Ahmed Adil (18L-1289)

Mahad Naveed Qureshi (18L-1039)

# Hardware’s description:

|  |  |  |
| --- | --- | --- |
| COMPONENTS | DESCRIPTION | MARKET PRICE |
| AT89C52 Microcontroller | 8KBs of (PEROM), 256 bytes of RAM, 1000 write/erase cycles | Rs: 180/ piece |
| Piezo active buzzer | 3V-12V, 12mm round speaker, audible 2KHz range, 80dB of sound output | Rs: 20/ piece |
| Arduino 3x4 keypad | 12 keys membrane switch, 5 pin 2.54mm pitch connector | Rs: 130/ piece |
| Blue 1602 Arduino display LCD | HD44780 parallel interface chipset, LED backlight, 5x7 dot matrix character + cursor | Rs: 240/ piece |
| Veroboard | 15x7cm, 1.22mm, single sided copper PCB | Rs: 40/piece |
|  |  |  |
|  |  |  |

# Description of Algorithm:

* Main:

In the main routine first of all, scanning (**scanning\_input**) is performed 6 times according to the given input format \*/# HHMM \*/#, and stored in registers.

Checks are implemented after scanning

When the first input is ‘ \* ’ then call **time\_set** function for setting time, a further check is implemented (**check\_for\_AM\_PM**) after the delay to check the AM/PM for the last given input \*/# respectively.

When the first input is ‘ # ’ then call **alarm\_set** function for setting alarm, again this check is implemented (**check\_for\_AM\_PM**) after the delay to check the AM/PM for the last given input \*/# respectively.

When input is not equal to format (\*/# HHMM \*/#)

Output: “wrong input” (string)

Scanning 6 times again…. Checks: that the first and last input is \*/#

* Designing the Clock

For designing 1 second the interrupt of timer 1 is used, for which the processor jumps to 1BH location where the subroutine is called similarly, 1 minute is designed using CJNE and similar for 1 hour.

The **disp** function is called after every 60 seconds which displays the changed time on the LCD screen. A delay of 50 milliseconds is made for which it checks whether the timer is equal to the given alarm time (50ms x 20ms = 1sec).

Starting point is 4B78H for timer 0 and 1

* A function of setting time

Timer was cleared and then the input data was separated, minutes and hours were stored in r2 and r3.

In **time\_set** the digits which were separated are stored in accumulators and then added in respective registers (r2 and r3) for hours and minutes separately so that rather than two separate digits they can be represented as 1 digit.

**Input\_dsep** functionswaps the 10s value and adds it to the 1s value thus creating as a single two-digit number for representing minute and hours.

* A function of setting alarm

Interrupt of timer 0 is used in it which check after every 50ms that is it alarm time or not

Timer was cleared and then the input data was separated, minutes and hours were stored in r2 and r3.

In **alarm\_set** again digits are separated for creating as a single two-digit number for representing minute and hours.

Check

When time = alarm

Buzzer is on (t = 5 secs), p1.4 is grounded and the routine jumps back to main.

* Display

It just moves the input values from different functions that were being performed on it to given memory locations then send it to the LCD after delay.

**Commwrt**: clears pin to 0

**Datawrt**: sets pin to 1

# Other approaches/ limitations:

Interrupt of timer 1 could be used in setting alarm, but generally timer 0 has higher priority than timer 1 and alarm had to be an immediate interrupt in the normal clock and to signal the buzzer, otherwise the alarm wouldn’t have performed it function.

Scanning could also be done using interrupts in the Main but then a delay of 5 sec would have been encountered, when the buzzer command has been executed because while the buzzer was on, both the operations (clock and buzzer ringing) couldn’t run simultaneously.

# Circuit:

