Design and Development of Microcontroller Based Digital Bangla Clock

Victor Kathan Sarker, M. Ataur Rahman, and M. A. Matin

***Abstract*—In this paper, a microcontroller-based digital Bangla clock has been constructed with ATmega32 and its software is written with C language. The designed clock is reliable, portable and fully functional with alarm system and its Bangla digits are displayed accurately with a 10-segment display. The clock is also functional in dual language mode both English and Bangla.**

***IndexTerms*—Bangla, digital, clock, display, microcontroller, 10 segments.**

I. Introduction

Now a day, time is mostly measured using electronic clocks instead of mechanical ones. Though all clocks measure time, but different clocks can have different status or importance. In 2008, *Pan Thu Tun* has designed a digital Clock using microcontroller with seven-segment display [1]. However, there are some limitations in his design. It could not display year, month or day. Moreover, his system became relatively expensive due to using external decoder. Few researches have been done for displaying Bangla numerals. *M. S. Arefin et al.* [2] designed a 24 segment display for Bangla characters and Numerals. But this design is redundant when it comes to display only numerals. Similarly Sabbir Ahmed et al [3] designed a 10-segment display for Bangla digits but their segment were not uniform and in addition to it their design has some controversy regarding portraying digits '1', '2', '3' and '7' accurately. In our design, we corrected those problems. *M.O Rahman and et al.* [4] also designed an 11-segment display for Bangla, Arabic and English numerals. Their design almost uniform and has no segment intersections.

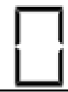
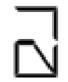
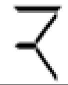
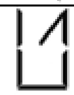
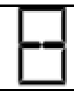
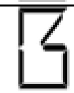
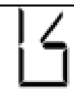
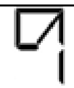
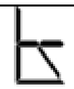
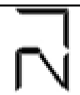
In this paper, a microcontroller based Digital Bangla Clock has been designed and constructed. Here, we used four 10-segment displays and an AVR microcontroller ATmega32 from ATMEL and other necessary elements as per our design requirement. This clock also has 24 hr or AM/PM notification, user selectable audible alarm system with snooze function.

II. System Design

In our proposed design, we have used 10-segment display for portraying the Bangla digits as well as the typical English digits which is shown in Table I. The user can choose the display language (Bangla or English) while setting up the

display language (Bangla or English) while setting up the clock. A display driver was designed initially for the clock so that valid time could be displayed on the 10-segment display.

TABLE I: English Digits and Their Equivalent Bangla Digits by 10-Segment Display

Bangla Digits	Equivalent Bangla digit shown in our 10-segment display
০	
১	
২	
৩	
৪	
৫	
৬	
৭	
৮	
৯	

In our clock (Fig. 1), there are four display units, each having a total of ten segments which can display either Bangla or English numeric digits. The cathodes of these segments are connected together and this type of connection is commonly known as common-cathode configuration. If the anode is connected to the positive supply while the common cathode is connected to ground, a segment will be lit up. Each of these segments and the four display units can be activated/deactivated (turned on/off) individually using proper combination of voltages to appropriate segment anode and cathode pins.

Manuscript received August 10, 2012; revised October 20, 2012.

Victor Kathan Sarker and M. Ataur Rahman are with the Department of Electrical Engineering and Computer Science, North South University, Dhaka, Bangladesh (e-mail: kathan_bd@hotmail.com).

M. A. Martin is with the Brunei Institute of Technology, Brunei.

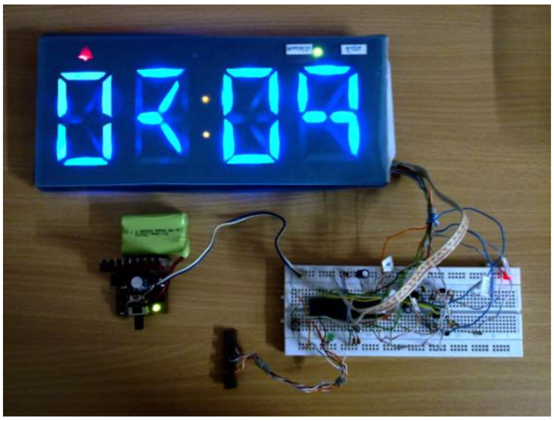


Fig. 1. Digital bangla clock showing time 02:07 pm.

The ‘heart’ of this clock is a microcontroller unit (MCU) and ATmega32 microcontroller is chosen due to the availability in the local market though other MCUs are compatible with this clock design. Manufactures can choose one according to the availability in a certain locality. The popular ATmega32 MCU from ATMEL ® [5] has built upon the famous AVR RISC (Reduced Instruction Set Computer) architecture which is available in most electronics markets in Bangladesh. It is an 8-bit, 40 pin DIP (dual in-line package) integrated circuit with built in powerful instructions set and it is compiled with the popular programming language ANSI C.

The alarm unit consists of one ROM (read only memory) melody IC and very few passive components to generate a suitable alarm sound. In this design, the tiny 3-pin UM66 IC is used to generate a nice sound which is pre-programmed into it and he sound is further amplified using a transistor.

The power supply unit consists of a voltage regulator IC LM7805 (from 78xx family).Few passive elements are used to feed stable, regulated and transient-free 5 Volt to the MCU, alarm unit and the display unit. The coupling capacitors are used at the input, output circuit to filter out the ripples and noises that could be produced from the main power supply or the voltage regulation process.

As it is known, human eyes can refresh the light of image at the rate of 10 Hz. if any other image is placed before the eyes for less than 0.1 second and moved away while viewing a particular image, the eyes can neither distinguish nor detect the presence of the second image. In this clock design, we are using ten *segment-driver* lines and four *position-controller* lines to display the valid time information. We have marked (Fig. 2) the position of the four display units (from right to left) as ‘units’, ‘tens’, ‘hundreds’ and ‘thousands’. Let us assume that we want to display ‘12:34’ on our clock. At first MCU calculates position values for units (in this case 4), tens (in this case 3), hundreds (in this case 2) and thousands (in this case 1).

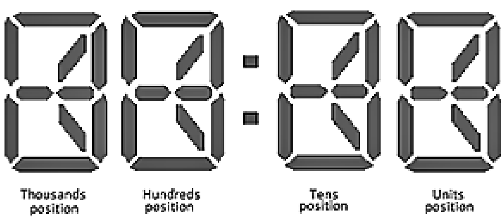


Fig. 2. Bangla clock display segment and their defined positions.

Then the display update process begins. Then, the MCU selects the unit’s position, displays the current position value and switches to the next position. This process is repeated until all the four positions are updated with current digit values for respective positions.

The digit display units are updated according to the hour and minute (*hh:mm*) values at a rate of more than 60 Hz. Thus, it is possible to flicker-freely display the time as if all the digits are being lit up all together what is in fact not true. This clever process of time sharing among the digits is known as *multiplexing*. This is the key to our ‘simplified’ clock display unit design. This helped in using least possible (only 14 pins instead of typical 44 pins) number of pins of the MCU and removing the need of using additional external decoders at the same time.

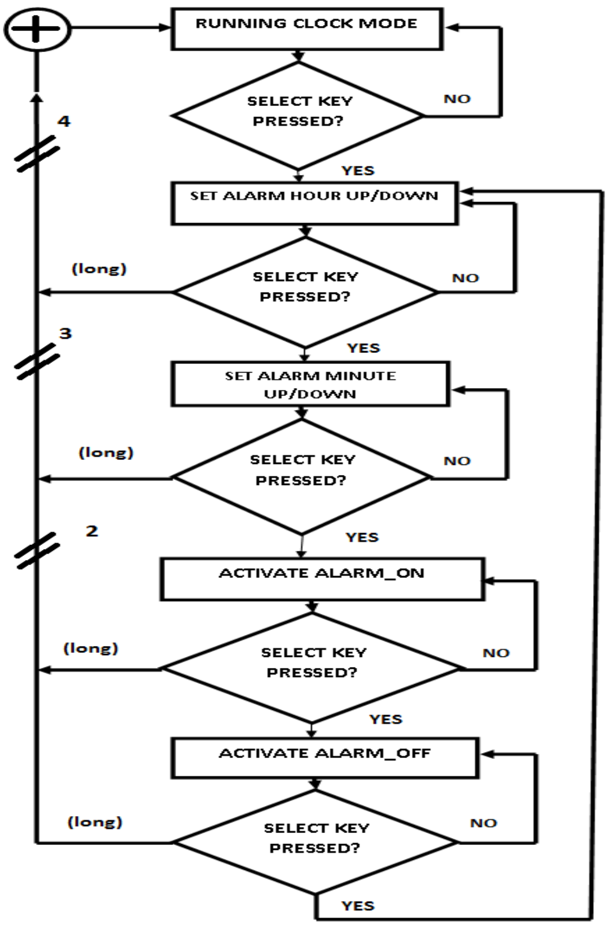


Fig. 3. Alarm setup process in the system.

After the time has been set by using four input keys (*Reset, Select, Down* and *Up*), the clock goes into operation. The MCU uses a quartz-crystal oscillator as the time-base for very simple yet accurate calculation of time. A clock signal interrupt is generated precisely at a rate of 1 Hz by the Timer (counter) of the MCU with the aid of the external crystal oscillator running at 4MHz. An LED blinks according to the clock pulse indicating the seconds. In addition, the MCU calculates minutes and hours by further dividing the clock pulse by 60 and 3600 respectively. AM/PM indication is also done by two other status LEDs. At every clock pulse interrupt stated before, new values are calculated and the display updating process repeats until next interrupt. Thus, keeping the time calculation process in the interrupt routine part helps to achieve high accuracy.

The user can also set an alarm at any time while the clock is running. There is an alarm status LED which will light up once the alarm has been set. The clock checks for a matching time for the set alarm time every minute if alarm is turned on (set). If a match is found, the MCU drives up the alarm out pin which is connected to the alarm unit. Thus a predefined alarm is generated which will continue up to one minute, and then it is turned off to save the power.

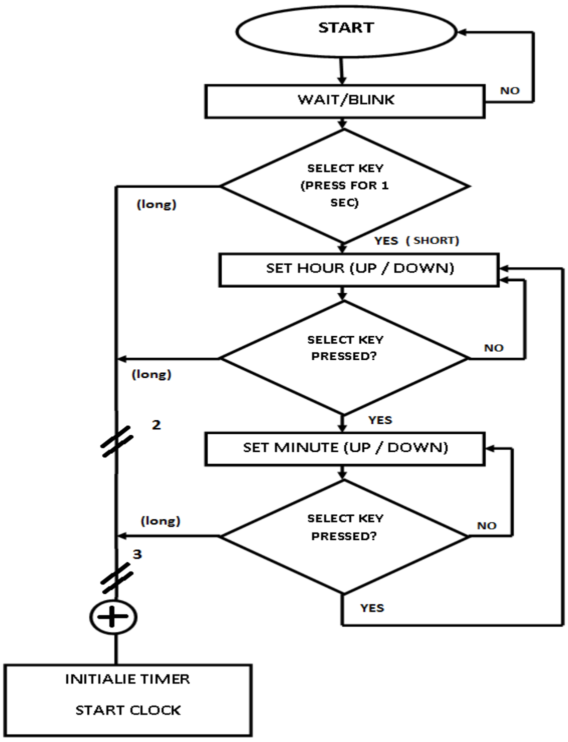


Fig. 4. Time setup process in the system.

III. Technical Data and Calculations

The AVR Timer (Fig. 5) is a register which runs independently and asynchronously to the CPU and can be used to calculate time precisely. The ATMega32 has three separate timers- *timer0*, *timer1*and *timer2*. Also, there are four modes of operation among which we have used the *simple overflow* mode of *timer0*. The clock frequency for *timer0* is set by pre-scaling (dividing the external clock of 4MHz) with a factor of 64. That gives us (4 MHz / 64) = 62.5 kHz. As this timer has a width of 8-bit, a total of 2^8 or 256 levels can be achieved. So, we can have (62.5 kHz / 256) = 244.1 or about 244 interrupts per second. An interrupt is a special type of signal is a special type of signal generated by internal or external event and has the highest priority from the CPU if set up properly. Thus, whenever the timer0 reaches its highest value of 255 (*FF* in Hexadecimal), it overflows (Fig. 6) and sends an interrupt signal to the CPU via the built in ISR (Interrupt Service Routine) of ATMega32.

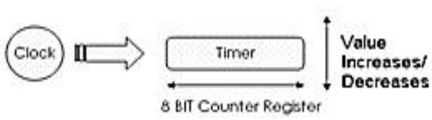


Fig. 5. AVR 8-bit timer operation.

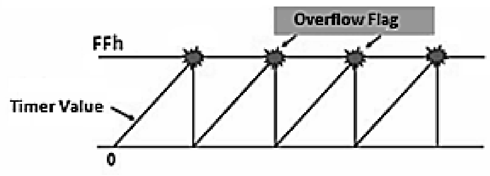


Fig. 6. Timer 0 Overflow process.

In the programming, the rate of interrupts is counted and stored to calculate the time accurately. Here, 244 interrupts accumulates 1 second. Furthermore, seconds are divided by 60 and 3600 to get minute and hour values, respectively. Normally, main routine keeps refreshing the display segments continuously. The hour and/or minute values are calculated while the ISR part is in operation (Fig. 7). The CPU halts while the ISR updates the time values and then resumes refreshing the digits continuously, and so on.

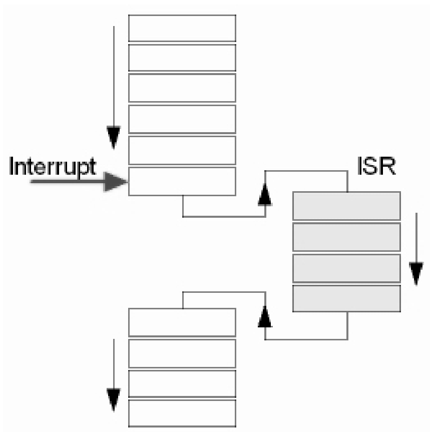


Fig. 7. CPU and ISR operation process.

IV. Conclusion

In the proposed design, a simple, portable cost-effective multifunctional system has been developed and implemented to show the time. Though simple hardware is used, its accuracy is high. Moreover, this clock is functional in dual Language mode both English and Bangla.

References

[1] P. T. Tun, “Development and implementation of microcontroller-based digital clock,” *World Academy of Science, Engineering and Technology*, Singapore. vol. 42, June 2008.

[2] M. S. Arefin, M. A. Dewan, M. I. Khan, and M. S. Islam, “Designing a 24 segment display for Bengali numerical digits and characters,” in *Proc 3rd International Conference on Electrical and Computer Engineering ICECE*, 2004, pp. 549-552.

[3] S. Ahmed and S. Monira, “Designing a 10 segment display for Bangla and English numerals,” in *the Proceedings of ICCIT*, 2007, pp. 602-605.

[4] M. O. Rahman, M. A. Azim, M. S. Chowdhury, and M. N. Islam, “Defferent segment displays for Bangla, English and Arabic digits,” in *the Proceeding of ICCIT*, 2003, pp. 299-302.

[5] A. corporation. (2011). 8-bit AVR Microcontroller with 32Kbytes In-S System Programmable Flash. Datasheet. [Online]. Available: http://www.atmel.com/avr