EE318
Electronic Design Lab Report
Electrical Engineering Department
IIT Bombay
May 2005

MEMO ALARM CLOCK WITH TEMPERATURE DISPLAY

GROUP B-09

VIKRANT KULKARNI	(02007021)	vikrant@ee.iitb.ac.in
SIDDHARTH CHAUDHARI	(02007022)	sidkc@ee.iitb.ac.in
NISHANT JAIN	(02007024)	nishantj@ee.iitb.ac.in

ABSTRACT

This project is aimed at designing a digital alarm clock with ambient temperature display. The clock accepts the alarm time and a message to be displayed when the alarm rings. The time is displayed in 24 Hr format. The clock comes with a 16 character X 2 line LCD display and a 16 key, 56 character keypad and runs on a 5V DC supply.

PROBLEM STATEMENT AND PRODUCT DESIGN

The problem statement entails the design of a digital alarm clock with ambient temperature display. The clock should also be able to accept the alarm time and a message to be displayed when the alarm rings. This project is based on an Atmel 89C52 Microcontroller. The peripheral chips include an ADC0809 Analog to Digital Converter, INA118 Instrumentation Amplifier, IC555 Timer and a DS1302 Trickle Charge Timekeeping Chip. Other peripherals are a 4X4 Matrix Keypad, JHD162A Liquid Crystal Display (LCD), a PT100 Temperature Sensor and a push button for generating interrupts. The block diagram is given in Figure 1.

This project requires the interfacing and integration of four major components: the LCD (display unit), Matrix Keypad (input unit), Real Time Clock (timekeeping unit) and the temperature sensor. The approach followed was to individually set up each component (hardware and software), test it and finally integrate all the four. As seen from the block diagram all the hardware is directly interfaced to the AT89C52 (except the IC555, INA118 and PT100). The software code runs in one of two major loops i.e. one loop continuously takes input from the RTC and Temperature Unit, displaying it on the LCD. It runs in the other loop on receiving an interrupt where it can take input from the keypad. In the first loop it also checks whether the alarm condition is met

and if so it flashes the time and associated message. The current time, alarm time and the message are stored in memory as shown in Figure 2.

Figure 1: Block Diagram

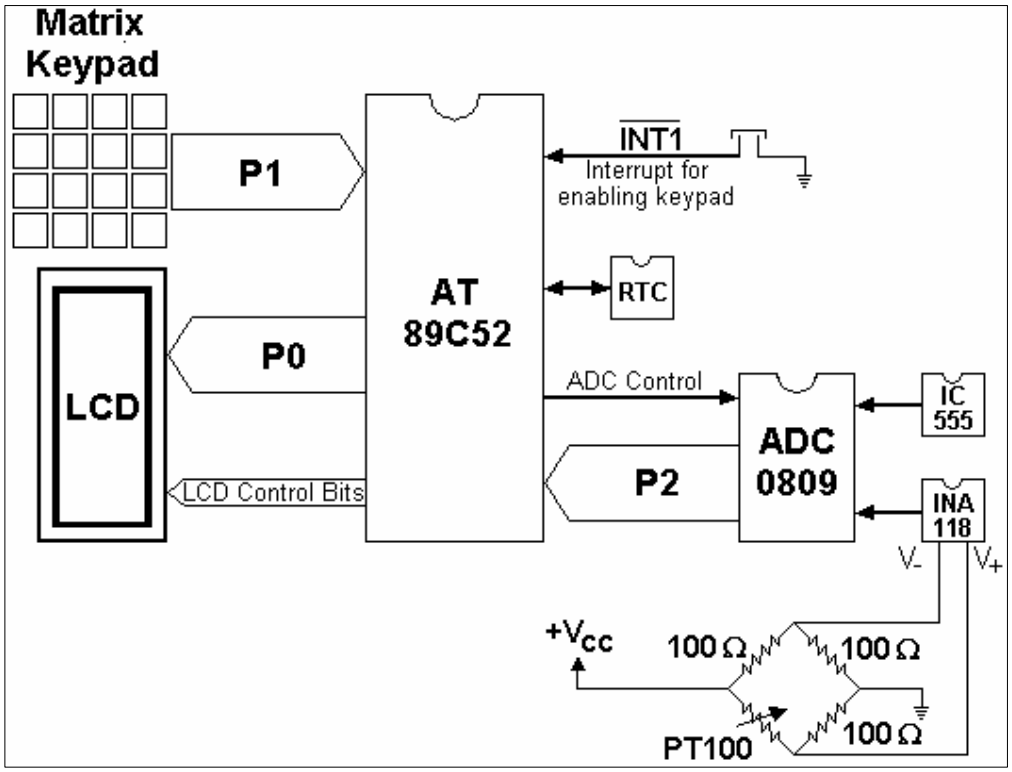
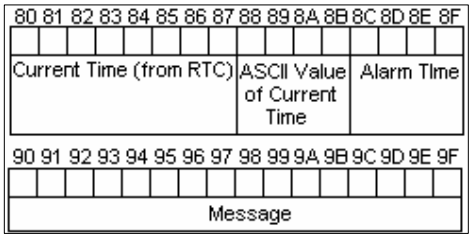


Figure 2: Memory Utilization

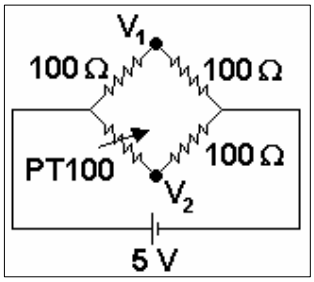


CIRCUIT DESIGN

The circuit design problem mainly consisted of interfacing all the chips to the microcontroller. The 4X4 keypad requires 8 data lines and it is interfaced on Port 1. The LCD module requires 11 lines (8 data lines and 3 control lines) and it is interfaced on Port 0 (data lines) and pins 3.0, 3.1 and 3.2 (control lines). The temperature measurement module requires 9 lines (8 data lines and 1 control line) and it is interfaced on Port 2 (data lines) and pin 3.4 (control line). The timekeeping unit (RTC) requires 3 lines on pins 3.5, 3.6 and 3.7. Pin 3.3 is connected to an external button which when pressed interrupts the main program and enables the keypad.

The RTC chip gives the advantage of timekeeping without the processor getting involved in the process. Any software clock (even one using AT89C52 timers) suffers from minute delays associated with unequal processing time of microprocessor instructions. This leads to gross approximations in the long run. An RTC functions independently of the microcontroller and thus provides an accurate value of time.

Figure 3: PT100 Circuit



The temperature unit consists of a PT100 temperature sensor followed by an INA118 amplifier and an 8 bit ADC. The PT100 is a resistance with resistance varying as

$$R(T) = 100\Omega + 0.385T, \text{ where T is in } ^oC$$

In this design it is used in a Wheatstone bridge circuit (Figure 3) and the voltages at the two ends are monitored. From circuit equations we see that

$$V_1 = 2.5V \text{ and } V_2 = 5 \cdot R(T)/[R(T) + 100]$$

$$\Delta V = V_2 - V_1 = 5[R(T)/(R(T) + 100) - 0.5]$$

This ΔV is amplified by an INA118 instrumentation amplifier with a gain of 12.5V/V.

$$\Delta V_{amp} = 62.5[R(T)/(R(T) + 100) - 0.5] = 31.25[(R(T)-100)/(R(T)+100)]$$

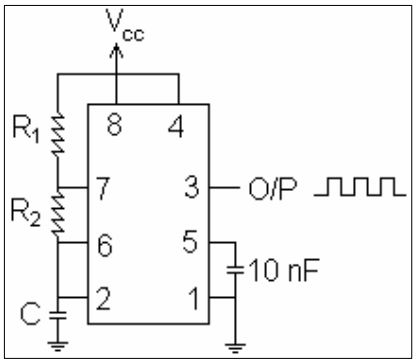
Since ambient temperature is 25 ^oC,

$$R(T) + 100 \approx 209.625\Omega \text{ and } R(T) - 100 = 0.385T$$

From the ADC for an input of 5V we get an output of 11111111 i.e. 255_{10}. Thus for an input of ΔV_{amp} we get an output value of

$$Output_{10} = (255/5) \cdot \Delta V_{amp} = 2.927T \approx 3T$$

Figure 4: IC555 Circuit



The ADC0809 chip also requires a square wave input of 500kHz that is provided by an IC555 timer (Figure 4). The frequency of oscillation is given by

$$f = 1.44/[(R_1 + 2R_2)C]$$

Choosing R1 = R2 = 436Ω and C = 2.2nF gives the required square wave.

The Matrix Keypad does not require any interfacing hardware apart from an external pull up provided by a 4.7kΩ CIP as shown in Figure 5.

The LCD module also has simple connections as shown in Figure 6. We require the 22kΩ potentiometer to change the screen contrast of the LCD.

Figure 5: Matrix Keypad

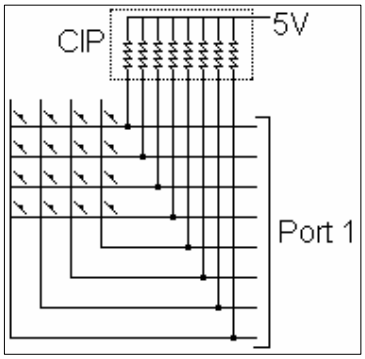
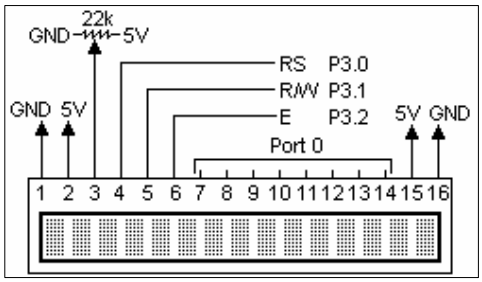


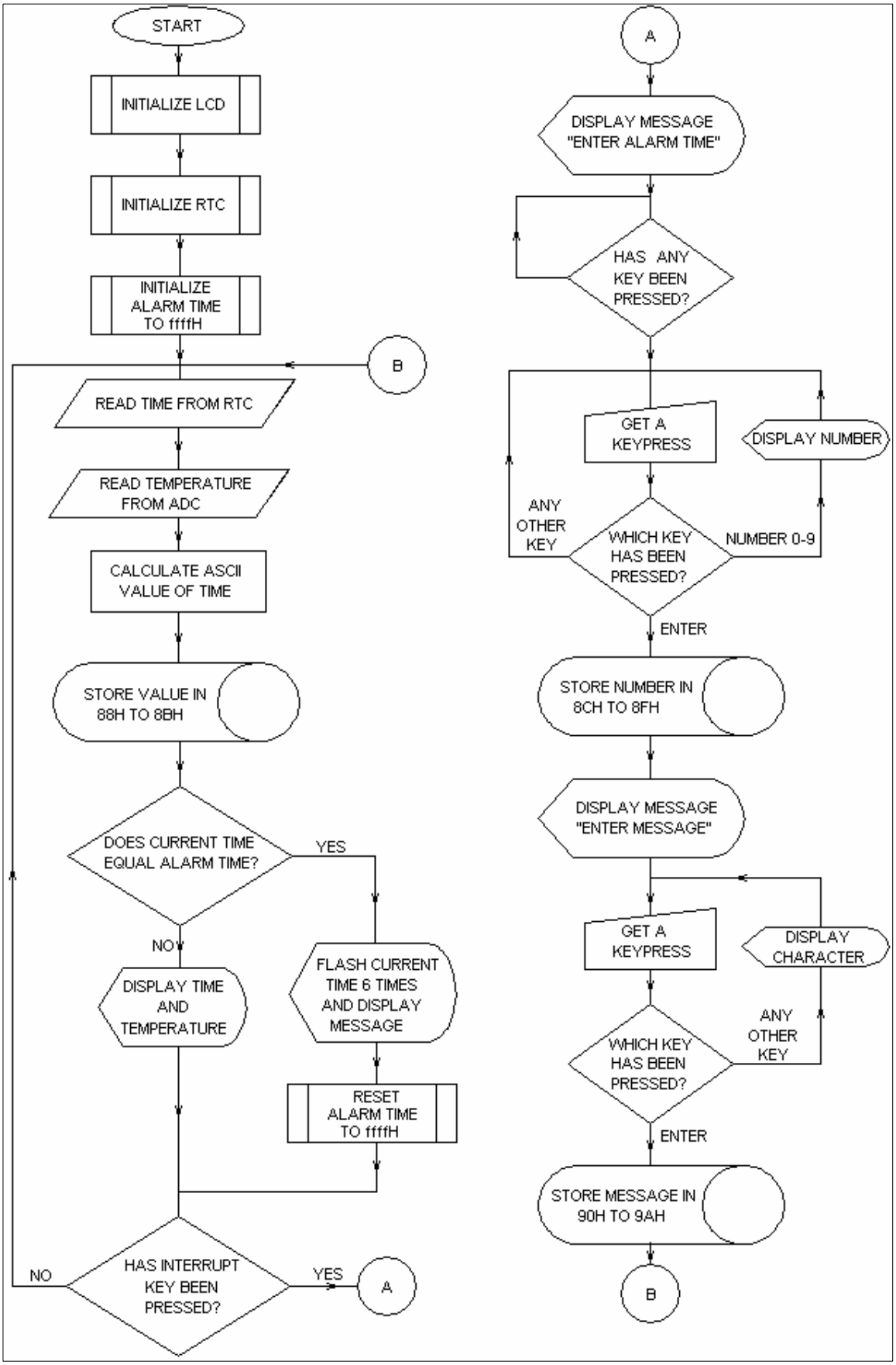
Figure 6: LCD Module



FLOWCHART

The flowchart is as given in Figure 7.

Figure 7: Flowchart of algorithm



TEST RESULTS

On power on, the time is displayed as 13:59:30 along with temperature up to one decimal point. On pressing the interrupt key the display is cleared and we get a message, “ENTER ALARM TIME”. On any key press the display clears and we get to enter alarm time. On pressing the enter key we get a message, “ENTER MESSAGE”. We then enter the message and again press the enter key. This brings it back to the time display mode.

SUGGESTIONS FOR IMPROVEMENT

The extra button provided for interrupting the microcontroller can be done away with by having a latch circuitry connected to the keypad. This will enable the keypad by pressing any keypad key. By utilizing more memory we can incorporate more alarm settings. The user may also be allowed to set the current time. A buzzer may be provided for giving a sound output to indicate alarm. This will ensure that even in keypad mode the alarm is sounded. Since we do not require very accurate analog to digital conversion, the ADC0809 chip may be removed and instead the inbuilt comparator and timer of the AT89C52 may be used for A to D conversion. This will bring down the cost of manufacturing. The PCB design had some flaws which also need to be rectified. The device also needs to be packaged.

CONCLUSIONS

This device has good demand in the small electronic goods sector. If the above mentioned suggestions are incorporated then it will be a low cost alternative to other alarm clocks with the added feature of message and temperature display.

REFERENCES

1. Ilett, Julyan; *How To Use Intelligent LCDs* in Everyday Practical Electronics, February 1997.
2. *The 8052 Tutorial and Reference Version 1.00, 25 June 2004* from www.8052.com. © Craig Steiner and Vault Information Services LLC
3. *Datasheet of ADC0808/ADC0809 8-Bit µP Compatible A/D Converters with 8-Channel Multiplexer* from www.national.com. © National Semiconductor Corporation, 1999
4. *Datasheet of ATMEL 89C52, 8 bit Microcontroller with 8K bytes Flash* from www.atmel.com. © ATMEL Corporation, 1999
5. *Datasheet of DS1302 Trickle Charge Timekeeping Chip* from www.maxim-ic.com. © Maxim Integrated Products, 2004
6. *Oriole LCD Module, User’s Manual*, © Oriole Electronics Pvt. Ltd.

APPENDIX
USER'S MANUAL

The device requires a 5V DC supply that can be availed from a DC battery. In the normal mode it displays time in HH:MM:SS format and temperature up to a tenth of a degree. To set alarm time press the interrupt button once. The screen displays a message "ENTER ALARM TIME". Press any key to clear screen. The screen now displays, "00:00". Press any of the numeric keys to enter the time. For example after inputting 14:25 we see "14:25" on the screen. The cursor cycles back to the first position after 4th number is entered. Press enter key once to accept the time. The screen now displays "ENTER MESSAGE". Press the required keys to input message. Use the backspace key to perform any modifications. When done press the enter key to accept message and return to normal mode. The keypad layout is given in Figure 8.

Figure 8: Keypad characters