World Academy of Science, Engineering and Technology
International Journal of Electrical and Information Engineering
Vol:5, No:2, 2011

Design and Implementation of a Microcontroller Based LCD Screen Digital Stop Watch

Mr. Khalid I. Saad, Ms. Nusrat Afrin and Mr. Rajib Mikail

***Abstract*—**The stop watch is used to measure the time required for a certain event. This is different from normal clocks in many ways, one of which is the accuracy of time. The stop watch requires much more accuracy than the normal clocks. In this paper, an ATmega8535 microcontroller was used to control the stop watch, by which perfect accuracy can be ensured. For compiling the C code and for loading the compiled .hex file into the microcontroller, AVR studio and PonyProg were used respectively. The stop watch is also different from traditional stop watches, as it contains two different timing modes namely 'Split timing' and 'Lap timing'.

***Keywords*—**Stop Watch, Microcontroller, Split timing, Lap timing, LCD.

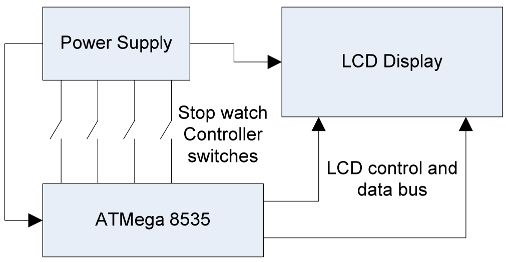

Fig. 1 Block diagram of the stop watch system

I. Introduction

FROM time immemorial, people are trying to measure time. Once, people anticipated about time by the position of the Sun, the Moon and the other stars in the sky. Then one day, the first clock was built. That time, clocks were sundial clocks. Then the evolution went on, mechanical clock, analog electronic clock, digital electronic clock etc. were invented time to time. The history of clock is as old as that of mankind. And the history of stop watch is also not very new.

When people understood the importance of measuring time in their day to day life, they understood the importance of measuring a certain amount of time as well, i.e. the time required to do a certain thing. For example, the time a person takes to win a race or a swimming competition. And to measure the certain period of time, the sand clock was built, which could measure an hour. Then people tried to measure minutes, seconds and also fractions of seconds. The more accurate measurement of time, the faster the human race would be. That is why the research on stop watch is a very important one in the history of technology, so that we can measure time exactly, with a cent percent accuracy.

Like the clock or watch, stop watch was also developed mechanically first. Then the electronic technology was introduced. And nowadays, the embedded system is always there to make our life easier. Like all the electronic devices, stop watches can also be designed and made with the help of a microcontroller, which is more accurate, simpler to understand and easier to operate. In this paper, an accurate and easy-to-use stop watch, which is designed and operated by a microcontroller, is presented.

The model of the microcontroller used here is ATmega8535 and the LCD screen is a 16 × 4 alpha-numeric character LCD display [1], [6]. Four push-to-on control switches and a power supply consisting of a 7805 regulator IC and two 1000 µF capacitors are also used.

Mr. Khalid I. Saad is a graduate in EEE from Bangladesh University of Engineering and Technology (BUET) (phone: +8801817181476; email: Khalid.imtiaz.saad@gmail.com).

Ms. Nusrat Afrin is a graduate in EEE from Bangladesh University of Engineering and Technology (BUET) (email: n.a.pinki@gmail.com).

Mr. Rajib Mikail is a faculty member of EEE in Bangladesh University of Engineering and Technology (BUET), (email: rajib_mikail@eee.buet.ac.bd)

II. Design and Implementation

The main parts of the software design and hardware implementation phase were to select the timing mode, to differentiate among different functions of a single interrupt, the timing itself, the de-bouncing effect and last but not the least, controlling the LCD display and displaying the timing onto it. A total of two timers and two interrupts were used in the stop watch. The usages are summarized as follows:

A. Initialization and Timing Mode Selection

TABLE I
Summary of the Interrupts and Timers

Interrupt / Timer	Function
Interrupt 0	To start and stop the timing
Interrupt 1	To split, lap and reset the timing
Timer 0	Used for de-bouncing
Timer 1	The main timer

First, the ports, timers and interrupts are set to their proper operating modes [9]. There are two timing modes, namely 'split timing' and 'lap timing' [5]. In split timing mode, when the timing is split, the instance of splitting is shown and timing is continued. On the other hand, when the timing is lapped in the lap timing mode, the timing is restarted after showing the lapping instance. When the stop watch is switched 'ON', the user selects the timing mode. The timing cannot be started, until the mode is selected (by pressing either of PORT_C0 or

PORT_C1), as the timing selection is done by 'polling method', i.e. the micro-controller continues to search for input, until it is not given. No other inputs (e.g. start, stop etc.) are accepted either [7] – [8].

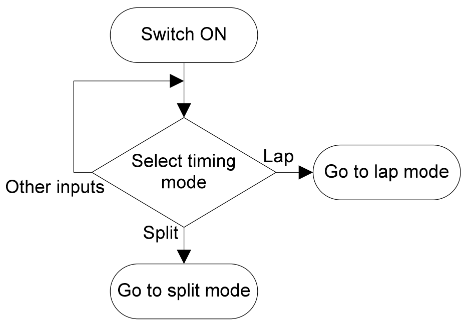

Fig. 2 Timing mode selection

B. Start / Stop the Timing with Interrupt 0

After mode selection, the stop watch is ready to be used for counting time. Interrupt 0 is used to both start and stop [1], [2], [9]. That is why, it is essential to check whether the stop watch is currently stopped (so that it will be started by Interrupt 0) or running (so that it will be stopped by the same). When the interrupt 0 is pressed for the first time, the timer is started. For pressing next time, it is stopped. The next time, timer is resumed and so on.

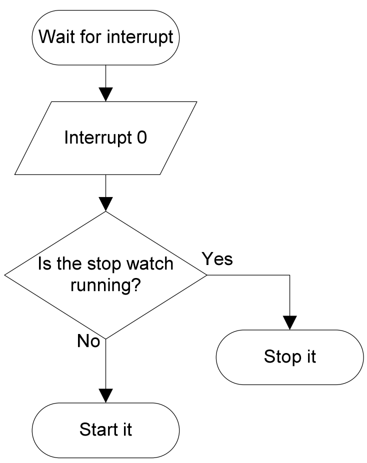

Fig. 3 Interrupt 0

C. Split / Lap / Reset the Timing with Interrupt 1

Interrupt 1 is used to split or lap, if the timer is running; and to reset, if the timer is stopped. The checking of whether the timer is running or not is done by counting the number of instances the interrupt 0 is pressed. The check-points can be shown as follows:

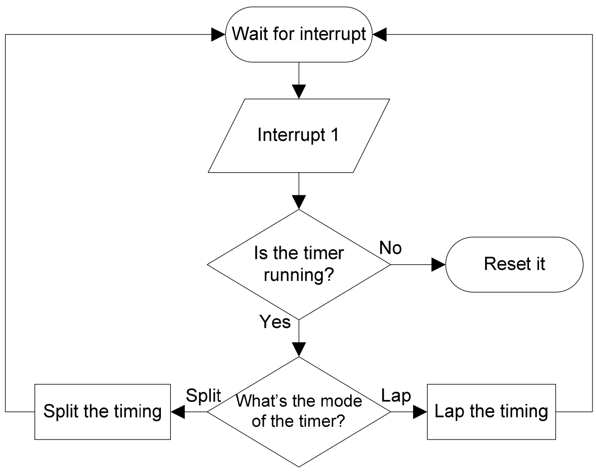

Fig. 4 Interrupt 1

III. TIMING PRINCIPLE

The timing is done by the timer 1 of the micro-controller [1], [2], [9]. It is set to 'output compare mode' and an output compare value is given so that when the timer reaches that value, a hundredth fraction of a second (i.e. 10 millisecond or 1 degree) is passed. This value is based on the frequency of the timer. For selecting it properly, necessary calculation and calibration is done. Either the internal clock or an external oscillator can be used for timing. In this project, the internal clock of the microcontroller was used. When 1 degree is passed, the right digit of degree is increased by 1. If it becomes equal to 10, it is reset and the left digit of degree is increased by 1. Again, if the left digit of the degree reaches 10, both the digits of degree are reset and the right digit of second is increased by 1. Thus the timing continues.

The timing is done in this way, since the LCD cannot print numbers unless they are treated as characters [1]. When the LCD prints a number, e.g. 58, it does not print the value 58; rather it prints the character '5' and then the character '8' (which ASCII value is 53 and 56 in decimal respectively).

IV. OPERATION

As discussed earlier, the number needed to be printed in the LCD is first separated into two digits, then the ASCII value of 0 (i.e. 48) is added with the individual digits to convert them to a character. Then the digits are shown in the LCD [1].

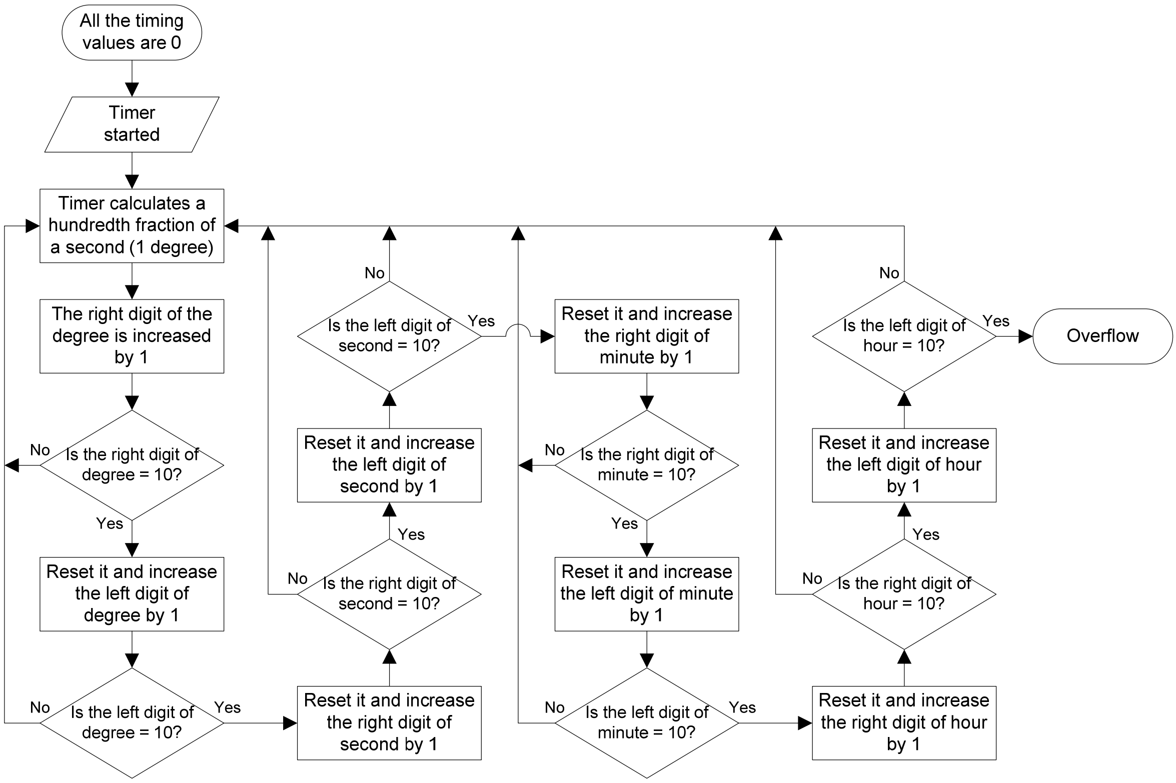

Fig. 5 The timing principle

The bouncing effect is a common practical problem as far as switching is concerned. The interrupts fed to the microcontroller must be a single fine edge (in this paper, a 0 to 1 transition) [9]. But practically, when a switch is pressed, some noise is produced and more than one successive and unwanted interrupts occur. To prevent this, the interrupt must be disabled just after the first interrupt. This is done by timer 0, which is set to overflow mode [2], [9]. When an interrupt (either start/stop or split/lap/reset) occurs, the timer 0 starts timing and before finishing the timing (which is approximately half a second), all the interrupts are kept disabled. When timer 0 overflows, it triggers its internal interrupt, which task is to enable the interrupts again. Thus, for about half a second after a fine interrupt edge, unwanted interrupts are prevented from occurring.

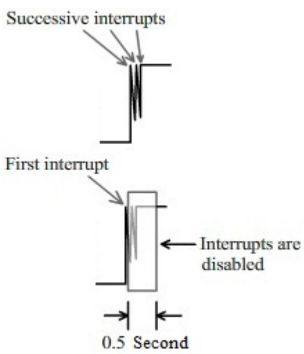

Fig. 6 De-bouncing

V. Data and Circuit Diagram

The port A of the microcontroller is used for displaying the time on the LCD, which is operated in 8 bit mode. The 3 LSBs of port B is used to control the LCD, by sending the control words to it. Four push-to-on control switches are connected to the 2 LSBs of port C (used to get the mode selection input) and the interrupts. These switches are for the end users to use the stop watch. The power supply for the whole system is designed with a 7805 voltage regulator and two capacitors of 1000 µF each as shown in figure 7 [10].

The microcontroller clock frequency is 2 MHz with a prescalar 8 [9]. The output compare value that is used for timer 1 to measure a hundredth fraction of a second is 1249. From the formula of microcontroller timer, the frequency of timing can be calculated as follows:

$$\text{Timer Frequency [9]}, f_{OCn} = \frac{f_{clk\_I/O}}{2\,.\,N\,.\,(1+OCR_n)} = \frac{2\times10^{6}}{2\times8\times(1+1249)} = 100\ \text{Hz} \qquad (1)$$

So, the time period of the timer is 1/100 = 0.01 second (1 degree).

The circuit diagram of the whole system is as follows:

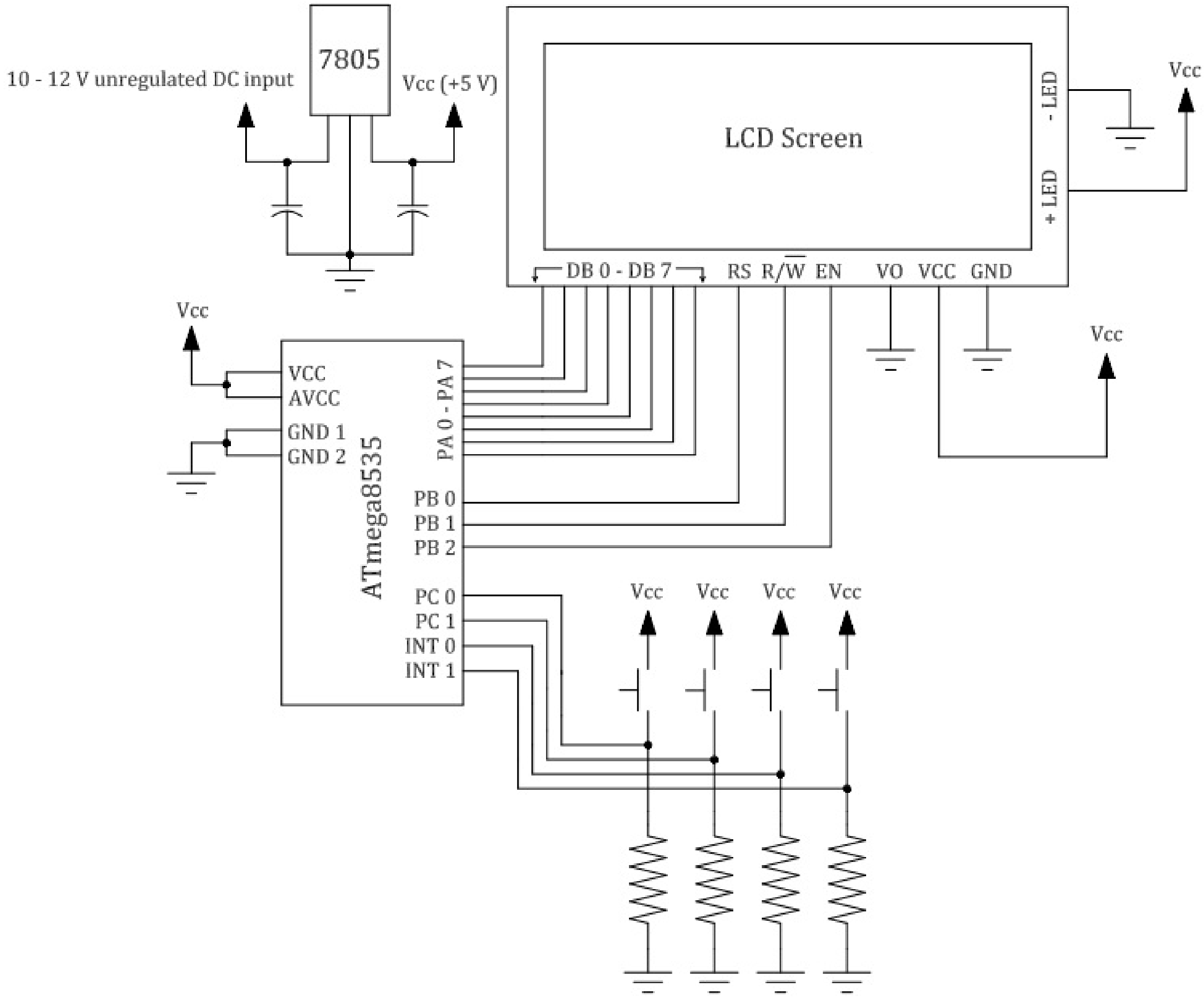

Fig. 7 Circuit diagram of the stop watch

VI. Conclusion and Future Prospects

The cost for this project is more or less like follows:

TABLE II
COST ANALYSIS

Equipment	Cost in BDT (local currency)	Cost in USD
Microcontroller	250	3.5 – 4
LCD	400	5 – 6
Switches	80	1 – 1.5
Others	100	1 – 1.5
Total	*830*	*10 – 13*

In this paper, the model of the microcontroller is ATmega8535, which is a 40 pin IC [9]. But the number of pins used for the system is only 15. So, a smaller microcontroller can be used to optimize the solution. The timer here is run by the internal clock, which is made by resistors and capacitors. That is why; it might not be precise always. A better way to get the most accurate data may be using a crystal oscillator instead [9].

VII. Experimental Results

A few snapshots of the system and the stop watch are given here.

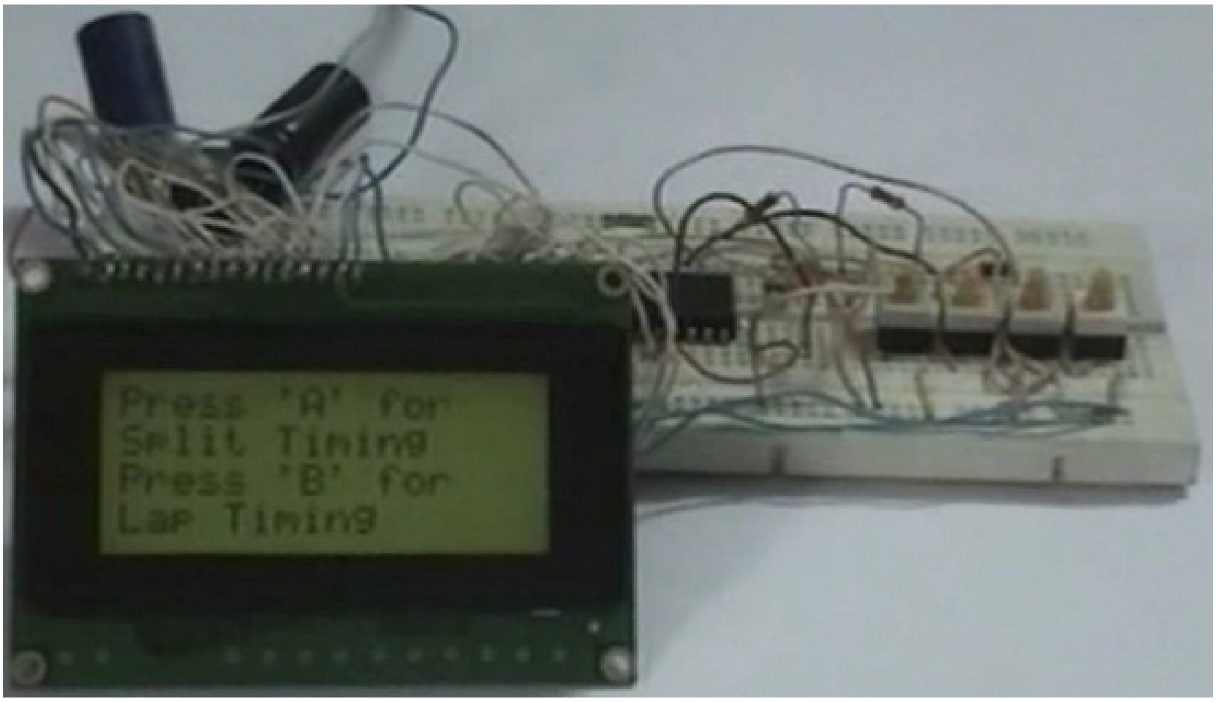

Fig. 8 Snapshot of the whole system

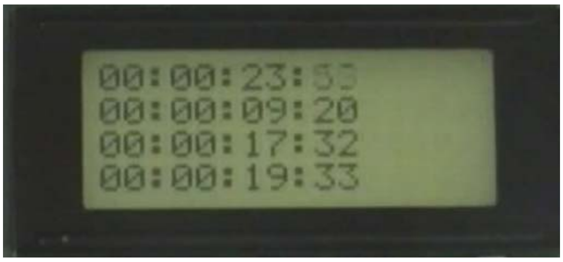

Fig. 9 Split timing

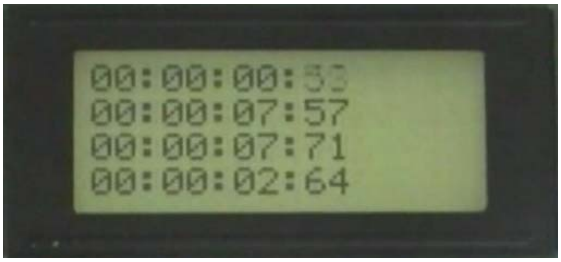

Fig. 10 Lap timing

REFERENCES

[1] *Embedded Systems Design with the Atmel AVR Microcontroller* by Steven F. Barrett.
[2] *Atmel AVR Microcontroller Primer: Programming and Interfacing* by Steven F. Barrett, Daniel J. Pack.
[3] *Microcontrollers: theory and applications* by Ajay V Deshmukh.
[4] *AVR RISC microcontrollers handbook* by Claus Kühnel.
[5] http://en.wikipedia.org/wiki/Stopwatch
[6] *Single- and multi-chip microcontroller interfacing* by G. Jack Lipovski.
[7] *C programming for microcontrollers: featuring ATMEL's AVR butterfly and the free WinAVR compiler* by Joe Pardue.
[8] *Exploring C for microcontrollers: a hands on approach* by Jivan S. Parab, Vinod G. Shelake, Rajanish K. Kamat.
[9] http://www.alldatasheet.com/datasheet-pdf/pdf/175175/ATMEL/ATMEGA8535.html.
[10] http://www.alldatasheet.com/datasheet-pdf/pdf/69437/KEC/7805.html.

Mr. Khalid I. Saad was born in November 05, 1986. He is a young graduate on EEE, passed from Bangladesh University of Engineering and Technology (BUET), in October 2009. After graduation, he joined a leading GSM mobile operator in Bangladesh in November 2009, in which he is currently working in the IN Planning department. He was a student member of **IEEE** since 2005, continuing the membership in his whole undergraduate academic career. He is also an associate member of **IEB** (Institute of Engineers Bangladesh) and life member of **BUETAA** (BUET Alumni Association). He is now planning to dedicate himself into further studies and research in the fields of EEE.

Ms. Nusrat Afrin was born in September 23, 1987. She is also a graduate on EEE, BUET and passed in October 2009. She was a student member of **IEEE** and is a life member of **BUETAA**. She is now working in a prominent GSM operator in Bangladesh since April 2010. She wants to dedicate herself in the research fields of EEE.

Mr. Rajib Mikail is a faculty of EEE, BUET. He did his graduation from BUET and joined the same university as a lecturer in 2006. Later he has gone abroad to pursue higher studies. Currently he is a PhD student at University of Akron, Ohio, USA.