A

PROJECT REPORT ON

"Automatic Bell System"

SUBMITTED TO

SHVIAJI UNVIERSITY, KOLHAPUR

SUBMITTED BY:-

1)Mr. Mali Vishvajeet Pandurang

2)Mr. Disale Nikhil Madhukar

3)Mr. Thombare Kunal Rakesh

UNDER THE GUIDANCE OF

Prof.Mrs. Manisha Thorat

Through

The Principal

KRISHNA COLLEGE OF COMMERCE AND SCIENCE,

WATHAR, (KARAD)

BCA-III (SEM-VI)

2022-2023

CERTIFICATE

This is to certify that Mr. Mali Vishvajeet Pandurang, Mr.Disale Nikhil Madhukar, Mr. Thombare Kunal Rakesh in a bonafied student studying in **BCA-III (SEM-VI)** his Project work of subject "**Automatic Bell System**" has been sincerely completed.

To my knowledge this project work is his original work.

DATE:-

PLACE:-WATHAR

.

PRINCIPAL

CERTIFICATE

This is to certify that Mr. Mali Vishvajeet Pandurang, Mr. DisaleNikhil Madhukar , Mr. Thombare Kunal Rakesh. Has satisfactorily completed their **VI** Semester. Project work entitled, **"Automatic Bell System " ,** under my guidance and supervision.

This work is being submitted for the award of the degree of **Bachelor of Computer Application** semester-**VI**, prescribed by Shivaji University, Kolhapur during the academic year **2022-2023**.

DATE:-

PLACE:-WATHAR

.

Project Guide
(Prof.Mrs. Manisha Thorat)

Head of Department
(Dr.Shripad Yelapure)

External Examiner

DECLARATION

To,

The principal,

Krishna College of Commerce and Science,

Wathar, Karad.

We hereby declare that this report is original in all aspects and the facts prescribed there in are true and not copied from any other project report. If proved otherwise. we will responsible for the consequences arising there from.

Sr.No	Project Members	Exam No	Signature
1)	Mr. Mali Vishvajeet Pandurang		
2)	Mr. Disale Nikhil Madhukar		
3)	Mr. Thombare Kunal Rakesh		

DATE:-

PLACE:-WATHAR

ACKNOWLEDGEMENT

This project report "**Automatic Bell System**" we tried to fulfill all the requirement related with the project. Many have contributed and helped us in completing the project. We thankfully acknowledge their contribution towards this project**.** **Bachelor of Computer Application**.

We are grateful to Prof. Mrs. Manisha Thorat Our guide for her valuable guidance, which she gave to use from time to time and bringing this project report in its most final from. We thanking to all our colleges for their moral support during the course of our project. We also thankful to each and every person, who is involved in this project their encourage & support being an important aspect of success of our project

Submitted By:-

1) Mr. Mali Vishvajeet Pandurang

2) Mr. Disale Nikhil Madhukar

3) Mr. Thombare Kunal Rakesh

INDEX

CHAPTER SCHEME

Chapter No - 1

INTRODUCTION TO SYSTEM

1) INTRODUCTION TO SYSTEM:-

- In any school, the classes are organized in periods and beginning of a period or break is alerted to the students and teachers by ringing the school bell. Conventionally, the school bell is rang by a peon or multi-tasking assistant. What if there would be a microcontroller based automatic school bell which rings itself according to a fed timetable. This project is the implementation of same functionality.

- The project is an Arduino based automatic bell system which can be configured for every class of the school. It is assumed that the school has five periods organized in a day for different subjects and have two breaks in between. The first break occurs after two periods and the next break occurs after the next two periods. After the fifth period, the school is over. The project allows to set duration for each period and assign subject from a list of subjects to each period. The user can set the time-table for six days of the week from Monday to Saturday. The user can also set the duration of both the periods.

- The Bell System using Arduino is a project that aims to design and implement an automated bell system using an Arduino microcontroller board. The system can be used in schools, colleges, or any institution where regular bell scheduling is required. This report presents an overview of the project, its objectives, methodology, implementation, and future enhancements.

Chapter No - 2

NEED FOR THE SYSTEM

2) NEED FOR THE SYSTEM:-

- With everything getting automated these days we think it's only fair to give good old school bell a touch of automation. So I thought to design and develop Automatic school bell system using Arduino development board. This system activates the School bell at predefined time each day. Add this project comes with a manual switch to activate the bell. This will give the liberty to the user to activate it manually when required.

Chapter No - 3

EXISTING SYSTEM

3) EXISTING SYSTEM: -

-The existing system is a manual system. The proposed system tries to simplify the difficulties encountered in manually handling information about the time setting, bell setting using push buttons.

-The existing system requires a number of records and takes a huge amount of time for the process completion. When we compare the existing system with the proposed system, there are many drawbacks for the 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

- 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.

-The existing system has limitations.

- The existing system is time consuming.
- Less reliable due to human errors.
- Fast processing speed.

-Current system does not provide any security.

- Searching or modification of are record is very difficult.

Chapter No - 4

NEED AND SCOPE

4) NEED AND SCOPE:-

- Bell system encompass efficient time management, flexible scheduling, increased productivity, enhanced safety measures, integration with other systems, remote management capabilities, and customization options. Implementing such a system can greatly streamline operations and improve overall functionality in various settings..

NEED:-

i) Time management: Precise timekeeping by automatically ringing bells at specific intervals.

ii) Scheduling flexibility: Ability to create and customize bell schedules to accommodate different events or activities.

iii) Efficiency and productivity: Elimination of manual bell operation, saving time and effort for personnel.

iv) Emergency notifications: Ability to sound specific alarm tones or signals for emergency situations like fire drills or lockdowns.

v) Integration with other systems: Capability to integrate with public address (PA) systems or intercoms for announcements and messages.

vi) Remote management: Ability to control the bell system from anywhere using web-based interfaces or dedicated software.

vii) Customization: Option to choose different bell tones or chimes for personalization and unique identity.

SCOPE: -

i) Schools: Automatic bell systems can streamline class schedules, breaks, and other school activities, enhancing time management and productivity.

ii) Workplaces: Efficiently mark shift changes, breaks, and meetings, improving workflow and employee punctuality.

iii) Public spaces: Use automated bells to signal opening and closing times, events, or public announcements in places like libraries, community centers, or parks.

iv) Emergency situations: Instantly activate alarm tones for fire drills, lockdowns, or other emergency procedures, ensuring a prompt response.

v) Large campuses or organizations: Manage multiple buildings or zones from a centralized system, making it easy to synchronize bells across the entire facility.

vi) Safety and security: Enhance safety measures by incorporating the automatic bell system into emergency response protocols, such as evacuation alerts or severe weather warnings.

vii) Scalability: Expand the system to accommodate additional locations or zones without significant hardware changes, making it suitable for growing organizations.

Chapter No - 5

PROPOSED SYSTEM

5.1) PROPOSED SYSTEM: -

i) Control Unit: The central control unit serves as the brain of the system. It manages the scheduling, timing, and activation of the bell signals. The control unit is typically a programmable device that allows users to set up custom schedules and manage various settings.

ii) Bell/Chime Outputs: The system includes multiple bell or chime outputs that are connected to the control unit. These outputs are responsible for generating the audible signals at specified times. The number of outputs can vary based on the size and complexity of the facility.

iii) Scheduling Interface: The system should provide a user-friendly interface, either through a physical control panel or a web-based software application. This interface allows authorized personnel to set up and customize bell schedules, define specific events or activities, and make any necessary adjustments to the system.

iv) Pre-recorded Tones/Chimes: The system may offer a library of pre-recorded bell tones or chimes to choose from. Users can select the appropriate sound for different events or activities, ensuring a distinct auditory signal.

v) Emergency Notification Integration: The system can integrate with emergency notification systems, such as PA systems or intercoms. This integration allows the automatic bell system to sound specific alarm tones or signals in emergency situations, providing immediate notifications to occupants.

vi) Remote Access and Management: Many modern automatic bell systems offer remote access and management capabilities. Authorized users can control and manage the system remotely using web-based interfaces or dedicated software, allowing them to make changes or monitor the system from any location with internet access.

vii) Expansion and Scalability: The system should be designed to accommodate future expansion or scalability. It should support the addition of extra bell outputs or zones as needed, allowing for the integration of multiple buildings or locations within a facility.

viii) Logging and Reporting: The system can log the bell events, providing a record of when each bell signal was activated. This feature enables administrators to review and analyze the system's usage and make any necessary adjustments.

ix) Notifications and Alerts: The system may have the capability to send notifications or alerts to authorized users. These notifications can be sent via email, SMS, or other means, providing important updates or reminders

5.2) OBJECTIVES: -

i) To design system which automate the ringing of bells at predetermined intervals using an Arduino-based system.

ii) To design system which simplify the scheduling and management of bell ringing events using an Arduino-based automation solution.

iii) To enhance efficiency and accuracy in bell ringing operations by implementing an Arduino-driven automatic bell system.

iv) To design system which eliminate the need for manual bell ringing by implementing an Arduino-based automated bell system.

Chapter No - 6

BLOCK

DIAGRAM

6) BLOCK DIAGRAM: -

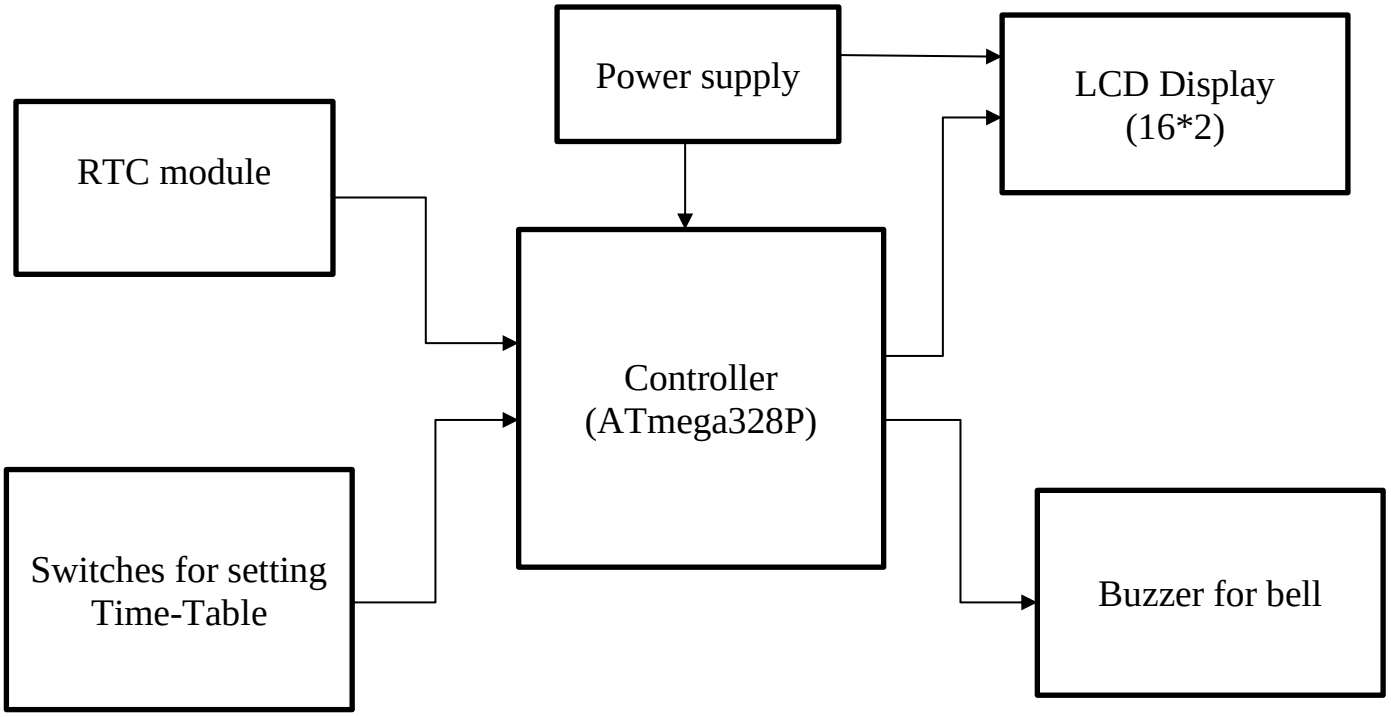


Fig- Block Diagram

Chapter No - 7

OPERATION

7) OPERATION: -

i) Setup and Configuration: Initially, the system needs to be installed and configured according to the specific requirements of the facility. This involves connecting the bell outputs to the control unit, configuring the scheduling interface, setting up any pre-recorded tones or chimes, and integrating with other systems if necessary.

ii) Scheduling: Once the system is set up, authorized personnel can access the scheduling interface, either through a physical control panel or a web-based software application. They can define the bell schedules by specifying the timing and duration of each event or activity. This includes setting the start and end times for classes, breaks, shift changes, or any other events that require a bell signal.

iii) Customization: The system may offer customization options, allowing users to select the desired bell tones or chimes for each event or activity. Users can choose from a library of pre-recorded sounds or upload their own if supported by the system. This customization allows for a unique auditory signal for different events.

iv) Activation and Deactivation: Once the schedules are set, the automatic bell system will activate and deactivate the bell outputs based on the defined timing. At the scheduled time, the system will trigger the corresponding bell output, generating the selected tone or chime. This process repeats according to the programmed schedule throughout the day.

v) Emergency Notifications: In emergency situations, the system can integrate with emergency notification systems such as PA systems or intercoms. When an emergency signal is activated, the automatic bell system will sound specific alarm tones or signals to alert occupants and provide instructions. This helps ensure the safety and security of everyone within the facility.

vi) Remote Management: If the system supports remote access and management, authorized users can make changes or adjustments to the bell schedules and settings from any location with internet access. This allows for convenient management and flexibility, especially in large campuses or organizations with multiple buildings.

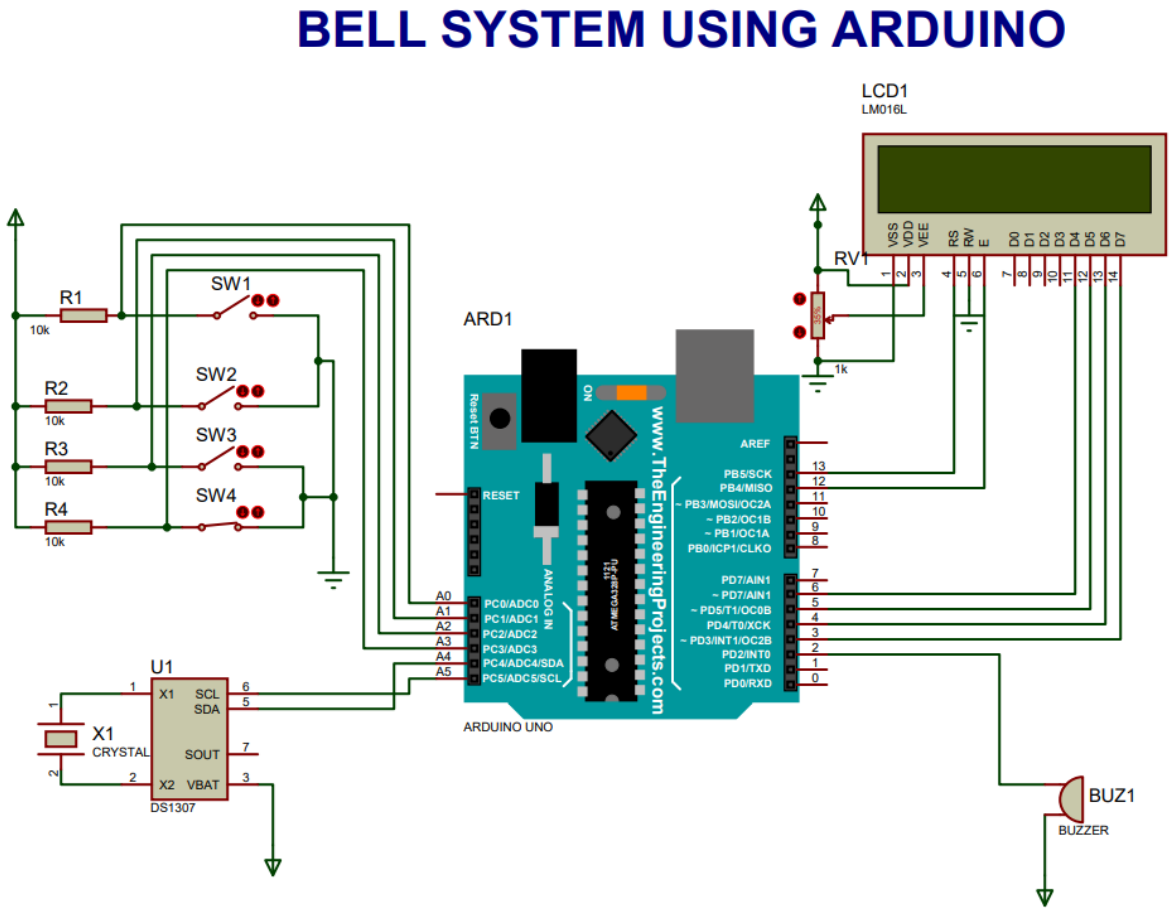
vii) Monitoring and Reporting: The system may provide monitoring and reporting capabilities, allowing administrators to track the system's performance and usage. This includes logging the activation of each bell event, generating reports on system activity, and identifying any potential issues or inconsistencies.

Chapter No - 8

CIRCUIT

DIAGRAM

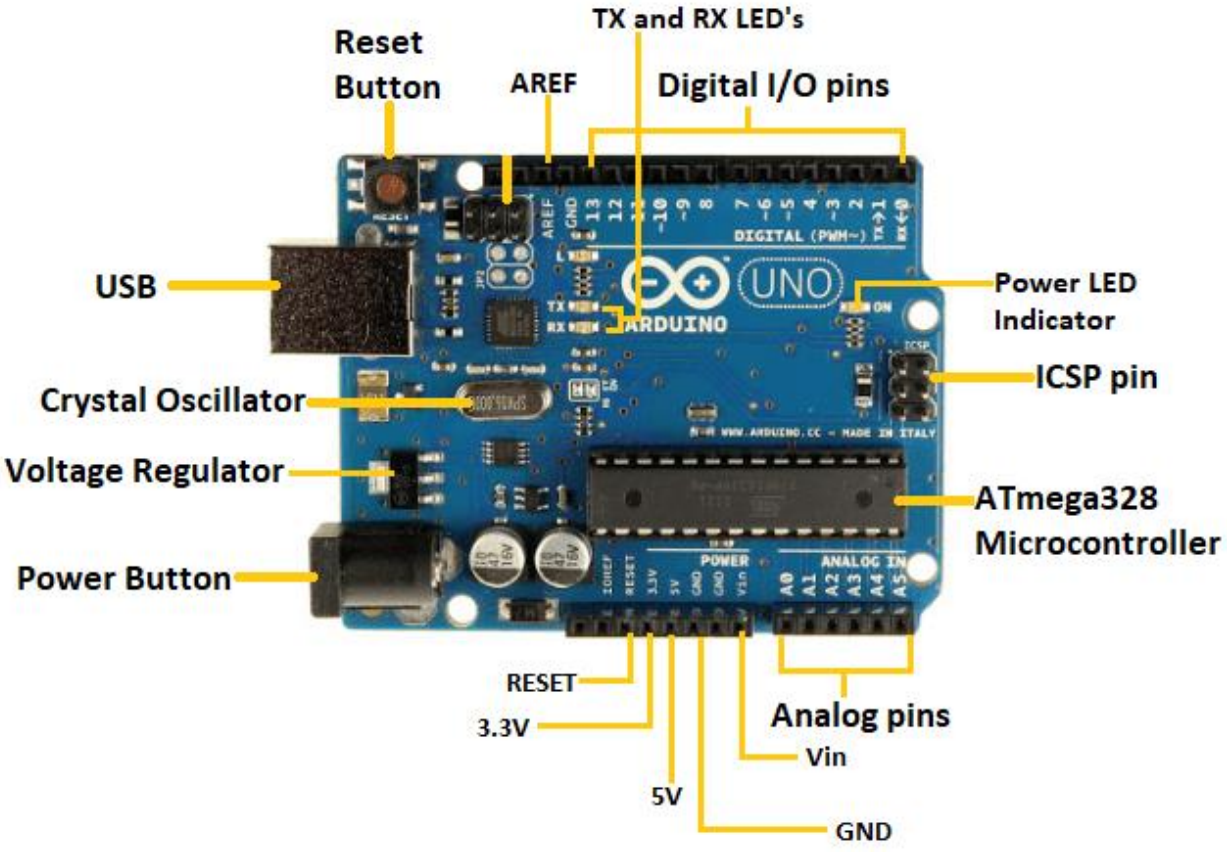
8) CIRCUIT DIAGRAM: -



Chapter No - 9

REQUIREMENTS

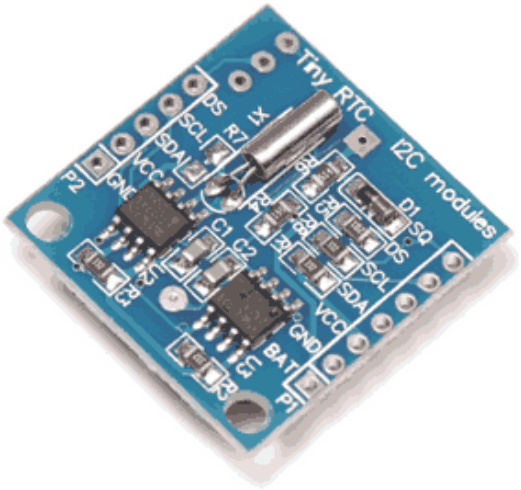
9.1) ARDUINO UNO BOARD:-

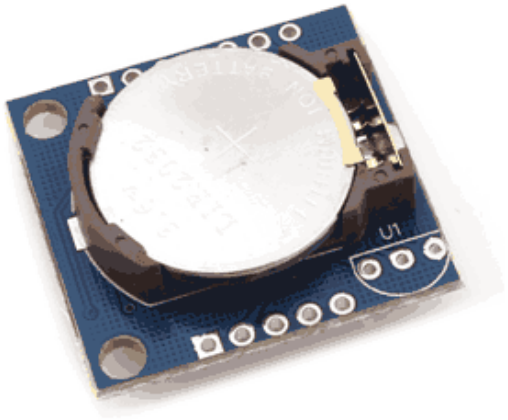


The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino The board is equipped with sets of digital and analogue input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analogue I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is similar to the Arduino Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

Microcontroller	ATmega328P
Operating Volt.	5V
Input Volt. (recommended)	7-12V
Input Volt. (limit)	6-20V
Digital I/O Pins	14 (6 PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm

9.2) TINY RTC MODULE (DS1307):-

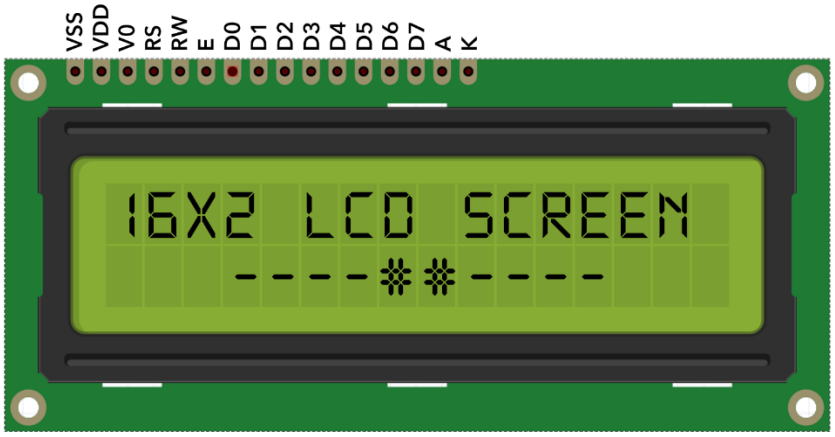




- The **DS1307** is a low power Full Binary (BCD) Real Time Clock (RTC) IC with 56 bytes of SVRAM that communicates through I2C Protocol. The IC can work from directly supply on Vcc and switch to Battery automatically when required.

Pin Number	Pin Name	Description
1,2	X1 , X2	Crystal Oscillator should be connected to these pins
3	V-Bat	Connected to Positive terminal of the battery
4	Ground	Ground pin of the IC
5,6	SCL and SDA	Pins for I2C communication with CPU
7	SQW / Out	Square wave output driver pin to obtain square wave frequencies.
8	Vcc	Powers the IC typically 5V

9.3) 16*2 LCD DISPLAY:-



No	Symbol	Function
1	VSS	Ground
2	VDD	5V +
3	V0	Contrast
4	RS	Register
5	RW	Read/Write
6	E	Enable
7	D0	Data bus
8	D1	Data bus
9	D2	Data bus
10	D3	Data bus
11	D4	Data bus
12	D5	Data bus
13	D6	Data bus
14	D7	Data bus
15	A	Anode (5V+)
16	K	Cathode (GND)

- Automatic Bell System, the 16x2 LCD display can be used to show the current time, bell schedule, system configuration options, error messages, and other relevant information. Users can navigate through the options and settings using buttons or a keypad connected to the system, while the LCD display acts as a visual output medium, providing real-time feedback and status updates.

9.4) PUSH BUTTONS:-

- A Push Button switch is a type of switch which consists of a simple electric mechanism or air switch mechanism to turn something on or off. Depending on model they could operate with momentary or latching action function. The button itself is usually constructed of a strong durable material such as metal or plastic.

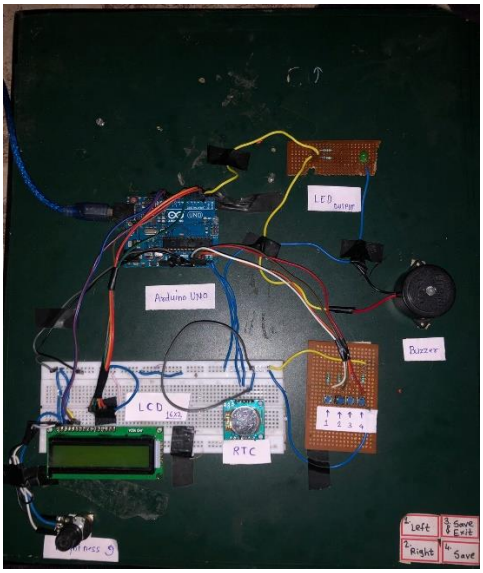
9.5) BUZZER:-

- A buzzer Typical uses of buzzers and beepers include alarm devices, timers, train and confirmation of user input such as a mouse click or keystroke.

9.6) POWER SUPPLY:-

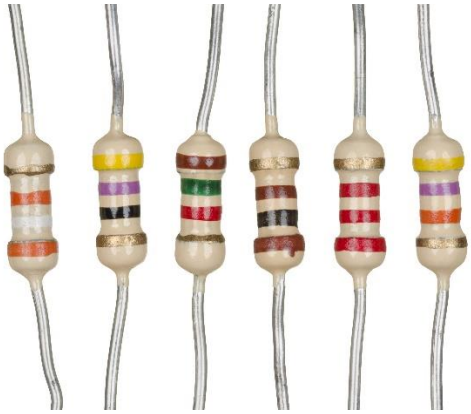
- Arduino board is by using its onboard USB connector. The USB connector provides a regulated 5V line to power the board's electronics. However, 5V from the USB connector can also power external components through the 5V pin that can be found in Arduino boards.

9.7) CONNECTING WIRES:-



- Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed.

9.8) RESISTORS:-



- The four band color code is the most common variation. These resistors have two bands for the resistance value, one multiplier and one tolerance band. In the example shown here, the 4 bands are green, blue, red and gold.

Chapter No - 10

SOURCE CODE

10) SOURCE CODE (Arduino language):-

```
#include <Wire.h>

#include<EEPROM.h>

#include <RTClib.h> //library for RTC

#include <LiquidCrystal.h> //library for LCD

LiquidCrystal lcd(13, 12, 6, 5, 4, 3);//  initialize the library with the numbers of the
interface pins ,Pins used for RS,E,D4,D5,D6,D7

RTC_DS1307 rtc;

char* mySubject[]={"MATHS", "ENG","BIO","PHY","CHEM","IT
LAB","HIST","GEO"};

char daysOfTheWeek[7][12] = {"SUN", "MON", "TUE", "WED", "THU", "FRI",
"SAT"};

int subject[7];

int buzzer=2;//Pin declare for buzzer control at pin number 24 of arduino mega

int b1 = A0;     // the number of the pushbutton pin

int b2 = A1;      // the number of the LED pin

int b3 = A2;      // the number of the LED pin

int b4 = A3;      // the number of the LED pin
int bS1 = 0;
```

```
int bS2 = 0;
int bS3 = 0;

int bS4 = 0;

int newTime,newHour,prevTime=0,prevHour=0,a=0,hoursE=0,hoursH=0, countTime=60;

int
i=0,j=0,k=0,var,nextSUB=0,value,pos=0,periodtime=0,starttime=0,endtime=0,totaltime=0
,break1=0,break2=0;

int monA=6,tueA=12,wedA=17,thuA=22,friA=27,satA=32,brk1,brk2,noPeriod=0;

int HOUR,MINUT,SECOND,timeH,timeM,setT=0,breakR=0;

byte Aalpha[8] ={ 0b00000,
                  0b01110,
                  0b10001,
                  0b11111,
                  0b10001,
                  0b10001,
                  0b00000,
                  0b00000
                  };

 byte Balpha[8]={ 0b00000,
```

```
        0b11100,
        0b10010,
        0b11100,
        0b10010,
        0b11100,
        0b00000,
        0b00000};
byte Calpha[8]={  0b00000,
        0b01110,
        0b10000,
        0b10000,
        0b10000,
        0b01110,
        0b00000,
        0b00000};
byte Dalpha[8] ={ 0b00000,
        0b11100,
```

```
        0b10010,

        0b10010,

        0b10010,

        0b11100,

        0b00000,

        0b00000};

byte Ealpha[8] ={0b00000,

        0b11110,

        0b10000,

        0b11100,

        0b10000,

        0b11110,

        0b00000,

        0b00000};

byte Falpha[8] ={0b00000,

        0b11110,

        0b10000,

        0b11100,
```

```
        0b10000,
        0b10000,
        0b00000,
        0b00000};
byte Galpha[8]={0b00000,
        0b01110,
        0b10000,
        0b10110,
        0b10010,
        0b01110,
        0b00000,
        0b00000};
byte Halpha[8] ={ 0b00000,
        0b10001,
        0b10001,
        0b11111,
        0b10001,
```

```
0b10001,

0b00000,

0b00000};

byte Ialpha[8]={0b00000,

0b01110,

0b00100,

0b00100,

0b00100,

0b01110,

0b00000,

0b00000};

byte Lalpha[8]={ 0b00000,

0b10000,

0b10000,

0b10000,

0b10000,

0b11110,

0b00000,
```

```
                0b00000};

byte Malpha[8] ={ 0b00000,

                0b10001,

                0b11011,

                0b10101,

                0b10001,

                0b10001,

                0b00000,

                0b00000};

byte Nalpha[8] ={0b00000,

                0b10001,

                0b11001,

                0b10101,

                0b10011,

                0b10001,

                0b00000,

                0b00000};
```

```
byte Oalpha[8]={ 0b00000,
                 0b01110,
                 0b10001,
                 0b10001,
                 0b10001,
                 0b01110,
                 0b00000,
                 0b00000};
byte Palpha[8]={  0b00000,
                  0b11100,
                 0b10010,
                 0b11100,
                 0b10000,
                 0b10000,
                 0b00000,
                 0b00000};
byte Ralpha[8]={    0b00000,
                 0b11110,
```

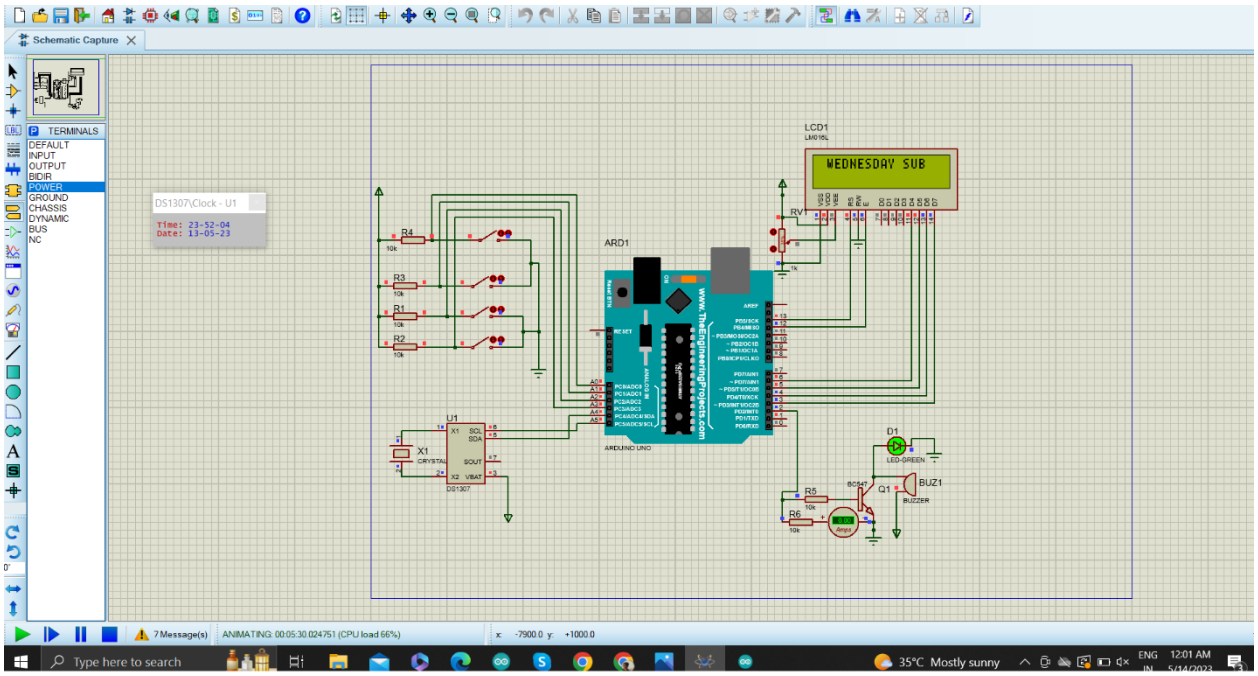
```
                0b10001,

                0b11110,

                0b10100,

                0b10010,

                0b00000,

                0b00000};

byte Salpha[8] ={ 0b00000,

                0b01110,

                0b10000,

                0b01110,

                0b00001,

                0b01110,

                0b00000,

                0b00000};

 byte Talpha[8] ={0b00000,

                 0b11111,

                0b00100,
```

```
        0b00100,

        0b00100,

        0b00100,

        0b00000,

        0b00000};



byte Ualpha[8]={  0b00000,

        0b10001,

        0b10001,

        0b10001,

        0b10001,

        0b01110,

        0b00000,

        0b00000};

byte Walpha[8] ={  0b00000,

        0b10001,
```

Chapter No - 11

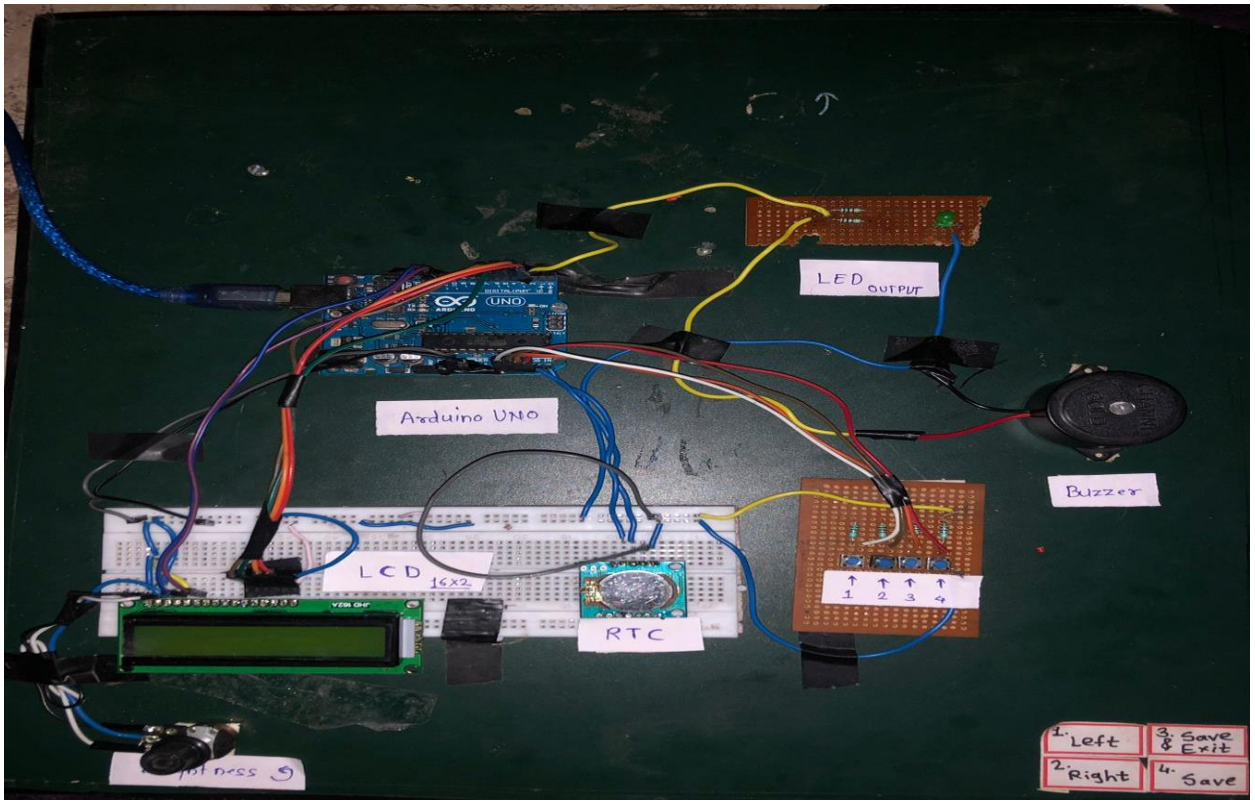
RESULTS

11.1) SIMULATION OUTPUT:-

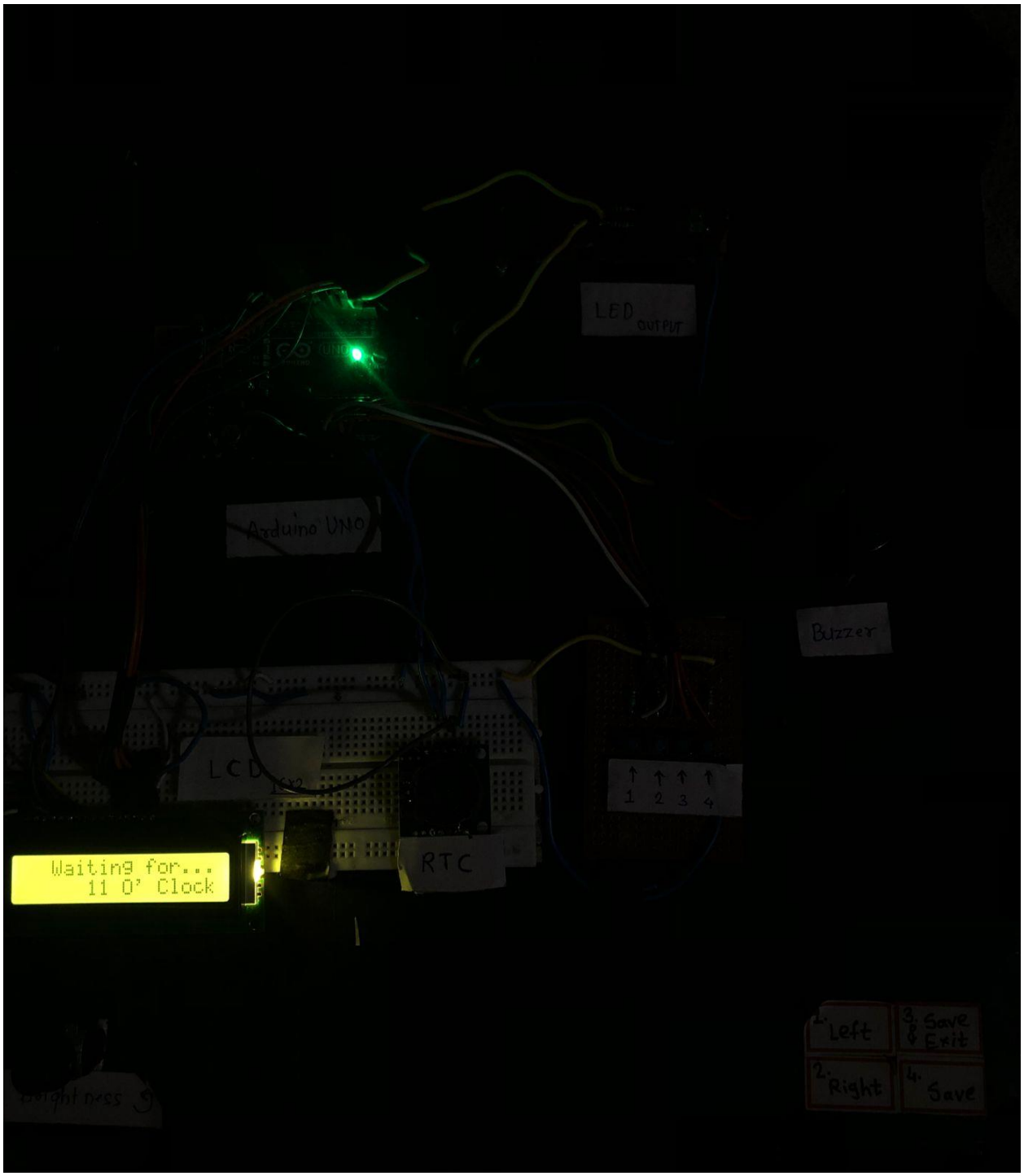


11.2.1) ACTUAL OUTPUT:-

i)



ii)


LED OUTPUT
Arduino UNO
Buzzer
LCD
1 2 3 4
Waiting for...
11 0' Clock
RTC
1. Left
3. Save & Exit
2. Right
4. Save

Chapter No - 12

CONCLUSION

12) CONCLUSION:-

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 andsoftware 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 At mega 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).

Chapter No - 13

REFERENCES

13.1) BOOKS:-

A) "Arduino: A Technical Reference" by J. M. Hughes.

B) "Programming Arduino: Getting Started with Sketches" by Simon Monk.

C) "Arduino for Beginners: Essential Skills Every Maker Needs" by John Baichtal.

D) "Exploring Arduino: Tools and Techniques for Engineering Wizardry" by Jeremy Blum.

13.2) WEB / BLOGS:-

A) www.arduino.com

B) www.wikipedia.com

C) www.google.com

D) www.github.com

E) www.youtube.com

F) www.opensource.com

STUDENT GUID MEET RECORD: -

Title of Project	**AUTOMATIC BELL SYSTEM**	**Class: BCA-III**
Student Names	**1)** Mali Vishvajeet Pandurang **2)** Disale Nikhil Madhukar **3)** Thombare Kunal Rakesh	**Guide Name:** Mrs.Manisha Thorat

Sr.	Date	Description	Signature of Guide	Signature of Student/s	Guide Remark
1		Problem Identification and Topic and title finalization (1^{st} week of semester)			
2		SRS submission and approval (3^{rd} week of semester)			
3		Logical Design of System (Block Diagram and Circuit Diagram) (5^{th} week of Semester)			
4		Simulation (7^{th} week of Semester)			
5		I/O Design (with Reports) (8^{th} of Semester)			
6		Submission of Draft Project Report (9^{th} Week of semester)			
7		Submission of Final Project Report (10^{th} Week of semester)			

HOD/ Director/Principal