IMPLEMENTATION OF AUTOMATIC COLLEGE BELL SYSTEM USING ARDUINO

Burgoji Santhosh Kumar
Assistant Professor, Dept Of ECE, Anurag Group Of Institutions, Ts, India.

Abstract

The main objective of this project is to implement an Automatic college bell system using an Arduino. In this Arduino based college bell circuit, we used three major components which are IC RTCDS1307, Arduino Uno Board, and 16x2 LCD modules. Here, Arduino is used for reading time from ds1307 and display it on 16x2 LCD. DS1307 sends time/date using 2 lines to arduino. A buzzer is also used for alarm indication, which beeps when alarm is activated, simultaneously voice module will read out the data displayed on the LCD module. Here, the code is designed in a way so the bell will be activated for every 50 minutes as per the college schedule, the bell will be ringing for 50 seconds from the instant it is activated and simultaneously LCD screen will display the completion of a particular session, which indicates the students and faculty about the completion of a particular session and simultaneously the voice module will also read out the data which is being displayed on the LCD screen.

Keywords: Aurdino, Atmega 328, Microcontroller based bell system

1. Introduction

In today's life, everyone gives importance to time. Time does not wait for anybody. Everything should be performed in time & accurately. Now a day's school or college bells are manually operated. Hence there is a big question of accuracy. Also there is necessity of manpower and money. Hence here we should use automatic control system, which saves our manpower and money & also highest accuracy. [1]. In this project the scope is to design a AUTOMATIC COLLEGE BELL and its implementation on ARDUINO UNO BOARD. An AUTOMATIC COLLEGE BELL is a digital circuit that is used for the purpose of automatic switching of bell as per the given schedule without any human intervention. Generally, wherever we may go, it might be a school or an organization if start or stop of any process is to be conveyed to a large number of people, a bell is used over there which signals the start or stop of any process. So, all these bells are generally operated by the humans directly which is not advisable always as it is not efficient and even accuracy of the time is also being changed. So, in order to avoid this automation based bell system is to be introduced. [2]. Automation or automatic control, is the use of various control systems for operating equipment such as machinery, processes in factories, boilers and heat treating ovens, switching on telephone networks, steering and stabilization of ships, aircraft and other applications with minimal or reduced human intervention. Some processes have been completely automated. The biggest benefit of automation is that it saves labour; however, it is also used to save energy and materials and to improve quality, accuracy and precision.[3]

2. Existing system

In market there many digital clocks available with bells but rings only at specific time. For e.g. Alarm Clock and some bells that ring after some time intervals and that cannot stop after specific time. For e.g. Musical Clock But all these limitation have been removed by our project. It rings only according to our college time table

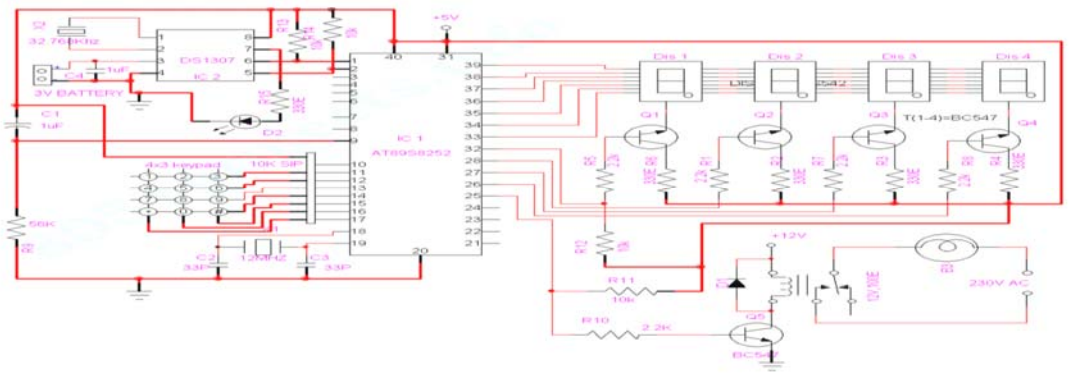


Fig1.MICROCONTROLLER BASED BELL SYSTEM

This Project takes over the task of Ringing of the Bell in Colleges. It replaces the Manual Switching of the Bell in the College. It has an Inbuilt Real Time Clock (DS1307 /DS 12c887) which tracks over the Real Time. When this time equals to the Bell Ringing time, then the Relay for the Bell is switched on. The Bell Ringing time can be edited at any Time, so that it can be used at Normal Class Timings as well as Exam Times. The Real Time Clock is displayed on LCD display. The Microcontroller AT89S52 is used to control all the Functions, it get the time through the keypad and store it in its Memory. And when the Real time and Bell time get equal then the Bell is switched on for a predetermined time.

3. Proposed Method

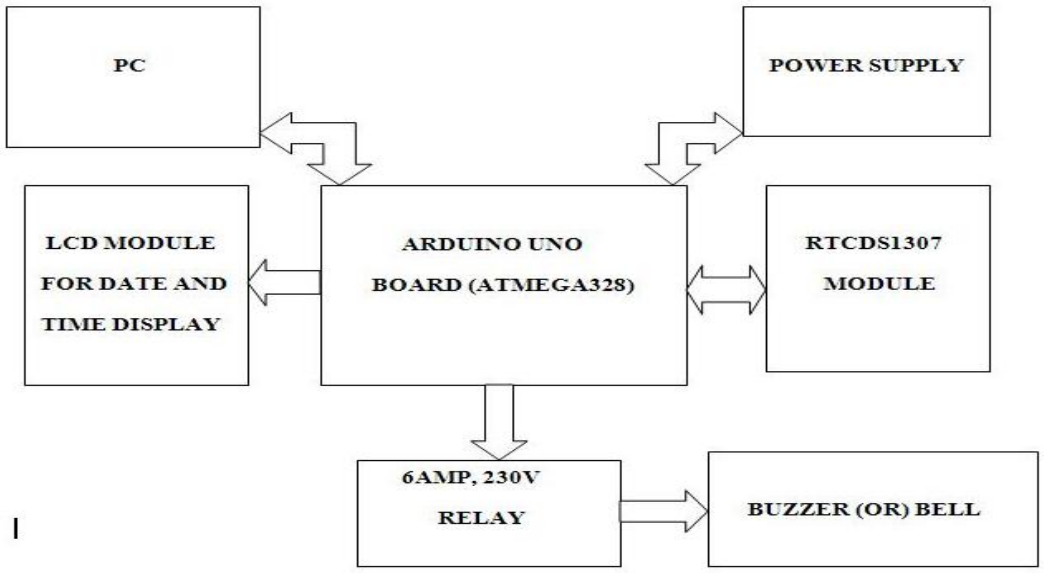


Fig 2. BLOCK DIAGRAM OF ARDUINO BASED BELL SYSTEM

Operation: From the above block diagram we can make out the circuitry involved in the design of an automatic college bell. Here, we are making use of ARDUINO UNO board for dumping the code written in ARDUINO IDE 1.6.7 software using Python coding and then we can check the required output of bell by interfacing it to the ARDUINO UNO board. The heart of the circuit is the ATMEGA microcontroller. The microcontroller we have used is ATMEGA328 which is the master device.

The slave device is the RTC IC DS1307, which automatically counts every second, once enabled. The intervals of time after which the bell should ring is already programmed and loaded into the microcontroller. Once the time that is fixed matches with the time in the RTC clock, the bell rings. The bell rings continuously for a fixed time (50 seconds in our implementation) which is also mentioned at the time of programming. The circuit is implemented by interface the DS1307 with the ATMEGA328 microcontroller. It is through this serial interface that the exact time is read into the ATMEGA328 microcontroller and is compared against the set of time in the code. If the present time matches with the time that is set in the program, that is when the bell should ring, logic HIGH is driven to the output port of the microcontroller. This small voltage (5V) acts as the enable to the relay circuit, which turns on the 230V to the bell and the bell rings. Another part of the system is the time display. The time value read into the microcontroller from RTC is also given as output through its port pins every instant to be displayed, along with comparing the values internally. The output value from the microcontroller pins are displayed in the 16X2 LCD display, which gets automatically updated every minute. In the application of the automatic bell that we have used, the microcontroller is configured as the master device. Microcontroller serially communicates with the RTC (DS1307), which is the slave device.

FEATURES OF ARDUINO:

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

4. FLOW CHART:

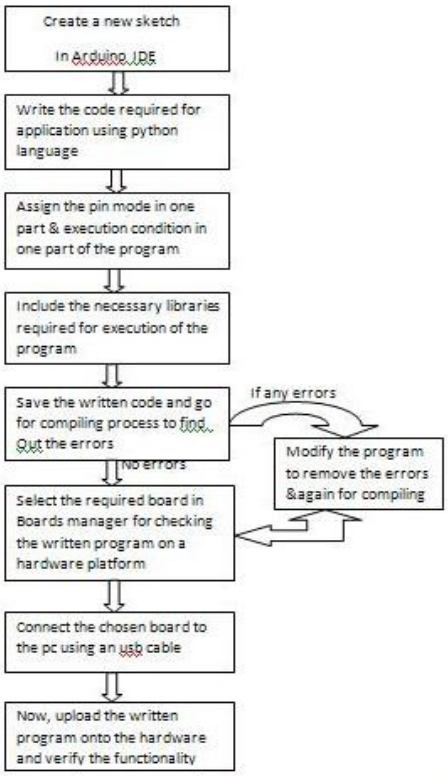
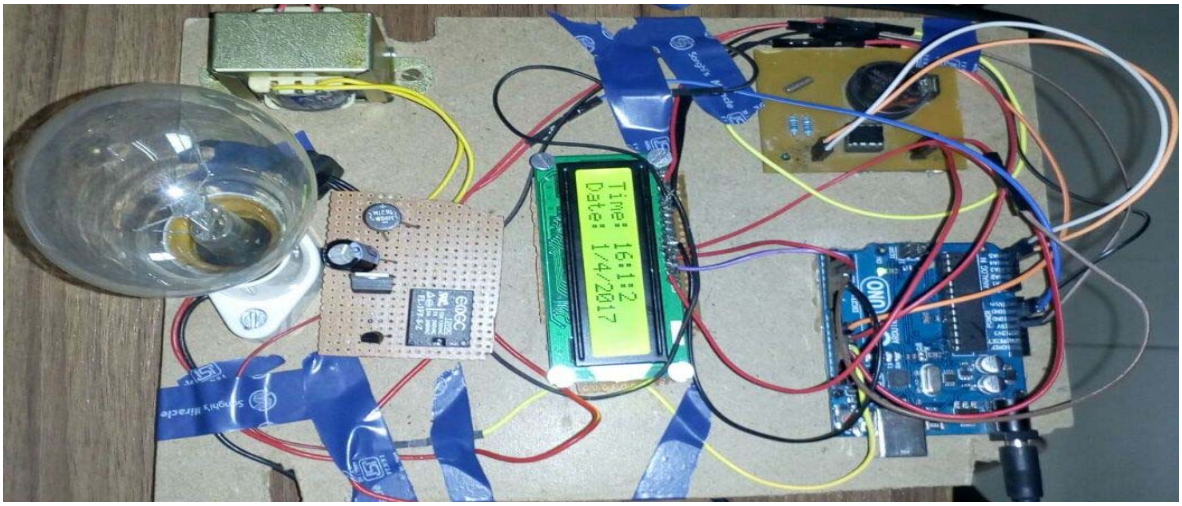


Fig 3. Flow chart

5. Results and discussions:

So, coming to the results, The program have been written in such a fashion that, the bell should ring simultaneously for every 50 minutes along with the display of the date and time on the lcd screen indicating the completion of a particular session and beginning of another session exactly at that instant of time at which the bell rings continuously for 50 seconds from the movement it is activated. Here, in our implementation, the bell rings at 9 instants of time in a day's schedule which is according to the program we have assigned.



The time instants at which the bell rings in a day's schedule is as follows:

TIME	DATA DISPLAYED ON LCD	STATUS OF THE BELL
8.45	DISPLAYS DATE AND TIME	OFF STATE (INACTIVE)
9:00	FIRST HOUR BEGINS	ACTIVATED (From 9:00:00 to 9:00:50)
9:49	DISPLAYS DATE AND TIME	INACTIVE
9:50	FIRST HOUR COMPLETED	ACTIVATED (From 9:50:00 to 9:50:50)
10:40	SECOND HOUR COMPLETED	ACTIVATED (From 10:40:00 to 10:40:50)
11:30	THIRD HOUR COMPLETED	ACTIVATED (From 11:30:00 to 11:30:50)
12:20	LUNCH HOUR STARTED	ACTIVATED (From 12:20:00 to 12:20:50)
13:10	LUNCH HOUR COMPLETED	ACTIVATED (From 13:10:00 to 13:10:50)
14:00	FIFTH HOUR COMPLETED	ACTIVATED (From 14:00:00 to 14:00:50)
14:50	SIXTH HOUR COMPLETED	ACTIVATED (From 14:50:00 to 14:50:50)
15:40	END FOR TODAY	ACTIVATED (From 15:40:00 to 15:40:50)

Conclusion and future scope

The whole point of the "Arduino Platform" is to allow for easy and fast prototyping. Being able to just hook up an LCD and be able to display messages on it in a matter of minutes, instead of hours, is just amazingly powerful and convenient when you have an idea in your head and just want to see if it works. When you need more control and are actually thinking on converting your prototype into a real product, then yes, you need to get deep down into the microcontroller and get rid of all the excess fat, trim the circuit to just the bare bones, optimize the code, etc. For prototyping, the Arduino platform gives you a lot of pre-wiring and free code libraries that will let you concentrate on testing your idea instead of spending your time building supporting circuitry or writing tons of low level code.

Using an Arduino simplifies the amount of Hardware and software development you need to do in order to get a system running. On the software side, Arduino provides a number of libraries to make programming the microcontroller easier. The simplest of these are functions to control and read the I/O pins rather than having to fiddle with the bus/bit masks normally used to interface with the Atmega I/O (This is a fairly minor inconvenience). More useful are things such as being able to set I/O pins to PWM at a certain duty cycle using a single command or doing Serial communication. Personally, I think the greatest advantage is having the hardware platform set up already, especially the fact that it allows programming and serial communication over USB. This saves me the trouble of having to do my own PCB (which can cost more than an Arduino) or bread boarding (which most people won't like doing).

REFERENCES:

[1]. Naresh, p.raveendra babu, k.satyaswathi, ijsetr "mobile phone signal jammer for gsm, CDMA with coverage prescheduled time duration using arm7" , volume2, issue september 2013

[2]. Ally s. Nyamawe, nixin mtonyole ijsetr, "the use of mobile phones in university exams cheating : proposed solution" volume 17 2014.

[3]. Alaparthi pardhasaradhi, rayalaravi kumar jsr, India online ISSN: 2319-7064 signal jamming and its modern applications.