ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijareeie.com

Vol. 6, Issue 5, May 2017

Automated College Bell System with Wireless Control

Vaishnavi.D.R[1], Neha Khanum[2], Apoorva Singh A[3], Sumaya Afreen[4]

UG Student, Dept. of EEE, GSSS Institute of Engineering and Technology for Women, Karnataka, India [1,2,3,4]

ABSTRACT: Time scheduling forms the heart of every commercial and non commercial organisation's operation. In order to follow the planned schedule, organisations often employ the method of assigning an employee the task of keeping a watch on the time and ringing the bell/alarm when the schedule has elapsed, or is yet to begin. This method is not only a waste of human resource, but is also highly inaccurate and inconsistent. While there are solutions available in market to automate this task, they are also cumbersome to operate and overpriced for the limited features they offer. The project discussed in this paper was built using an Arduino Uno which has the ATMega328p microcontroller, DS3231 RTC module for timekeeping and HC-05 Bluetooth module for wireless communication. This design provides multiple time table setting features along with an elaborate event holding capacity which is far more versatile than the timer system commercially available in the market.

KEYWORDS: Arduino Uno, OLED display, Automation, RTC, Bell timer, Menu (user interface), U8glib.

I.INTRODUCTION

The world is running a race against time. As much as it is important to schedule our tasks, it is also important that this schedule is being followed. For educational institutions that too, time keeping forms a vital but invisible task. The automated bell system that is discussed in this paper is a mechanism developed to eliminate the human intervention in the process of ringing the college bell.

It is not only necessary to automate this mundane task of time keeping but is also essential that the process of feeding in the timetable must not be laborious or inconvenient in any manner. It must also be noted that educational institutes often have distinct timetable for regular working days and on days of monthly assessment and exams.

While the commercially available automatic bell timers do quiet some justice in solving these problems, there does seem to be a larger scope to improve their capability with the use of available contemporary technologies. It is not true that, just by upgrading to the latest technology one can solve problems in better ways. But the key features of an embedded system project, such as available program and data memory of the microcontroller, available number of general-purpose-input-output pins also play a major role in setting the scale of the project.

Since the primary aim of this project was to eliminate human intervention in the bell ringing process, the model was built around the following objectives,

- To design a low cost automatic bell system.
- Developing an algorithm to set and retrieve the time in hassle free manner.
- A robust model that could last for years with the least maintenance.
- Building the timer system with following features:
    - Having provision for distinctive long bell and short bell.
    - Able to store and handle multiple timetables.
    - Easy navigation and layman friendly user interface (UI).

The project could be carried out using various technologies such as Arduino Uno, Raspberry Pi, Beaglebone black and other such prototyping boards. Each of these boards makes the project more flexible and stringent in their own ways;

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijareeie.com

Vol. 6, Issue 5, May 2017

like, the Beaglebone black would help with ease of programming and memory specifications. But the cost of project would be unnecessarily high. Arduino Uno on the other hand, poses limitations with respect to RAM capacity and number of general-purpose-input-output pins available. However we have built this project using Arduino Uno due to the low cost solution it provides and its ability to incorporate wide range of peripherals. Also the vast volume of libraries made available by Arduino community makes Arduino Uno the best choice for this project.

II.RELATED WORK

A Survey was carried on various existing automated bell systems to compare with the proposed model and their choice of components. Many automated bell systems were built using 80C51 and AT89S52 microcontroller with DS1307 RTC [8, 9, 10]. 80C51 microcontroller is of CISC architecture, hence lot of effort in programming is needed even to execute basic tasks. Also it does not provide flexibility in incorporating peripherals.

All the commercially available models have a 3 button input method. Meaning, every time a person wants to set the number '9', they will have to press the increment or decrement button 9 times at least. The proposed model uses a 4x4 matrix keypad which has dedicated keys for navigation and each single digit number can be input in only one key press as the numeric keys are only used for entering in the values. Also, the existing models make use of DS1307 RTC which is less accurate in terms of time keeping as it is not temperature compensated.

Thus the components chosen for the following project helped us to overcome flaws and limitations of the existing system.

III.COMPONENTS USED

1. Arduino Uno [1] [2]
Arduino Uno is an entry level microcontroller board which is driven by the ATMega328p microcontroller. ATMega328 is a single chip microcontroller created by Atmel in megaAVR family. These are modified Harvard Architecture 8bit RISC single chip microcontroller with 20MHz clock oscillator, 32kB Flash, 2kB SRAM, 14 I/O programmable pins, 6 channel 10 bit ADC and 6PWM output pins.

It has an easy to learn language and libraries based on C++ language and IDE environment for proper programming interface. Arduino is regarded as platform independent Hardware and can operate on Windows, Linux, and MAC operating system. The board can be powered through the USB connection or with an external power supply. It can operate on an external supply of 6-20V. But the recommended voltage range is 7-12V.

2. DS3231 RTC module [3]
DS3231 is an extremely accurate I2C Real time clock with an integrated temperature compensated crystal oscillator. The device incorporates battery backup and maintains accurate time keeping when supply to device is interrupted. RTC maintains hours, minutes, second, year, month, day and date information. Clock operates either in 12hour or 24 hour format with AM/PM indicator. Data and Addresses are transferred serially through an I2C bidirectional bus.

3. Bluetooth HC-05 module
The HC-05 module is an easy to use Bluetooth Serial Port Protocol module, designed for wireless serial connection setup [4]. The HC-05 Bluetooth Module can be used in a Master or Slave configuration. The Role of the module can be configured only by AT commands. It provides UART interface with programmable baud rate. The default baud rate is 34800.

4. OLED display [7]
The OLED display measures 0.96" diagonally; it is still very readable due to the high contrast of the display. The display makes its own light, hence no backlight is required. This reduces the power required to run the OLED. On average the display uses about 40mA from the 3.3V supply.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijareeie.com

Vol. 6, Issue 5, May 2017

5. 4X4 Matrix Keypad

Matrix keypad is a good alternative to the regular push buttons as they provide with more number of inputs while still occupying lesser number of input pins of the microcontroller board [5]. The keys are arranged in a matrix form of 4 rows and 4 columns. Hence, for a keypad with 16 inputs, the number of input lines required is only 8.

6. EEPROM 8k

The AT 24C08A is 8K bit electrically erasable programmable memory chip, internally organized with 64 pages of 16 bytes each. It requires 10 bit data word for random word addressing [6]. The memories operate with a supply value as low as 1.8V. The memories are compatible with the I2C standard, two wire serial interfaces which uses a bi-directional data bus and serial clock.

7. 5v Single channel relay

The relay is used to switch an AC device on and off. It operates on 5V DC and AC side terminals are accessible with screw terminals. This relay can handle current up to 10A.

IV.DESIGN AND OPERATION

The design of the project is explained in this section. The components are grouped into four blocks based on their function. This can be observed in Fig. 1.

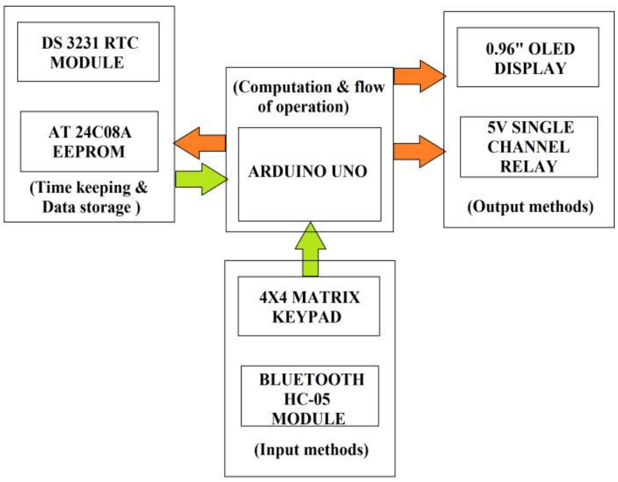

Fig. 1 Block diagram of automated college bell timer system with wireless control.

While the green arrows denote that the data is flowing into Arduino, orange arrows indicate that data is being flown out of Arduino. The RTC module, OLED display and the external EEPROM are connected to the Arduino through I2C interface. I2C interfacing has proven to be very advantageous here, as it allows us to connect numerous peripherals to the microcontroller board while still using only two pins of the board. The Arduino is programmed to retrieve date and time values from the RTC and display it in real time on the OLED display as seen in Fig. 2.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijareeie.com

Vol. 6, Issue 5, May 2017

Fig. 2 Clock display on OLED display in real time.

While this process is being carried out, the Arduino also simultaneously compares the real time values of Hour and Minute provided by the RTC with that values of Hour and Minute stored in the external EEPROM. This process can be understood by referring the flowchart shown in Fig.3.

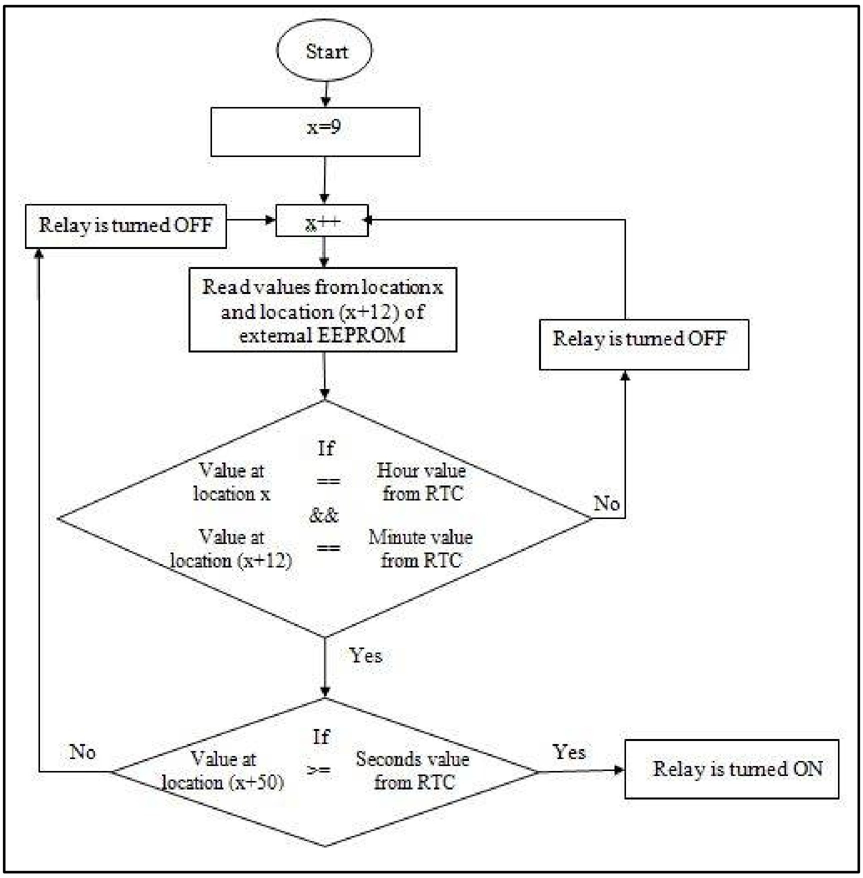

Fig. 3 Flowchart of proposed system (Illustrated for short bell operation)

When the stored and real time values of Hour and Minutes match, the Arduino compares the Seconds value from the RTC with the value stored in EEPROM that indicates duration of bell ring in Seconds. The relay is then kept active for this duration and then turned off. When relay is turned ON, the bell rings. A relay is necessary here as most of the bells operate at 230v, 10A AC ratings.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijareeie.com

Vol. 6, Issue 5, May 2017

The desired values of Hours, Minutes and Bell duration are stored in predefined locations of external EEPROM. So the Arduino would not end up comparing the wrong set of data (values). The memory allocation for storing values of Hour and Minutes as per required timetable in the external EEPROM is described in Fig. 4. Since our model has provision to store timings as long bell or short bell, the memory is allocated separately yet in a systematic order, so that retrieving these values would require less effort. The data stored in these locations is used in a manner illustrated using flowchart in Fig.3.

Shortbell	Location	10	11	12	13	14	15	16	17	18	19	20	21
	Hour value	09	11	12	14	15	00	00	00	00	00	00	00
	Location	22	23	24	25	26	27	28	29	30	31	32	33
	Minute value	50	45	35	50	40	00	00	00	00	00	00	00

Long bell	Location	40	41	42	43	44	45
	Hour value	09	10	10	13	14	00
	Location	46	47	48	49	50	51
	Minute value	00	40	55	25	00	00

	Location	Duration value
Short bell	60	5
Long bell	61	10

Fig. 4 Block representation of memory allocation in external EEPROM

V.IMPLEMENTATION

A. NAVIGATION

To make the process of feeding in the timetable less intimidating, a menu system was developed. U8g library was used to build the graphical interface. The non-numeric keys were purely dedicated for navigation and ease of operation. The navigation can be understood in this section by referring Fig. 5.

1. State 1: Default state. When '*' is pressed, transits to state 2.
2. State 2: When 'A' is pressed once, option 1 is highlighted, transits to state 3.
3. State 3: When 'A' is pressed once again, option 2 in same list is highlighted, transits to state 4.
4. State 4: When 'B' is pressed while at the second option, transits to state 5.
5. State 5: When 'B' is pressed while at the first option (after pressing 'A' once), transits to state 6.
6. State 6: When 'B' is pressed while at the first option (after pressing 'A' once), transits to state 7.
7. State 7: When 'B' is pressed while at the first option (after pressing 'A' once), transits to state 8.
8. State 8: When 'A' is pressed once, then followed by a 'B', transits to state 9.
9. State 9: Pressing 'D' once would allow one to enter Hour value. Pressing 'D' twice would allow one to enter Minute value and pressing 'C' would save it and return control to stage 8.

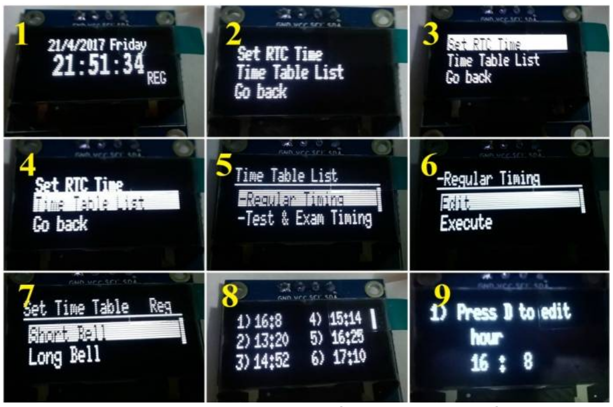

Fig. 5 Screenshots of the user interface

B. WIRELESS CONTROL

Another method to control the bell is via Bluetooth using an android app. The app was designed using MIT app inventor 2. For the moment, the Bluetooth app is only used to actuate the relay and in turn operate the bell. A 3.3v Red LED is connected between the state pin and ground of the Bluetooth module. The LED glows when Bluetooth is connected and turns off when disconnected. Refer Fig. 6.

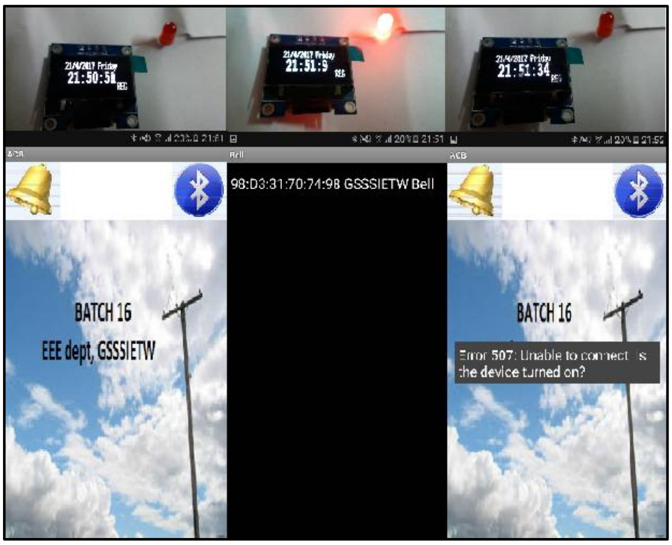

Fig. 6 Screenshots indicating LED glow on successful Bluetooth connection

VI. CONCLUSION AND FUTURE DEVELOPMENT

The project was reviewed by the teaching and non-teaching staff of EEE department of GSSSIETW and thorough testing was carried out to check if there were any fall backs. The navigation method and UI were well appreciated as it allowed the operator to navigate around the system in a hassle free manner with less dependency on user instructions. In future, we look forward to add a calendar feature so that timetables can be pre-scheduled. Also, we look forward to interface a larger display screen instead of the 0.96" OLED display.

REFERENCES

[1] n.p. Arduino. Arduino, n.d. Web. Accessed on 12-01-2017. https://www.arduino.cc
[2] Anand Nayyar and Er. Vikram Puri, "A Review Of Arduino's Board's, Lilypad's And Arduino Shields", International Conference on Computing for Sustainable Global Development (INDIACom), Electronic ISBN: 978-9-3805-4421-2, Pages 1485–1492, 2016.
[3] Maxim Integrated Products Inc.,"DS3231 Extremely Accurate I2C-Integrated RTC/TCXO/Crystal" DS3231 Datasheet, 2015.
[4] iteadstudio.com. "HC-05 Bluetooth module" Datasheet, Accessed on 06.18.2010.
[5] Zx Lee, 4x4 matrix keypad, Personal blog on Electronics and Programming, 24-07-2013. Web. Accessed on 01-02-2017. https://iamzxlee.wordpress.com/2013/07/24/4x4-matrix-keypad/
[6] Atmel Corporation, "Two wire automotive serial EEPROM" AT24C08A/16A Datasheet, 2004.
[7] Fried, Limor. "Monochrome OLED Breakouts." Adafruit. n.d. Web. Accessed on 30-01-2017. https://learn.adafruit.com/monochrome-oled-breakouts
[8] Abyash Gautam, Deepak Rasaily and Sejal Dahal, "Microcontroller Controlled Automated College Bell", International Journal of Engineering Trends and Technology (IJETT), Volume 32, Issue 4, Pages 184-187, February 2016.
[9] Sheenu Choudhary, Shrikant and Priyanka Sharma, "Automatic college bell system", International Journal of scientific research and management (IJSRM), Volume 2, Issue 3,Pages 668-673, 2014.
[10] Shweta Butoliya, Nupur Shal, Snehal Girhepunje, Aakanksha Rannaware, Prerna Baddalwar and Vaishali Badwe, "Microcontroller based automatic college bell with monitoring system", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 4, Issue 2, Pages 401-404, February 2015.