 bit mode for interfacing with the MCU that saves MCU I/O pins.
The datasheet [8] specifications are given below:
Power supply voltage, Vdd-Vss = 5.5 V (max)
LCD operation voltage, Vop, = 4.5 V
LCD current consumption (No B/L), Idd = 1.5 mA
Backlight current consumption, LED/edge = 40 mA, with VB/L = 4.2V
Backlight current consumption, LED/array = 120 mA, with VB/L = 4.2V
Input voltage, Vin = Vdd+0.3 V (max).

The specification of the chosen LCD is compatible with the circuit's power supply and MCU output voltage levels. The contrast adjustment of the LCD is made using potentiometer VR_3 whose value is chosen as 10KΩ.

$$i = V_{cc} / R \tag{5}$$

i = 5v / 10000 = 0.0005A = 0.5 mA. The power dissipated by the pot is $I^2R = 0.0005^2 \times 10000 = 0.0025$ W = 2.5 mW. The current level is safe.

The LCD backlight requires a +4.2 v at its 'A' pin. R_{10} is used to divide the supply voltage and provide the required current for the backlight operation.
From datasheet, LED/array = 120 mA (typical)
$V_{B/L} = 4.2$ V
Thus, $R_{10} = 4.2 / 120 \times 10^{-3} = 35\ \Omega$ for typical operation. However, a value of 47 Ω is selected as this is readily available.

2.2 Firmware (Software) Design

The firmware (software) for DigiThermo is designed to control the device as well as perform digital filtering of raw data sampled from the ADC. The program is designed to interface the AT89C4051 with the CA3162, a 10-bit ADC which digitizes the data from the LM35 temperature sensor, and the 16 x 1 line LCD display which displays time in sec units, and temperature in Celsius. FIR filters do filtering of raw data producing 0.1 °C reading.

2.2.1 Design of Control Program

The flow chart depicting the design of the control program is given in fig. 4 below. Tasks (functions performed periodically) are executed every 100 ms. In order to precisely execute these tasks every 100 ms, they are run within timer1 interrupt service routine (ISR) which occurs whenever timer1 overflows after counting 100 ms. This design employs the simple Embedded Operating System concept (sEOS) [11]. For controlling the 16 x 1 character LCD display, the LCD initialization process makes use of the Hitachi 44780 4 bit mode initialization steps [11].

2.2.2 Design of Digital Filter

The digital output of an ADC is equivalent to the continuous input plus a quantization error [3] due to its finite resolution. The quantization error is an unavoidable imperfection in all ADCs [3]. The quantization error appears very much like *random noise* and in most cases, quantization results in nothing more than the addition of a specific amount of random noise to the signal [3].

The digital filter chosen to remove this random noise is the moving average filter. This is because apart from its simplicity, its ease of understanding and use, it is optimal for reducing random noise while retaining a sharp step response and a suitable filter for time domain encoded signals [3]. Additionally, it is easy to implement without requiring floating point mathematics and our program is designed around integer mathematics. This enables it run faster [3]. The moving average filter is a Finite Impulse Response (FIR) filter. Our design makes use of a 10 point filter kernel to implement a 10-point moving average filter by convolution. The amount of noise reduction of this filter is equal to the square-root of the number of points in the average. In equation form the filter is written as:

$$y[i] = \frac{1}{M}\sum_{j=0}^{M-1} x[i+j] \tag{6}$$

where M = 10 is the number of points used in the moving average, x[ ] is the input signal, y[ ] is the output signal.

Filter kernel:
…, 0, 0, 1/10, 1/10, 1/10, 1/10, 1/10, 1/10, 1/10, 1/10, 1/10, 1/10, 0, 0, … .
The moving average filter is a convolution of the input signal with a rectangular pulse having an area of one.

Noise reduction factor:
$$\sqrt{10} = 3.16 \qquad (7)$$
Thus it reduces the noise by a factor of 3.16

The moving average filter operates by averaging 10 points from the input signal to produce each point in the output signal. The flowcharts illustrating how this is implemented using two tasks are given in fig. 5 below. The program captures 10 samples (during the 1st one second during which time it displays an out of range value), then performs one sided averaging on them, before producing the first point in the output signal. The delay enables the program to calculate the first sample in the output signal only when the filter kernel is fully immersed in the input signal [3]. Successive output points are produced every 100 ms using moving data in the 10 point FIFO buffers.

2.2.3 Firmware Description

The program, mythermo.c that controls the DigiThermo was written in 'C' language. The choice for C was made because of its efficiency, support for 'high-level' features (such as support of functions and modules) as well as 'low-level' features (such as good access to hardware via pointers) and most especially because of the availability of well proven C compilers for every embedded processor including the AT89C4051. Tasks are separated into six tasks: time(), putxin(), puttemp(), start(), and reset() which are performed / executed every 100 ms. For precise timing, these tasks are run within the timer1 ISR, update_task() which occurs when timer1 overflows every 100 ms.
Time() sets FLAG bit0, bit1 and bit2 every 100 ms, after first 10-samples, and every 1s (as long as the second counter is enabled) respectively.
Putxin() shifts a converted digital word to LSW of 10-word registers performing 10-point data moving. As this is done every 100 ms, a sampling frequency of 10 Hz is achieved. A subroutine, readtemp(), reads the raw data into the FIFO buffer/register.
Puttime() computes average value of 10-sample, producing an output sample signal from a 10-point moving average filter, converts this into the corresponding temperature in °C and sends this value to the LCD buffer.

Similarly, *puttime()* writes variable count (i.e. second count) to LCD buffer. *Start()* checks the state of the SW_3 pin every 500 ms. Whenever the switch is momentarily closed, it toggles the state of the second timer – either starts it or stops it. The 500 ms delay prevents switch bounce which was responsible for the non responsiveness of the timer start/stop switch due to high speed toggling of the timer at 100 ms intervals.

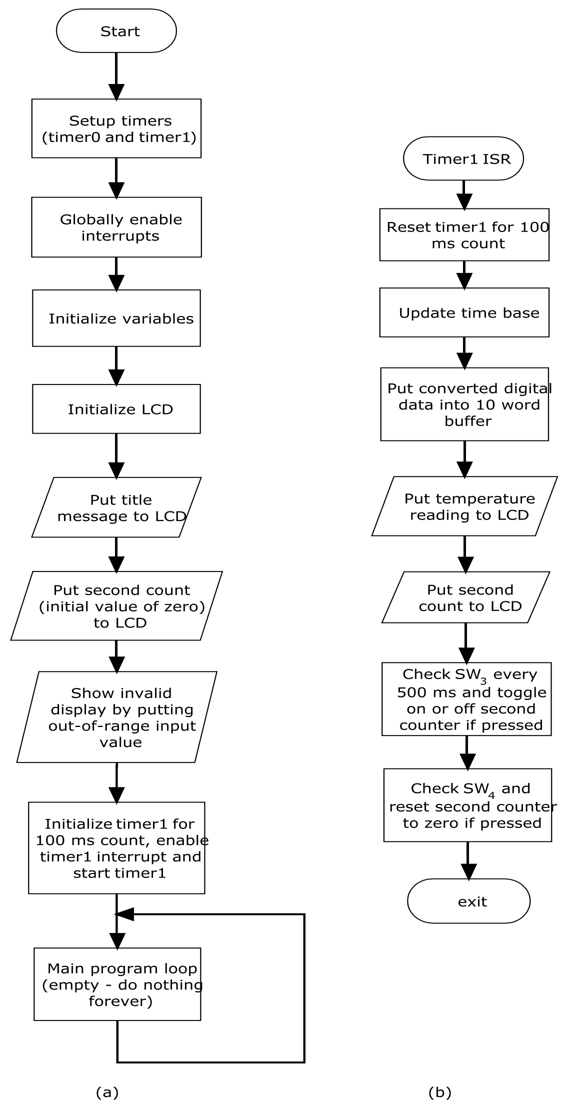

Figure 2: Flowcharts for DigiThermo Firmware (a) Main Program (b) Timer1 ISR executes tasks

Reset() resets the second counter to zero whenever SW_4 is momentarily closed.

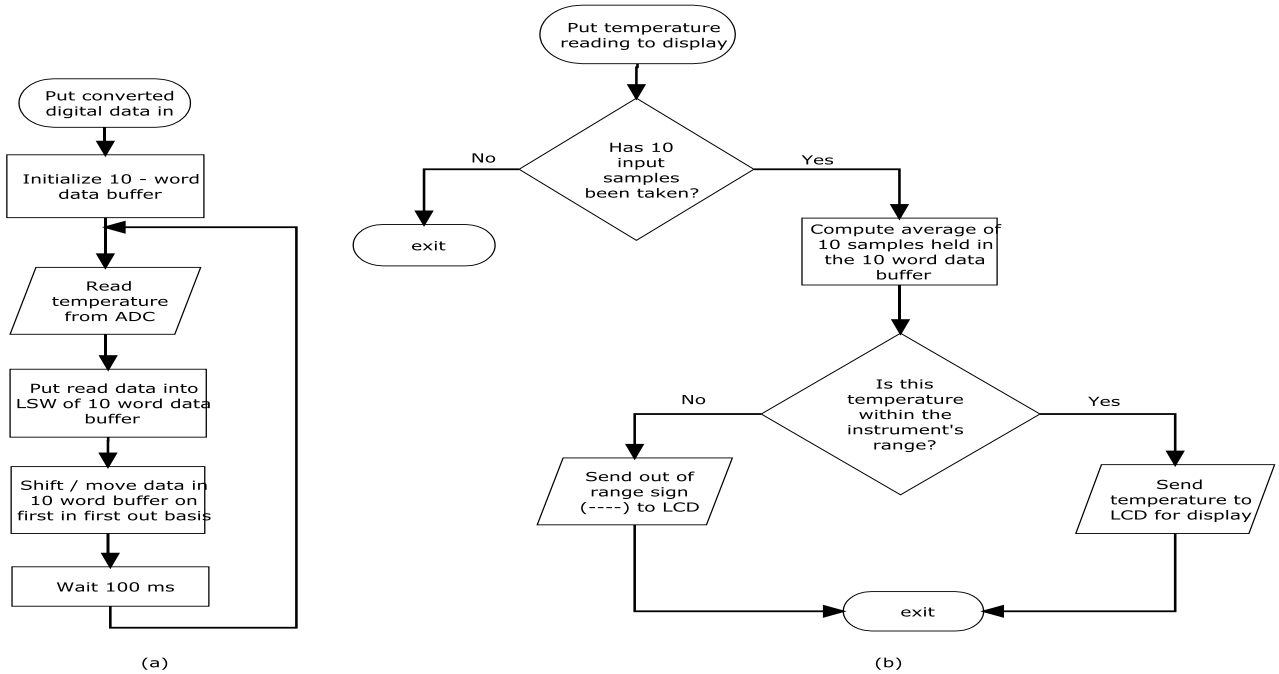

Figure 3: Flowcharts for digital filter implementation (a) convolution (b) sampling.

The driver routines are as follows:

average() – computes average of 10-samples in FIFO buffer

readtemp() – reads the BCD value from the CA3162 ADC

i_LCD() – initializes the LCD using the 4 bit initialization algorithm for the Hitachi 44780 LCD controller

LCDWI() – writes LCD instruction into the LCD in 4 bit mode

LCDWD() – writes ASCII code to LCD buffer

pulseE() – generates LCD enable pulse

pause() – provides a very short delay

delay(n) – delay n milliseconds

delay_1ms() – creates a 1 ms delay with 6 MHz XTAL using timer0

init_timer() –initializes timer1 for a 100 ms tick interval for the sEOS with 6 MHz XTAL

puttitle() – puts the title message to LCD buffer

2.2.4 Microcontroller Programming

The firmware (software) was written using the high level programming language, C. The source code was saved as the file *mythermo.c*. A special C compiler targeted at the 8051 compatible microcontrollers, the SDCC was used to compile the source code. The output machine code was a hex file, *mythermo.ihx* with *.ihx extension. The tool packihx was used to convert this hex file with *.ihx to *.hex producing the packed Intel hex file *mythermo.hex.*

The compiler runs under a dos environment. The following commands were used in the compiling process: "C:\sdcc\bin\sdcc mythermo.c" compiled the source code into the object code: mythermo.ihx while "C:\sdcc\bin\packihx mythermo.ihx" packed the intel hex file into mythermo file with the .hex extension suitable for burning into the microcontroller using the BlowIT programmer circuit. Variables and stack use the area of 128-byte on-chip RAM while the special function registers (SFR) use the area of 128-byte on-chip RAM reserved for them.

Burning of the object code into the microcontroller, that is, the programming of the microcontroller was done using an AT89C4051 microcontroller programmer circuit called *BlowIT 2051*, a burner designed for AT89C2051 Flash Microcontroller by Silicon Studio Ltd and a BlowIT executable modified by Dincer Aydin that supports Intel hex files and is able to able to program the AT89C1051, 2051 and 4051. The BlowIT.exe program runs under real Dos mode.

The object code and the programmer were placed in a removable drive containing a copy of MS-DOS. The microcontroller was placed into a 20-pin DIP socket on the BlowIT burner. The burner was connected to a PC's parallel port and was powered. The PC was powered and booted into DOS. The BlowIT program was run using the following command:

"A:\Blowit mythermo.hex [1]"

The target MCU, 4051, was selected and the chip was programmed. The BlowIT power was turned off and disconnected from the PC, and the PC was shutdown. The MCU was removed and was ready to be installed into the circuit.

2.3 Principle of Operation of the DigiThermo

When power is turned on via SW_1, the 9V battery provides power to the circuit. This power is regulated by IC_1, the 7805 5V regulator to provide +5V on which the circuit operates. The Microcontroller, the ADC, the temperature sensor and the LCD are all driven by this source.
The reset circuit of the microcontroller (MCU) resets the microcontroller during the first 0.5 s after power has been turned on, as the reset pin is held high during this period until the 10µF capacitor becomes fully charged.
After being reset, the MCU starts executing the control program, mythermo.hex, which initialises the LCD, then controls the functionality of the circuit. The 6 MHz crystal connected to the MCU makes these instructions to be executed at a rate of 0.5 MHz (500,000 instructions/s).
IC_2, the LM35DZ temperature sensor, measures temperature producing an output signal of 10mV/°C. This output is filtered by the first order low pass filter formed by R_1 and C_1 and is sent to the ADC chip, IC_3, the Intersil CA3162, 3-digit DVM employing dual slope integration, through its differential input pins 11 for HI and 10 for LI signal. The ADC digitizes this signal at a rate of 4 Hz. Since the ADC converter is capable of providing 0 – 1000mV reading with 1mV resolution, the converter can resolve 0.1 °C (though not absolute accuracy). 3-digit digital output is sent to the MCU's port1 (P1.0-P1.3) via the ADC's multiplex four bit BCD outputs (pins 2, 1, 15, 16) starting from the MSD, LSD, and the NSD respectively. The MSD, NSD, and LSD digit select pins (pins 4, 3, and 5), tied to MCU's pins 11, 7, and 6 respectively, signify when these digits are ready to be read. The control program reads the appropriate digit when any of the pins 11, 7 or 6 is low every 100 ms. The MCU thus samples the ADC at a rate of 10 Hz. It performs digital filtering on the sampled data using FIR digital filters to produce 0.1 °C reading. The temperature reading is sent to the LCD and is updated every 100 ms.
The timer is implemented in the control program making use of one of the MCU timer circuitry, timer1. Timer1 overflows every 100 ms providing a 100 ms time-base producing 1 s for time counting. Each second, the control program sends the second count to the LCD. SW_3 is used to start / stop the timer while SW_4 resets the second counter to zero when pressed respectively. The states of these switches are monitored by the control program, mythermo.hex.
The 16 x 1 character LCD, connected in 4-bit interfacing to pins P1.4 – P1.7 with control signal RS and E to P3.4 and P3.5 respectively, displays time in 1 second unit and temperature in 0.1 °C resolution and can show second count of up to 5 digits while temperature is shown in 3 digits. On power up, the LCD is initialized by the MCU, after which it displays the title message before showing the variables being measured. SW_2 is used to turn on/off the LCD background light. The device is turned off via SW1.

3. Construction

The circuit was first assembled on prototype board to enable testing of the circuit design and easy modification when necessary. All components were connected according to the circuit diagram. C_3 was placed as close as possible to the CA3162E's ground and power pins as suggested in the ADC's datasheet. Other special procedures for handling some other components were adhered to during assembly.

3.1 Calibration of the ADC

Calibration of the ADC was done as follows: ZERO adjustment was done by shorting pin HI and LO to GND and then the 50k POT at pin 8 and 9 was adjusted until the temperature reading was 0.0 °C. A reference voltage source of 391mV was connected to the input of the ADC and the 10k POT was adjusted until the display showed 39.1 °C.

3.2 Preliminary Testing

Preliminary test was done on the fully assembled circuit. This was aimed at ascertain whether it was working and if it performed as expected. The timer and temperature measuring units were tested for accuracy as well as functionality. The timer did not measure time accurately when compared with the timer on a Nokia phone, and the temperature reading was unstable.
This led to a revision / modification of both the firmware and the hardware to tackle the problems discovered. The revised firmware was burnt to the MCU and the circuit was modified to handle the multiplexed output of the ADC much more effectively.
The tests were repeated until a reliable, accurate output was obtained from the .

3.3 Permanent Assembly

The circuit (figure 5) was then transferred to a matrix board where all the components were assembled, connected and soldered into a permanent form. This formed the circuit's mainboard. The layout of components on this board had been carefully planned based on component type, interconnection amongst them as well as the size of the board.

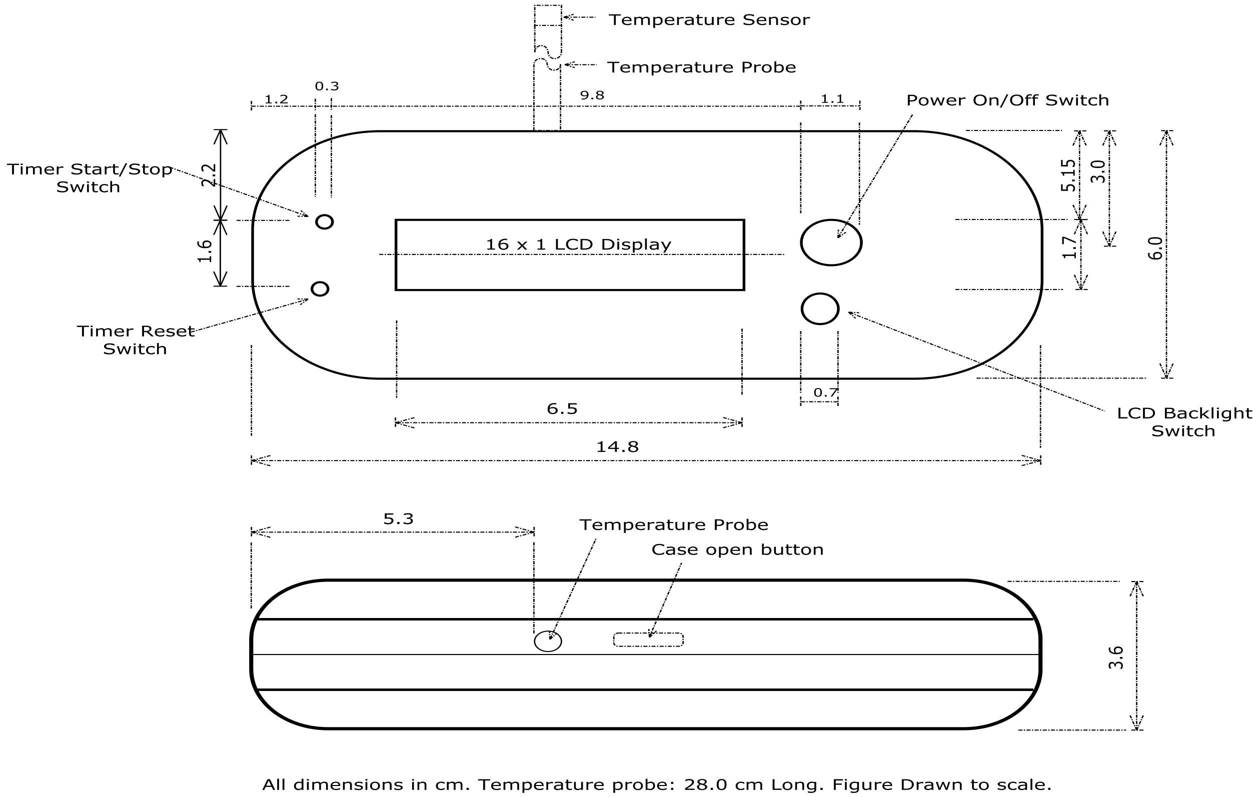

Figure 4: The Casing for DigiThermo

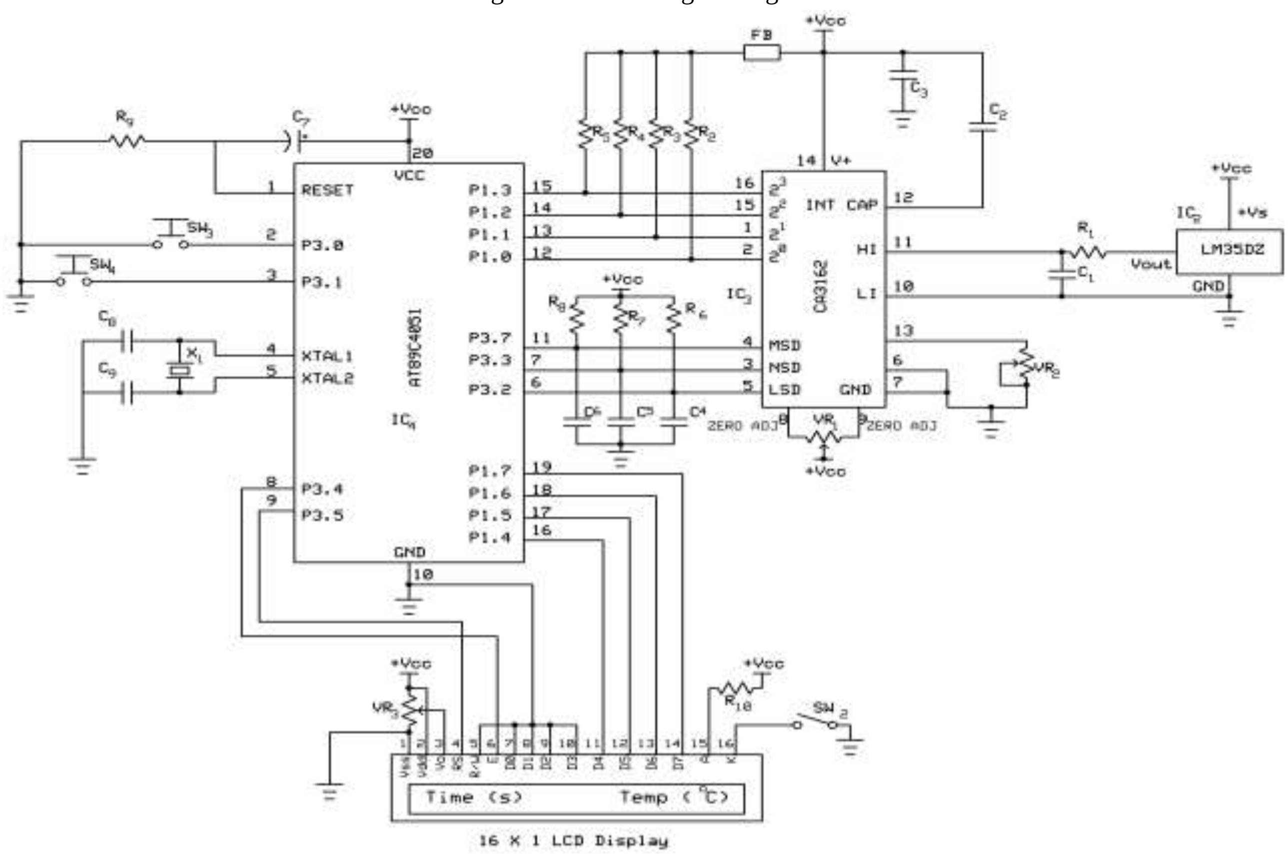

Figure 5: The complete circuit diagram of DigiThermo.

The size of the matrix board was determined by the size of the selected case, which was chosen to make the device portable. The case is a rounded cuboid with dimensions of 14.8 cm x 6.0 cm x 3.6 cm taken across centres (figure 4,6 and 7). A rectangular opening was made on the top of the case to provide a window for the LCD. A plastic transparent sheet, cut to size, was glued underneath the opening to protect the LCD that was later attached just below it to the case using screws. Circular holes were made for the switches on the case. The circuit's four switches were mounted on two small matrix boards with connecting wires to the mainboard. Four plastic round covers were placed over the switches. The switch boards were attached to the upper surface of case on both sides of the LCD using glue. The circuit's mainboard was then fixed into the bottom half of the case. A probe made of rubber tubing and well sealed to keep off moisture and water, was attached to the top of the casing with the temperature sensor at the other end. The probe measures 28.0 cm. This provides flexibility and enables temperature to be measured easily at some distance away from the device itself. Measurement of temperature of fluids is also made easy by this. The LM35's 3 pins were insulated with rubber sheets before being placed within the rubber tubing. The colours of the connecting cables used on the sensor are as follows: orange to $+V_s$ pin, orange with white to V_{out} pin, and green to the GND pin.
A 9V battery was attached to the circuit through a 9V battery connector on the mainboard.
The case was then closed and power was turned on. The ADC was recalibrated. The device worked as expected.
Labels for the switches, as well as the device's name tag, etc were glued at appropriate places on the case. Figure 8 shows the device in its final casing.

4. Results and Discussion

After the final assembly, the circuit was retested and the final results were as shown on Table 1.

The final result obtained shows that the temperature sensor's output is linear. Both the temperature and time measurement aspects of the device are performing satisfactorily.

Table 1: Test and Result

TEST	***METHOD***	***RESULT***
Timer operation	Timer was compared with the timer on a Nokia mobile phone. Both timers were started simultaneously.	There was one to one correspondence with each second count
Thermometer operation	Temperature probe was left to measure the ambient temperature. Later, it was placed close to a heat source, a laptop air vent blowing out warm air. For a second test, the tip of 's sensor was placed on the sides of a hot soldering iron for some time, and then transferred to a cup of cold water containing some ice.	The temperature read 28.0 °C in the first case, and increased to 32.0 °C when it was placed close to the heat source for some time. The temperature reading rose linearly from room temperature to 43.0°C when placed on the soldering iron during the second test and then fell gradually in a linear manner to 5.0°C when immersed in the cold water.

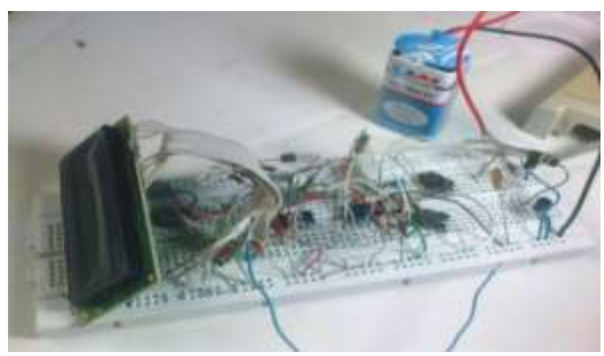

Fig. 6: Preliminary assembly on breadboard

Fig. 7: Case opened showing assembled circuit on matrix board.

Fig 8: Front view of device in final casing.

5. Conclusions

The main aim of this work was to design and construct a microcontroller based digital thermometer with timer. This has been achieved. The device has been tested and is working. This project illustrates the use of embedded systems particularly in instrumentation design and generally in the design of electronic devices. Embedded system design should be encouraged to simplify and provide flexibility for electronic circuits / electronic designs. Those seeking guidance on embedded system design that employ ADC interfacing, specialised LCD interfacing, digital filtering, etc, should avail themselves with this work.

Acknowledgments

Author(2) wishes to thank Prof. Wichit Sirichote, of the Department of Applied Physics, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, for his work on Microcontrollers and embedded systems and his prompt response.

References

[1] “Digital Thermometers”, GlobalSpec, the Engineering Search Engine, available online at http://material-handling.globalspec.com/, accessed on 16/12/2011
[2] LM35 Precision Centigrade Temperature Sensors National Semiconductor Corporation, November 2000, pp. 1- 4.
[3] S. W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing, CA: California Technical, 2003, pp. 36, 54, 191-350.
[4] B. Grob, Basic Electronics, Ohio: Glencoe/McGraw Hill, 8th Ed., 2002, pp. 649.
[5] CA3162, Intersil Corporation, Florida, 2002.
[6] AN9214, Using Intersil High Speed A/D Converters, Intersil Corporation, Florida, September,1993, pp. 2
[7] Fair-Rite Product's Catalog Part Datasheet 274300111, Fair Rite Products Corp., 08/2008
[8] 8-bit Microcontroller with 4K Bytes Flash: AT89C4051, Atmel Corporation, California, 2008.
[9] PC 1601-A: Outline Dimension and Block Diagram (Powertip Corporation)
[10] M. J. Pont, Programming Embedded Systems 1 using C, a Seminar Slide, University of Leicester, UK, 2006, pp. 34-148.
[11] HD44780U (LCD-II): Dot Matrix Liquid Crystal Display Controller/Driver, Tokyo, Japan: Hitachi, Ltd., 1999, pp. 17- 46.

Biography

Abayomi-Alli, A. obtained his B.Tech Degree in Computer Engineering from Ladoke Akintola University of Technology (LAUTECH), Ogbomoso in 2005, MSc Computer Science from the University of Ibadan, Nigeria in the 2009. He started his career at Igbinedion University Okada, Nigeria as a Graduate Assistant in 2007 before moving to the Federal University of Agriculture Abeokuta (FUNAAB), Nigeria in 2011. His current research interests include microprocessor systems and applications, biometrics, image quality assessment and machine learning.

Etuk, **E**. graduated with a first class Degree in Computer Engineering from the Department of Electrical and Computer Engineering at Igbinedion University Okada in 2009 and is presently a post graduate student on commonwealth scholarship at Northumbria University.

Ezomo, P. I. has a B.Eng and M.Eng Degree in Electrical Electronics. He is presently a lecturer in the Department of Electrical and Computer Engineering at Igbinedion University Okada. He has extensive experience in microprocessor systems and automation while he worked in the oil and gas industry for over twenty years.

Izilein, F. A. has a B.Eng and M.Eng Degree in Electrical Electronics. He presently lectures at the Department of Electrical and Computer Engineering in Igbinedion University Okada.

Akingboye, A.Y. graduated from the Ladoke Akintola University of Technology, Ogbomoso with a Master of Technology (M.Tech) and Bachelor of Technology (B.Tech) degrees in Computer Science in 2012 and 2005 respectively. His research interests include Microprocessor Systems and Human Computer Interaction. He is presently with the Department of Electrical and Computer Engineering at Igbinedion University Okada.

Erameh, **K. B**. graduated with a first class Degree in Computer Engineering from the Department of Electrical and Computer Engineering at Igbinedion University Okada in 2010 and is presently a graduate assistant on post graduate studies at the University of Benin.