Implementation of Digital Stopwatch with 7-Segment Display
Bikash Jyoti Das, Nayan Patowary, N. Jugal Hansraj, LPi P Naidu, Sanjeeva Jeevana Jyothi
BTech Student, Assistant Professor
Department of ECE
Lords Institute of Engineering & Technology, Hyderabad, India

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, 8051 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, 7-Segment Display.

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-touse stop watch, which is designed and operated by a microcontroller, is presented. The model of the microcontroller used here is 8051 and the 7-Segment display is used and six push-to-on control switches and a power supply consisting of a 7805 regulator IC and two 1000 µF capacitors are also used.

II. BLOCK DIAGRAM

A Power supply is a component that supplies power to at least one electric load. Typically, it converts one type of electrical power to another, but it may also convert a different form of energy – such as solar, mechanical, or chemical - into electrical energy. A power supply provides components with electric power. We couldn’t wait to show you how much fun it is to make the microcontroller interact with real physical devices. So in this section we will take a short side step from the business of concepts and theories to teach you how to connect switches and LEDs to the microcontroller. We will use switches to input data and use LEDs to output results. In this chapter, we will illustrate the design, construction and testing of an embedded system. The switch LED and LaunchPad will then be combined to create a system. You will need a solid understanding of Ohm’s Law, so you may need to review current, voltage, power, and resistance. The 8051 Microcontroller is one of the basic type of microcontroller, designed by Intel in 1980’s. This microcontroller was based on Harvard Architecture and developed primarily for use in embedded systems technology.
Normally, this microcontroller was developed using NMOS technology, which requires more power to operate. Therefore, Intel redesigned Microcontroller 8051 using CMOS technology and their updated versions came with a letter C in their name, for instance an 80C51 it is an 8 bit microcontroller. These latest Microcontrollers requires less power to operate as compared to their previous versions. The 8051 Microcontroller has two buses and two memory spaces of 64K X 8 size for program and data units. It has an 8 bit processing unit and 8 bit accumulator units. A seven-segment display (SSD), or seven-segment indicator, is a form of electronic display device for displaying decimal numerals that is an alternative to the more complex dot matrix displays.

III. 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 7-segment 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

SUMMARY OF THE INTERRUPTS AND TIMERS

Interrupt / Timer Function Interrupt 0 To start and stop Interrupt 1 To split, lap and reset Timer 0 Used for de-bouncing Timer 1 The main timer First, the ports, timers and interrupts are set to their proper operating modes. There are two timing modes, namely ‘split timing’ and ‘lap timing’. 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 microcontroller continues to search for input, until it is not given. No other inputs (e.g. start, stop etc.) are accepted.

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. 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 interval 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.

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.

IV. TIMING PRINCIPLE

The timing is done by the timer 1 of the micro-controller. 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.

V. OPERATION

As discussed earlier, the number needed to be printed in the SEGMENT DISPLAY 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 7-Segment display.

VI. SOFTWARE USED

1. KEIL software Keil Software, world’s leading developer of Embedded Systems Software, makes ANSI C compilers, macro assemblers, real-time kernels, debuggers, linkers, library managers, simulators, integrated environments, and evaluation boards for the 8051, 251, ARM7, and C16x/ST10 microcontroller families. Keil Software implemented the first C compiler designed from the ground-up specifically for the 8051 microcontroller.

2. KEIL compiler The Keil C51 C Compiler for the 8051 microcontroller is the most popular 8051 C compiler in the world. It provides more features than any other 8051 C compiler available today.

3. Embedded C Embedded C is a set of language extensions for the C Programming language by the C Standards Committee to address commonality issues that exist between C languages for different embedded systems. Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations. In 2008, the C Standards Committee extended the C language to address these issues by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as, fixed-point arithmetic, named address spaces, and basic I/O hardware addressing.

VII. CONCLUSION

This paper represents 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, 8051 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, atv studio and PonyProg were used respectively.

VIII. REFFERENCE

[1]. Embedded Systems Design with the Atmel AVR Microcontroller by Steven F. Barrett.


[4]. AVR RISC microcontrollers handbook by Claus Kühnel.


[7]. C programming for microcontrollers: featuring ATMEG's AVR butterfly and the free WinAVR compiler by Joe Pardue.