DIGITAL WATCH

DEVANG KHAMAR
13BEC023
DEPT. OF ELECTRONICS AND COMMUNICATION
INSTITUTE OF TECHNOLOGY, NIRMA UNIVERSITY.
AHMEDABAD, INDIA.

PRASHANT GANDHI
13BEC029
DEPT. OF ELECTRONICS AND COMMUNICATION
INSTITUTE OF TECHNOLOGY, NIRMA UNIVERSITY.
AHMEDABAD, INDIA

ABSTRACT: **This paper presents an efficient design of a Digital watch. To implement the digital watch the microcontroller ATmega16A and RTC ds3231 are used. The output can be seen on 16x2 LCD. The main purpose of selecting this project is to explore more features of microcontrollers which are available in it. The basic information about microcontroller ATmega16A and RTC module is given in this paper. The algorithm and flow-chart of how time will be set and how communication and interfacing of RTC module and microcontroller is shown in this paper.**

KEYWORDS: **Microcontroller ATmega16A, RTC ds3231.**

I. INTRODUCTION

The aim of the project is to explore more about microcontrollers and their different features. For example this project is using microcontroller ATmega16A which has addition feature of Power Reduction Register (PRR) through which we can individually operate each blocks in it like interrupt and we can enable and disable whenever we want. Digital watch is designed using microcontroller ATmega16A, RTC module ds3231 and 16x2 LCD. The advantage of use of ATmega16A microcontroller is it has 16 kB flash memory and it's very low power consumption. It has three modes active mode, power-down mode and power-save mode. In active mode it consumes only 0.2 mA current. Also this project uses RTC module ds3231. It is a low-cost, extremely accurate I2C real-time clock (RTC) with an integrated temperature compensated crystal oscillator (TCXO) and crystal. The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted. It has dedicated temperature register which measure the temperature. The project concept is to set date and time and also show the temperature.

II. FEATURES OF ATMEGA16A

ATmega16A is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16A achieves throughputs approaching 1 MIPS per MHz allowing the system designed to optimize power consumption versus processing speed. It has 131 Powerful Instructions – Most Single Clock Cycle Execution, 32 x 8 General Purpose Working Registers, Up to 20 MIPS Throughput at 20MHz, On-chip 2-cycle Multiplier, 16KBytes of In-System Self-Programmable Flash program memory, 512Bytes EEPROM, 1KBytes Internal SRAM, and Two 8-bit Timer/Counters with Separate Pre-scalar and Compare Mode, Byte-oriented 2-wire Serial Interface (Philips I2C compatible). The special microcontroller features are External and Internal Interrupt Sources and Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby. It has three modes active mode, idle mode and power-down mode. In active mode it consumes only 0.6 mA current.

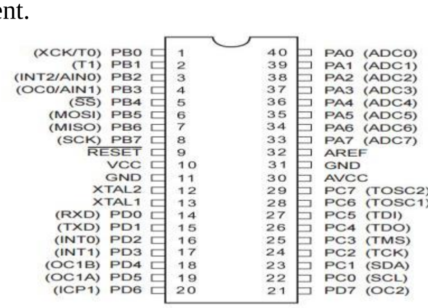

Figure 1 ATmega16A pinout

III. RTC MODULE DS3231

The DS3231 is a low-cost, extremely accurate I2C real-time clock (RTC) with an integrated temperature compensated crystal oscillator (TCXO) and crystal. The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. It has highly accurate RTC completely manages all timekeeping functions like Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year, with Leap-Year Compensation Valid Up to 2100. It has dedicated register for Seconds, Minutes, Hours, Temperature and they have dedicated addresses for that. We can set date and time through this registers.

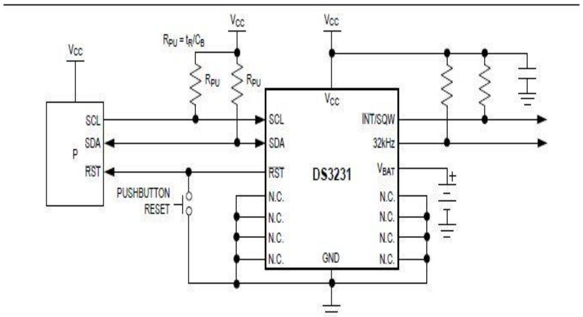

Figure 3 Typical ds3231 circuit diagram

IV. ALGORITHM

The algorithm for the Digital watch using ATmega16A microcontroller and RTC module ds3231 is as below.

1) Power On
2) Initialize timer0 to generate delay in mS
3) Initialize LCD in 4-bit mode
4) Initialize timer1 to generate interrupt at every 1 S
5) Otherwise enter in sleep mode
6) Check for interrupt
7) If interrupt occurs then check for the source of the interrupt and if there is no interrupt occurs then go to sleep mode
8) If source of interrupt is button then go to button Interrupt Service Routine and if timer is the source of the interrupt then go to timer Interrupt Service Routine.
9) If button is source then check which button is pressed B0 or B1
10) If B0 is pressed then check for duration of press if it is long press then go to set time mode and if it is short press then show the temperature
11) Exit the interrupt routine and go back to main function
12) If B1 is pressed then check for the duration of press if it is long press then go to set date function to set date and if it short press then show the date and exit the interrupt routine and go back to main function
13) If timer is the interrupt and timer interrupt service routine is called then disable the clock divider
14) Then fetch time and date and temperature via I2C from bus and then end the interrupt routine and go back to main function.

V. FLOW CHART

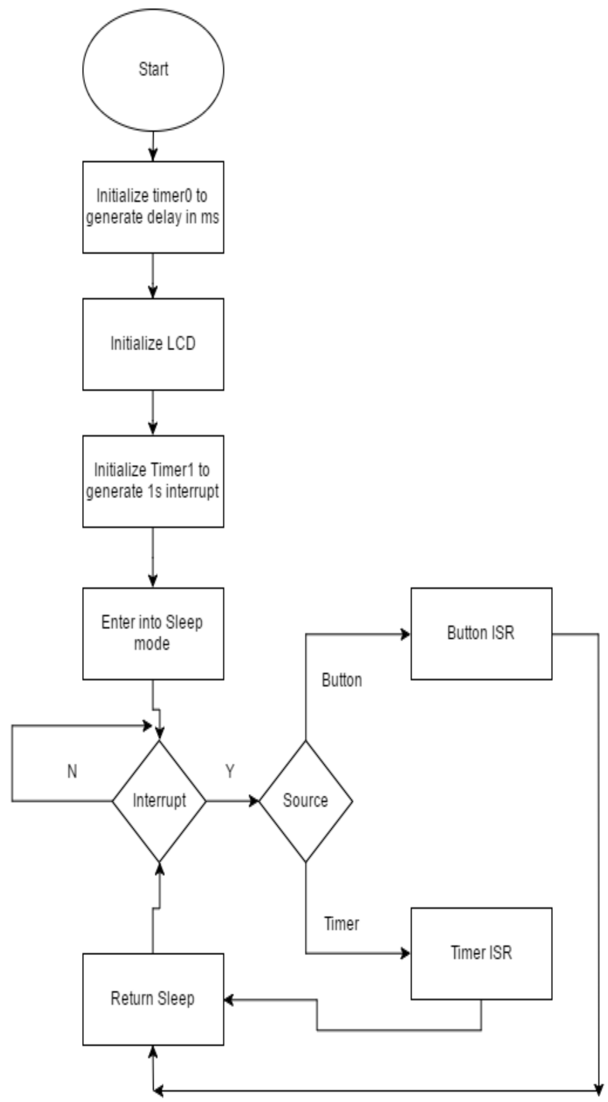

Figure 3.1 Flow-chart

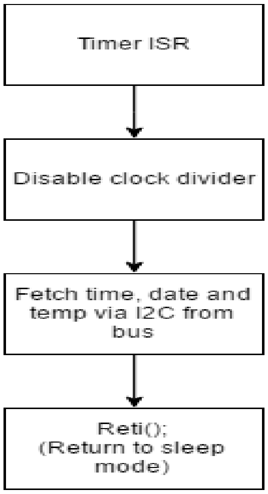

Figure 3.2 Flow-chart (continue)

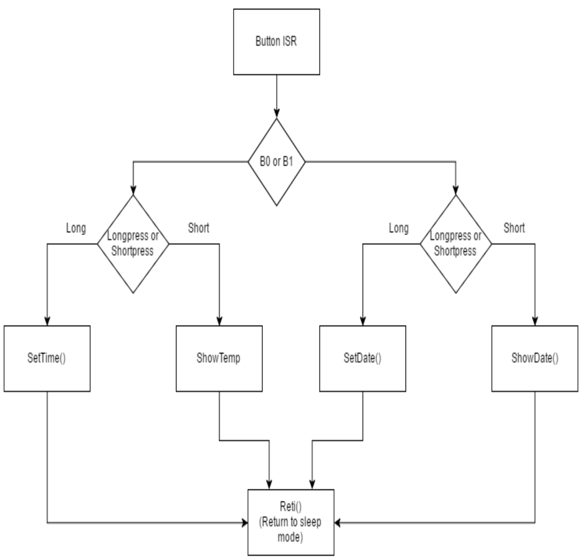

Figure 3.3 Flow-chart (Continue)

VI. SIMULATION

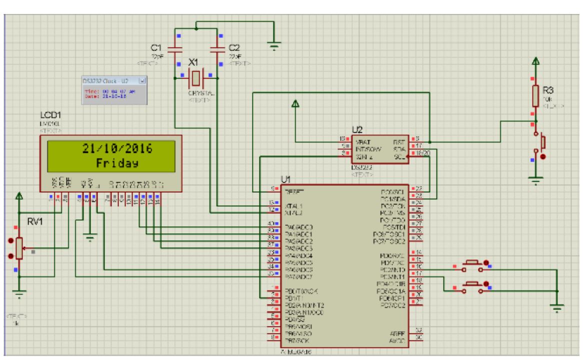

Figure 4.1 Proteus simulation date output

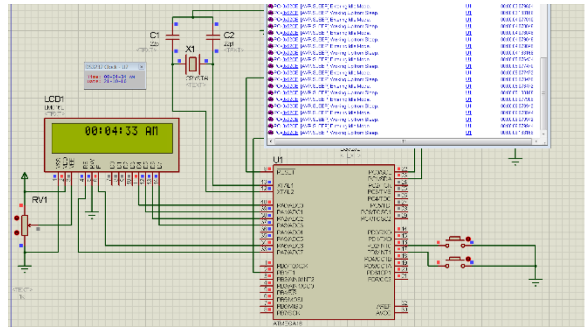

Figure 4.2 Proteus simulation time output

VII. REAL CIRCUIT OUTPUT

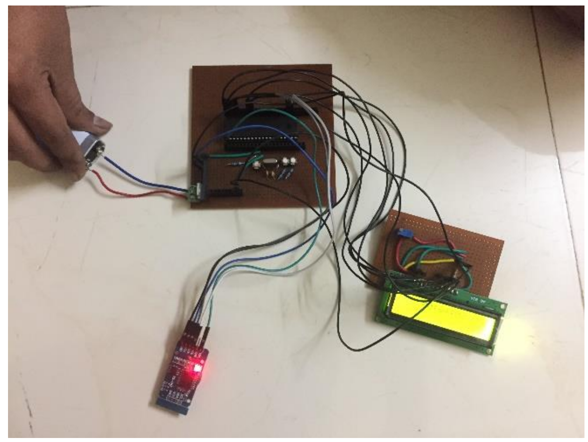

Figure 5.1 Real circuit with ATmega16A, LCD and ds3231 RTC module

Figure 5.2 LCD output

VIII. CONCLUSION

We explored the different microcontroller which we can use for the project and saw that we cannot use ATtiny2321 microcontroller because it has only 2kB flesh memory which is not sufficient for this application so we used ATmega16A microcontroller which has 16kB flesh memory. Also we explored different features of the microcontroller like low power modes, I2C communication and PRR- Power Reduction Registers. Also we studied about RTC module ds3231 and its feature

.

IX. REFERENCES

[1] www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf

[2] Datasheets.maximintegrated.com/en/ds/DS3231.pdf.

[3] http://nongnu.org/avr-libc/user-manual/

[4] https://www.youtube.com/watch?v=7aDjYqOMu9w

[5] http://stackoverflow.com/questions/7525373/eclipse-does-not-resolve-all-includes

[6] http://extremeelectronics.co.in/category/avr-tutorials/page/2/

[7] http://www.instructables.com/id/How-to-get-started-with-Eclipse-and-AVR/step12/Compilation-Errors/

[8] http://www.embedds.com/8-bit-timercounter0-operation-modes-part1/

[9] http://www.avrfreaks.net/forum/tut-modularizing-c-code-managing-large-projects?name=PNphpBB2&file=viewtopic&t=48535

[10] http://stackoverflow.com/questions/8148235/eclipse-cdt-shows-semantic-errors-but-compilation-is-ok

[11] https://www.youtube.com/watch?v=7aDjYqOMu9w

[12] http://www.atmel.com/Images/doc8163.pdf

[13] HD44780 Lcd Controller: https://www.sparkfun.com/datasheets/LCD/HD44780.pdf

[14] pico Power basics: http://www.atmel.com/Images/doc8349.pdf

[15] Ext Interrupts: http://www.atmel.com/Images/Atmel-8468-Using-External-Interrupts-for-megaAVR-Devices_ApplicationNote_AVR1200.pdf

[16] TWI Master: http://www.atmel.com/Images/Atmel-2564-Using-the-TWI-Module-as-I2C-Master_ApplicationNote_AVR315.pdf

[17] Peter Fleury TWI master Library: http://homepage.hispeed.ch/peterfleury/doxygen/avr-gcc-libraries/group__pfleury__ic2master.html

[18] ATmega 16A - Data sheet Complete: http://www.atmel.com/Images/Atmel-8154-8-bit-AVR-ATmega16A_Datasheet.pdf

[19] Migration note: http://www.atmel.com/Images/doc8163.pdf

[20] DS3231 Data sheet: https://datasheets.maximintegrated.com/en/ds/DS3231.pdf

[21] https://www.maximintegrated.com/en/products/digital/real-time-clocks/DS3231.html