VISVESVARAYA TECHNOLOGICAL UNIVERSITY
Jnanasangama, Macche, Santibastwada
RoadBelagavi-590018, Karnataka

A
BE MINI-PROJECT (19EC6DCMPR) REPORT
on

Solar powered Automatic Bell System

Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Engineering
in
Electronics & Communications Engineering - ECE
by

1DS20EC418 : Lohit Nimbagal
1DS20EC419 : Manoj M
1DS20EC426 : Pawan Kumar S
1DS20EC429 : Rahul M

Under the
guidance of
Dr. Ravindra S
Assistant Lect.
ECE Dept., DSCE, Bengaluru

Department of Electronics & Communication Engineering
(An Autonomous College affiliated to VTU Belgaum, accredited by NBA & NAAC, Ranked by NIRF)
Shavige Malleshwara Hills, Kumaraswamy
Layout,Bengaluru-560078, Karnataka, India

2021-22

Certificate

Certified that the mini-project work **(19EC6DCMPR)** entitled "Solar Powered Automatic bell system" carried out by **Lohit Nimbagal** (1DS20EC418), **Manoj M** (1DS20EC419), **Pawan Kumar S** (1DS20EC426), **Rahul M** (1DS20EC429) are bonafide students of the ECE Dept. of Dayananda Sagar College of Engineering, Bangalore, Karnataka, India in partial fulfillment for the award of Bachelor of Engineering in Electronics & Communication Engineering of the Visvesvaraya Technological University, Belagavi, Karnataka for the VI Semester course during the academic year 2021-22. It is certified that all corrections / suggestions indicated for the mini-project work have been incorporated in the mini-report submitted to the ECE department. This Mini-Project report has been approved as it satisfies the academic requirement in respect of mini-project work prescribed for the said degree.

Mini-Project Guide Sign : ______________________________
Name : **Dr. Ravindra S**

Mini-Project Section Coordinator Sign : ______________________________
Name : **Prof. Kavita S Guddad**

Mini-Project Convener & Chief Coordinator Sign : ______________________________
Name : **Dr. Shashi Raj K.**

Dr. T.C. Manjunath : ______________________________
HOD, ECE, DSCE

Dr. C.P.S. Prakash : ______________________________
Principal, DSCE

<u>External Mini-Project Viva-Voice (SEE)</u>

Name of the mini-project examiners with date :

1 : Signature : ____________________

2 : Signature : ____________________

Declaration

Certified that the mini-project work entitled, "Solar Powered Automatic Bell Sytem" with the course code **19EC6DCMPR** (100 Marks, CIE& SEE 50 marks each) is a bonafide work that was carried out by ourselves in partial fulfillment for the award of degree of Bachelor of Engineering in Electronics & Communication Engg. of the Visvesvaraya Technological University, Belagavi, Karnataka during the academic year 2021-22 for the VI Semester Autonomous Course. We, the students of the mini-project group/batch no.04 do hereby declare that the entire mini-project has been done on our own & we have not copied or duplicated any other's work. The results embedded in this mini-project report has not been submitted elsewhere for the award of any type of degree.

Student Name-1 : Mr. Lohit Nimbagal.
USN : 1DS120EC418

Sign : ________________

Student Name-2 : Mr. Manoj M.
USN : 1DS20EC419

Sign : ________________

Student Name-3 : Mr. Pawan Kumar S.
USN : 1DS20EC426

Sign : ________________

Student Name-4 : Mr. Rahul M.
USN : 1DS20EC429

Sign : ________________

Date : 10/08/2022
Place : Bengaluru -78

Acknowledgement

In this section, a brief acknowledgment has to be presented as a thanks giving to all the people who have directly or indirectly helped in the successful completionof the undergraduate mini-project work.

Thanks giving hierarchy to the members in this order:

1. Chairman – Dr. Hemachandra Sagar
2. Vice Chairman – Dr. Premachandra Sagar
3. Secretary – Galiswamy
4. Joint Secretary – Tintisha Sagar
5. Principal – Dr. C.P.S. Prakash
6. HOD – Dr. T.C. Manjunath
7. Mini Project Guide –Prof. Ravindra S
8. Mini-Project Section-wise Coordinators–Kumar P. (A) ; Kavitha S. Guddad (B) ; Vidyashree K.N. (C) ; Shobha A.S. (D)
9. Mini-Project Convener & Chief Coordinator – Dr. Shashi Raj K.
10. Dept. staff members – Teaching & Non-teaching
11. Family members & relatives
12. Friends
13. Almight

Abstract

The world over the decades has made considerable advancement in automation; it is employed in homes, industries, commercial and educational sectors. In present work, a solar power-operated microcontroller-based automatic college bell is designed and developed. For the harmonic tuning, converted normal college bell into automatic college bell and powered by solar PV system with battery backup. It uses electrical coil for generating the EMF for striking the clapper on the edge of a bell for making sound. It uses the Real Time Clock (DS1307) which tracks the real time. The Arduino UNO is used to control all the functions; it gets the time through the Switch Buttons and stores itin its memory. When programmed time equals the real time then the bell is switched on viaa relay for a predetermined time. The bell ringing time can be edited at any time so that it can be reused again and again at normal class timings as well as at exam times. The advantage of this design is that the bell rings at the start of each period without any human intervention to a great degree of accuracy and hence takes over the manual task of switching on/off the college bell with respect to time. It works on onetime time setup.

Table of Contents

List of Figures

Nomenclature and Acronyms

Abbreviations (Alphabetical Order):

IEEE	Institute of Electrical & Electronics Engineers
DSCE	Dayananda Sagar College of Engineering
ECE	Electronics & Communication Engineering

Chapter-1

Introduction

A bell is a percussion instrument used in schools or colleges that tells the students when it is time to go to class in the morning and when it is time to change classes during the day. No other instrument can do such work. So the bell is an important instrumentin schools, industries and other businesses offices where the bell timer plays a critical role in running the day. Because of in today's life, everyone gives importance to time and time does not wait for anybody. Everything should be performed in time and with accuracy. Now a day's schoolor college bells are operated manually andthey are using electrical bell. Hence there is a big question of accuracy and tuning of an electric bell is not like by most of the people. Also, there is a necessity of manpower and money. Hence here we should use automatic control system, which saves our manpower and money and also it gives the highest accuracy.

In this project, the scope is to designa mechanism for converting manual college bell into the automatic college bell and itsimplementation on Arduino Uno board. Anautomatic college bell is powered by solar PVsystem and 12 V battery connected for power backup in cloudy condition or absence of sunlight. An Automatic College Bell is a digital circuit that is used for the purpose of automatic switching of the bell as per the given schedule without any human intervention. The shape of the bell is hollow cup so that when struck vibrates in a single strong strike by internally connected clapper and creates a harmonic tune.

1.1 Overview

Usually, conventional methods require a peon or bell operator to attend and operate the bell system for every period and interval in schools and institutions. Such systems need sufficient human efforts to do so, and require advancement in order to become automated the ones that minimize human efforts. As the bell system is important in schools, homes, and industries, the automatic operation of this instrument must have to be performed with a precise time controller economically.

This automatic school bell timer system is designed using a Arduino UNO for managing time intervals. Read or Write memory is also necessary for storing bell timings, but for less number of timings this memory is not needed. This system also provides the display of timing information in the seven-segment display for a user interface purpose.

1.2 Literature survey

In early days schools and colleges had bell which were manually operated. Therefore, the were stared either early or late. Also, someone must be appointed to ring the bell every time. To overcome this drawback, we are using the automatic ringing bell. This bell will automatically ring for the time it has been set for. Binu, have presented a low-power, high-performance (8bit) AT89S52 microcontroller and EEPROM to store the alarm timings and a 7-segment display is used to display the timing. Ramesh Harajibhai Chaudhari, have presented a design methodology using Arduino microcontroller board which uses ATMEGA328 which has EEPROM of 1024 bytes (Electrically Erasable Programmable Read-Only Memory). This is a small space that can store byte variables. The variables stored in the EEPROM kept there even when the board is reset or powered off.

1.3 Objectives / Scope / Aim of the mini-project work

The main purpose of our project is to use renewable energy and accurate time keeping by replacing the manual effort.

1.4 Motivation & Problem Statement

In present system, bell for period or break are operated manually. After every class, an employee is engaged in ringing the bell. Human error, inaccuracy and inconsistency are the drawbacks of manual system. Hence there is a big question of accuracy. So there comes the need to have a system that can take proper and efficient time monitoring and notify scheduled class timing in the collage time table as saved on the system.

1.5 Existing & Proposed (Developed) Mini-Project module

The project is an Arduino UNO based automatic bell system which can be configured for every class of the Collage. It is assumed that the Collage 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 Collage is over. The project allows to set duration for each period. The user can set the time-table for six days of the week from Monday to Saturday.

The project device should be installed in individual classrooms and have the time table set manually. It has a 16X2 LCD display which keeps on displaying the present day of the week and current subject of the period on first row and date and time along with subject of the next period on its second row. The circuit has a buzzer which starts humming at the beginning and end of each period or break. The duration of each period and selection of the subject can be made through a four Push buttons on the circuit. The project utilizes an RTC module to keep track of real date and time. The project device is line powered which is regulated by a step-down transformer, full-bridge rectifier and 7805 and 7812 voltage regulator ICs.

1.6 Proposed Methodology

This model is designed with various inputs and outputs such as real-time clock (RTC), bell for ringing and LCDs for displaying the time. The Arduino UNO is used to program the output for each specified situation. In addition, this model is also added with solar cells and solar controller, which supplies power to Arduino and the bell. Thus, the process of Arduino will not be interrupted if conventional power supply from the main energy department is not available.

1.7 Organization of the mini-project report

This report is organized as follows:

- Chapter 1 & 2 describes about the objective, methodology, literature survey, problem statement from which we have referred the information which describes about which gives the idea of algorithm and new way of exploring the proposed system.
- Chapter 3 talks about the Hardware and software we have used to simulate.
- Chapter 4 shows us about the final results and discussions made.
- Chapter 5 tells us about the advantages, applications, limitations and outcomes.
- Chapter 6 describes conclusions.
- Chapter 7 Provides the references.

Chapter- 2

Block diagram, Circuit Diagram and Working principle:

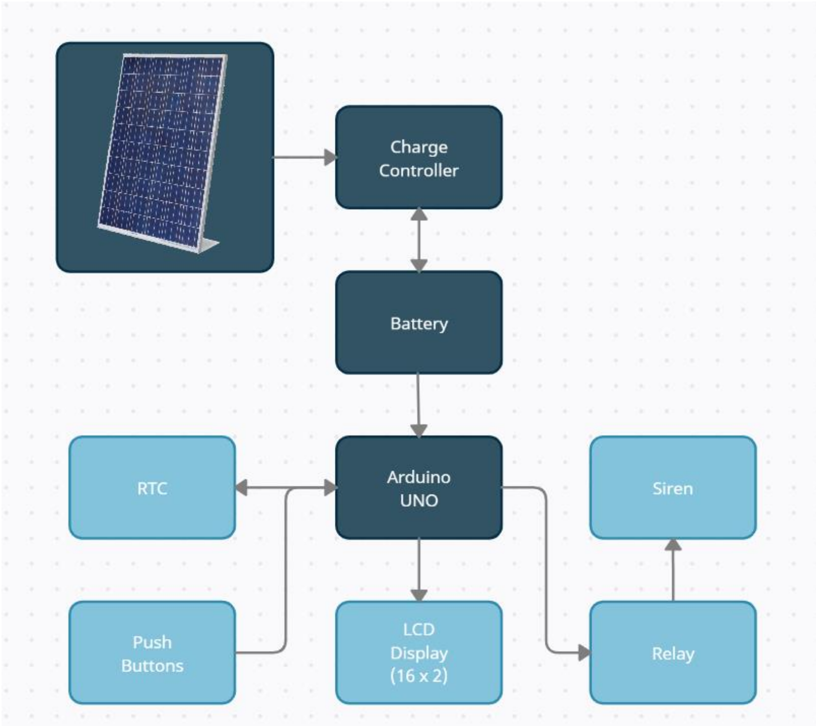

Fig.2.1: Block-diagram of the proposed methodology

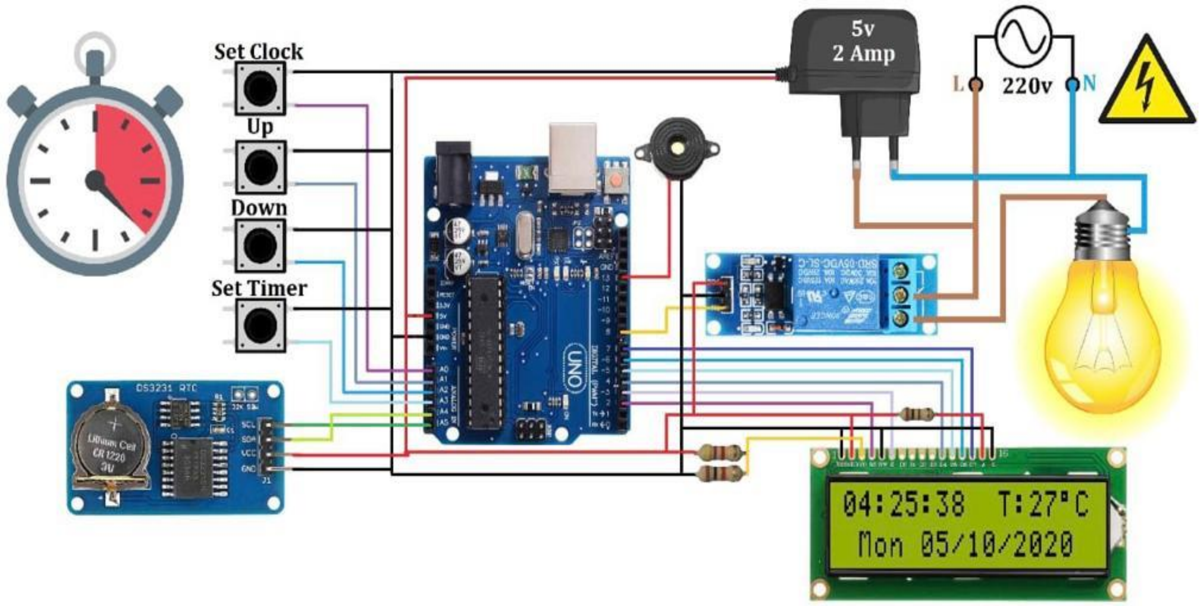

Fig.2.2: circuit connection of Automatic Collage bell system

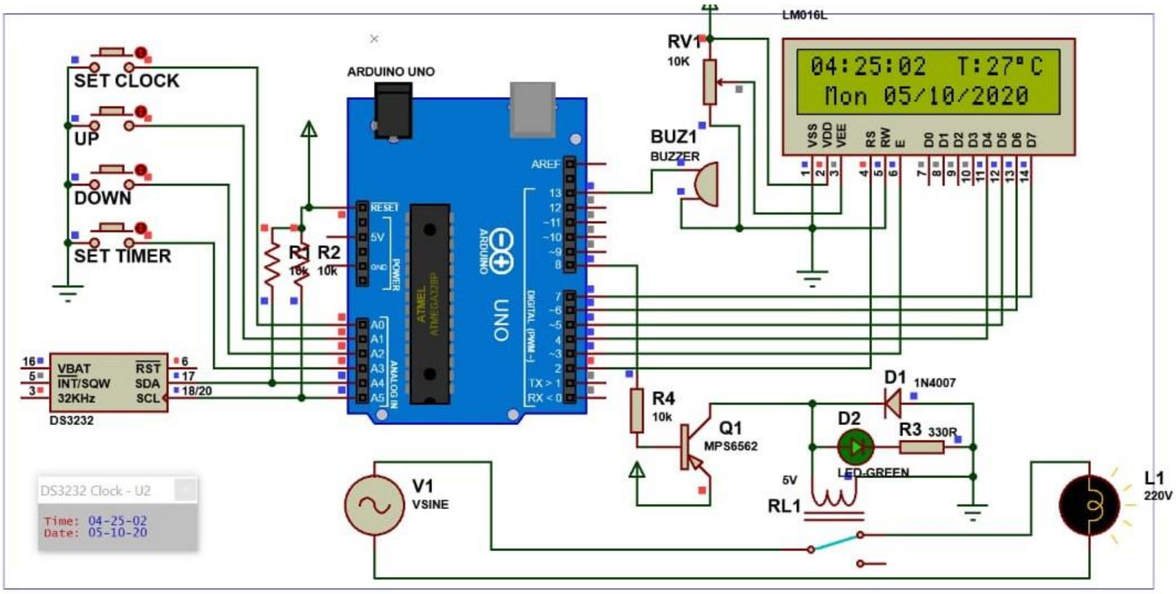

Fig.2.3: Circuit Diagram of Automatic collage bell system.

Working Principle:

This design takes the task of ringing the bell in college as the bell is ring automatically at the scheduled time. Arduino based systems have made lot of things automatically and easy to use. This is very wonderful design to control the working of college bell. 12 volt DC input is given to the Arduino and relay circuit for operation, 5 volt DC supply is given to the LCD display circuit and microcontroller circuit by the adapter from the charge controller. The Arduino Microcontroller is used to control all the Functions; it gets the different setup of time through the push buttons and stores it in its Memory. We used push buttons, so we can be resetting the alarms as per requirement and also can be save number of alarms for different time. So, it can be use at normal class timing as well as exam time. LCD is used for display the current time, temperature as well as different input settings. The Real Time Clock (DS1307) is used to provide real time which is displayed at the LCD screen. The control signals of relay and data signals of display, keypad is connected to the Arduino.

Real time clock tracks over the programmed alarm time when both time gets equalization then RTC send the instruction to the Arduino. Then Arduino Uno will send the power relieve command to the relay (Fig. 2 and 3). Then relay will supply the voltage to the coil as per operator need, for ringing duration at school for long bell or for short bell. Relay gives the power supply to the bell. After ringing duration microcontroller will discontinue the power supply.

Chapter-3

Hardware / Software tools / Description / Interfacing / Working of the complete mini-project module

Hardware:

The Hardware Components we used in our Project are:

- Solar Panel
- Solar Charge controller
- Lead Acid Battery
- Arduino UNO
- RTC Timer
- LCD Display
- Push Buttons
- Relay.
- Bell.

Software:

- Arduino IDE
- Proteus 8 Professional

3.1. Hardware Components Description:

3.1.1. Solar Panel

Fig.3.11. Solar Panel

Definition: Solar panels are those devices which are used to absorb the sun's rays and convert them into electricity or heat.

Description: A solar panel is actually a collection of solar (or photovoltaic) cells, which can be used to generate electricity through photovoltaic effect. These cells are arranged in a grid-like pattern on the surface of solar panels. Thus, it may also be described as a set of photovoltaic modules, mounted on a structure supporting it. A photovoltaic (PV) module is a packaged and connected assembly of 6×10 solar cells. When it comes to wear-and-tear, these panels are very hardy. Solar panels wear out extremely slow. In a year, their effectiveness decreases only about one to two per cent.
Most solar panels are made up using crystalline silicon solar cells. Installation of solar panels in homes helps in combating the harmful emissions of greenhouse gases and thus helps reduce global warming. Solar panels do not lead to any form of pollution and are clean. They also decrease our reliance on fossil fuels and traditional sources.
These days, solar panels are used in wide-ranging electronic equipment's like calculators, which work as long as sunlight is available. However, the only major drawback of solar panels is that they are quite costly. Also, solar panels are installed
outdoors as they need sunlight to get charged.

3.1.2. Solar Charge controller

Fig.3.1.2. Solar Charge Controller.

Charge controller is device which preventsbattery from overcharging and over discharging and also block reverse current flow from battery to panel during night; its power rating capacity is 12V and 6 Amp. It regulates the voltage and current coming from the solar panels going to the battery. Most "12 volt" panels put out about 16 to 20 volts, so if there is no regulation the batteries will be damaged from overcharging. Most batteries need around 14 to 14.5 volts to get fully charged. The obvious question then comes up - "why aren't panels just made to put out 12 volts". The reason is that if you do that, the panels will provide power only when cool, under perfect conditions, and full sun. This is not something you can count on in most places. The panels need to provide some extra voltage so that when the sun is low in the sky, or you have heavy haze, cloud cover, or high temperatures, you still get some output from the panel. A fully charged "12-volt" battery is around 12.7 volts at rest (around 13.6 to 14.4 under charge), so the panel has to put out at least that much under worst-case conditions.

3.1.3. Lead Acid Battery

Fig.3.1.3: Lead Acid Battery

A battery is a device consisting of one ormore electrochemical cells that convert stored chemical energy into electrical energy. Lead- acid battery of 12 V and 7 AH is used. It is the first type of rechargeable battery ever created. Compared to modern rechargeable batteries, lead–acid batteries have relatively low energy density.
Despite this, their ability to supply high surge currents means that the cells have a relatively large power-to-weight ratio. These features, along with their low cost, make them attractive for use in motor vehicles to provide the high current required by starter motors.

3.1.4. Arduino UNO

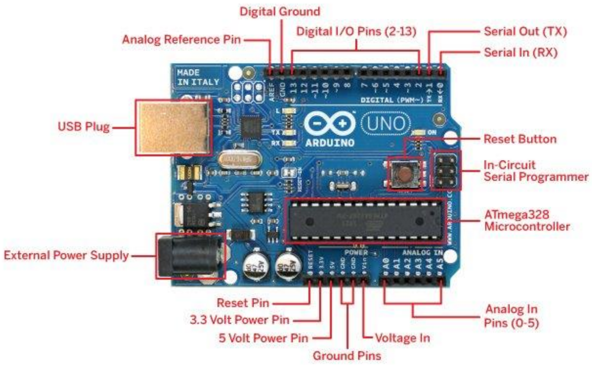

Fig.3.1.4: Arduino UNO

Arduino UNO is a microcontroller board based on the **ATmega328P**. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few rupees and start over again. It can be programmed to control the way buttons, motors, switches, lights, and other electronic parts work together. Arduino is hands-on, which is one of the many reasons it's appealing to kids. It can be plugged into your computer. You can write a program on the computer and send it to the Arduino.

Features of the Arduino UNO

- 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 of which 0.5 KB used by bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz

3.1.5. RTC Timer

Fig.3.1.5: RTC Timer.

We all know that most MCUs we use for our projects are time-agnostic; simply put they are unaware of the time around them. It's OK for most of our projects but once in a while when you come across an idea where keeping time is a prime concern, DS3231 Precision RTC module is a savvier. It's perfect for projects containing data-logging, clock-building, time stamping, timers and alarms.

DS3231 RTC chip

At the heart of the module is a low-cost, extremely accurate RTC chip from Maxim – **DS3231**.

It manages all timekeeping functions and features a simple two-wire I2C interface which can be easily interfaced with any microcontroller of your choice.

The chip maintains seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year (valid up to 2100).

The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. It also provides two programmable time-of-day alarms.

3.1.6. LCD Display

Fig.3.1.6. LCD Display

Nowadays, we always use the devices which are made up of LCDs such as CD players, DVD players, digital watches, computers, etc. These are commonly used in the screen industries to replace the utilization of CRTs. Cathode Ray Tubes use huge power when compared with LCDs, and CRTs heavier as well as bigger. These devices are thinner as well power consumption is extremely less. The LCD 16×2 working principle is, it blocks the light rather than dissipate. This article discusses an overview of LCD 16X2, pin configuration and its working.

The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi-segment light-emitting diodes and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

The 16×2 LCD pinout description is shown below.

- Pin1 (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.
- Pin2 (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
- Pin3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
- Pin4 (Register Select/Control Pin): This pin toggles among command or data register, used to connect a microcontroller unit pin and obtains either 0 or 1(0 = data mode, and 1 = command mode).
- Pin5 (Read/Write/Control Pin): This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation).
- Pin 6 (Enable/Control Pin): This pin should be held high to execute Read/Write process, and it is connected to the microcontroller unit & constantly held high.
- Pins 7-14 (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the microcontroller unit like 0 to 3, whereas in 8-wire mode, 8-pins are connected to microcontroller unit like 0 to 7.
- Pin15 (+ve pin of the LED): This pin is connected to +5V
- Pin 16 (-ve pin of the LED): This pin is connected to GND.

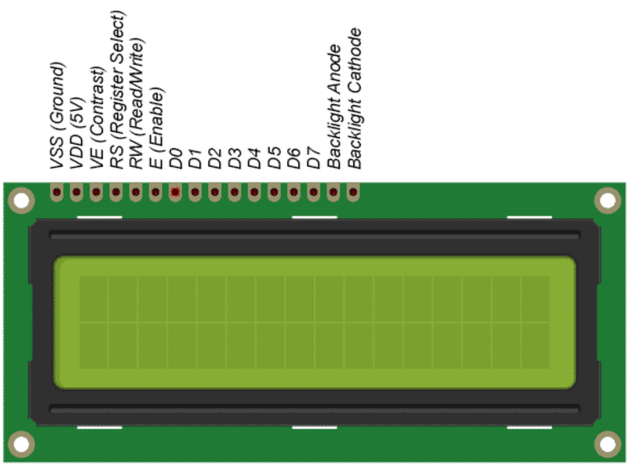

Fig.3.1.6.1. LCD 16X2 Pin Diagram

3.1.7. Relay Module

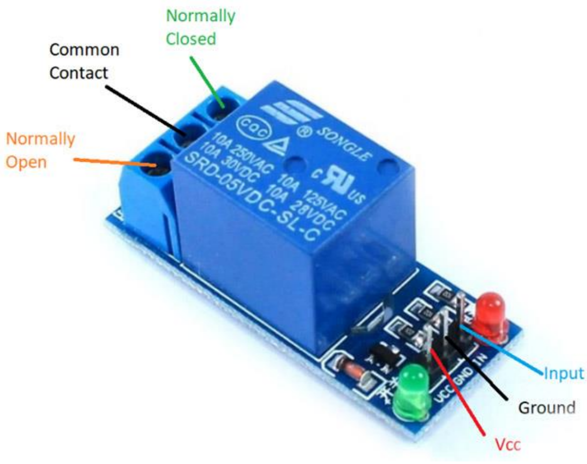

Fig.3.1.7. Relay Module.

A 5v relay is an automatic switch that is commonly used in an automatic control circuit and to control a high-current using a low-current signal. The input voltage of the relay signal ranges from 0 to 5V.

The relay module with a single channel board is used to manage high voltage, current loads like solenoid valves, motor, AC load & lamps. This module is mainly designed to interface through different microcontrollers like PIC, Arduino, etc.

5V Relay Module Pin Configuration:

The pin configuration of the 5V relay module description is shown below. This module includes 6-pins where each pin and its functionality are discussed below

- Normally Open (NO): This pin is normally open unless we provide a signal to the relay modules signal pin. So, the common contact pin smashes its link through the NC pin to make a connection through the NO pin
- Common Contact: This pin is used to connect through the load that we desire to switch by using the module.
- Normally Closed (NC): This NC pin is connected through the COM pin to form a closed circuit. However, this NC connection will break once the relay is switched through providing an active high/low signal toward the signal pin from a microcontroller.
- Signal Pin: The signal pin is mainly used for controlling the relay. This pin works in two cases like active low otherwise active high. So, in active low case, the relay activates once we provide an active low signal toward the signal pin, whereas, in an active high case, the relay will trigger once we provide a high signal toward the signal pin.

However, these modules generally work on an active high signal which will strengthen the relay coil to make contact with the common terminal with the normally open terminal.

- 5V VCC: This pin needs 5V DC to work. So 5V DC power supply is provided to this pin.
- Ground: This pin connects the GND terminal of the power supply.

3.1.8 Push Buttons:

Fig.3.1.8. Push Buttons

A push-button or simply button is a simple switch mechanism to control some aspect of a machine or a process. Buttons are typically made out of hard material, usually plastic or metal. The surface is usually flat or shaped to accommodate the human finger or hand, so as to be easily depressed or pushed. Buttons are most often biased switches, although many un-biased buttons (due to their physical nature) still require a spring to return to their un-pushed state. Terms for the "pushing" of a button include pressing, depressing, mashing, slapping, hitting, and punching.

3.1.8. Siren Alarm:

Fig.3.1.9: Siren Alarm

A siren is a loud noise-making device. Civil defense sirens are mounted in fixed locations and used to warn of natural disasters or attacks. Sirens are used on emergency service vehicles such as ambulances, police cars, and fire trucks.

2. Software tools:

2.1. Arduino IDE

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards. The source code for the IDE is released under the GNU General Public License, version 2.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution.

The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware. By default, avrdude is used as the uploading tool to flash the user code onto official Arduino boards. Arduino IDE is a derivative of the Processing IDE, however as of version 2.0, the Processing IDE will be replaced with the Visual Studio Code-based Eclipse Theia IDE framework. With the rising popularity of Arduino as a software platform, other vendors started to implement custom open-source compilers and tools (cores) that can build and upload sketches to other microcontrollers that are not supported by Arduino's official line of microcontrollers.

2.2. Proteus 8 Professional

Proteus is a Virtual System Modelling (VSM) that combines circuit simulation, animated components and microprocessor models to co-simulate the complete microcontroller based designs.

This is the perfect tool for engineers to test their microcontroller designs before constructing a physical prototype in real time. This program allows users to interact with the design using on-screen indicators and/or LED and LCD displays and, if attached to the PC, switches and buttons. One of the main components of Proteus is the Circuit Simulation -- a product that uses a SPICE3f5 analogue simulator kernel combined with an event-driven digital simulator that allow users to utilize any SPICE model by any manufacturer.

Proteus VSM comes with extensive debugging features, including breakpoints, single stepping and variable display for a neat design prior to hardware prototyping.
Proteus is the program to use when you want to simulate the interaction between software running on a microcontroller and any analog or digital electronic device connected to it.

Working of the complete mini-project module

This small-scale DC solar power system needs three basic components of solar panel installation system i.e. a Panel, Battery, Solar charge controller. As we have only single units of solar panel and battery etc. This way, we don't need to connect them in series, parallel or combo of series-parallel connection. We will directly connect them to the charge controller, battery and DC loads. The following solar panel wiring diagram shows that PV panel is connected to the solar charge controller (Panel Negative terminal of panel to the negative terminal of MPPT charge controller and vice versa for positive terminal. The battery is connected to the battery terminals of the charge controller (same like PV panel wiring i.e. positive terminal of battery is connected to the positive terminal of charge controller and vice versa for negative terminals).

The DC load is directly connected to the charge controller and can be fed up by two power sources. The direct load can be directly powered up through solar power and a charge controller during the sunshine/day time and during the shading/night, the DC load can be powered up using the battery stored energy as backup power.
The solar panel will also charge the battery but the charging time of the battery depends on the solar panel wattage, sunshine and ON/OF condition of direct load.
The circuit diagram of the Automatic School Bell system is very simple. It starts with the 4 push buttons. Each Push button is connected in series with a resistor. This is a Pullup resistor. When the push button is not pressed 5Volts are available at the controller pin. When the push button is pressed the ground is given as the signal to the Arduino's I/O pin. Push Buttons 1 to Push Buttons 4 are connected with the Arduino's Analog Pins A0 to A3.
The RTC DS1307 has a built-in button cell that allows keeping track of real-time irrespective of the power supply. For interfacing with the Arduino board, SDA and SCL pins of the RTC are connected to the SDA (Pin A4) and SCL (Pin A5) pins of the Arduino Uno board.
DC female power jack is connected to the 12V battery. The advantage of using 12V battery is that, we get 12Volts and the regulated 5Volts. 12volts can be used to power up the Relay while the 5Volts can be used to power up the Arduino or other 5 volts electronics.

The 16X2 LCD display is used to display the messages prompting to input time-table information. It is connected to the Arduino Uno board by connecting its data pins to pins 2 to 7 of the Arduino Uno board. The RS and E pins of the 16×2 LCD are connected to pins 3 and 2 of the Arduino Uno respectively. The RW pin of the LCD is grounded. Pin number 1 and pin number 16 of the LCD are connected with the ground while the pins 2 and 15 are connected with the 5 volts.

On the right side you can see a 5V 1-channel relay module of the type SPDT. You can use a readymade relay module or you can build the one by yourself by following these connections. As this is a 5V relay that's why one side of the relay coil is connected with the 5volts while the other side of the relay coil is connected to Siren. The Siren will be connected between the common and normally open legs of the relay.

Chapter-4

Results and Discussions

With this basic design of Automatic collage bell provides an opportunity of selecting the suitable time schedule for classes by momentarily pressing one of the push buttons. This signals the Arduino to carry out the specific task, thereby ringing the siren at a regular time interval and it is powered by Solar PV System. It will give maximum accuracy and as per the timing which can be easily reprogrammed by a Arduino IDE and can also vary timing for some classes as per the schedule of the college. We can say that it will be much useful for colleges or schools or other educational institutions.

Chapter-5

Applications, Advantages and Limitations

Applications:

This project can be used in

- Industries.
- Educational institutes like colleges, universities, schools.

Advantages:

- Efficient use of Renewable resources.
- Manpower can be eliminated.
- Stand-alone device – No Computer or any other Hardware are required.

Limitations:

- Cloudy days does not Produce as much energy

Chapter-6

Conclusions and Future Work

In conclusion, in present time world are going for automation accuracy and time saving, in the energy sector they are going for non- conventional energy sources. Here, automatic college bell can be successfully designed on Arduino based automated operation and also power by solar PV system. It can be applicable in school and colleges as per to save manpower and also to save time and decrease the human error. It will give maximum accuracy and as per the timing which can be easily reprogrammed by a Arduino IDE and can also vary timing for some classes as per the schedule of the college. We can say that it will be much useful for colleges or schools or other educational institutions. The world technology is going to more and more in automation for less man and money saving. It will also help in industries for performing different operation in specific time interval.

A lot more advancement can be done in this design. It can done by using GSM Module or Bluetooth Module. Through GSM the RTC can be controlled and so the timings can be edited. Automatic bell system with announcement can be made. In future much advanced automatic bell system can be made.

References

- Abubakar Isa, H. K. Verma, Abdurrahman Shuaibu, and Mubarak Auwal Saleh. (2018). Implementation ofMicrocontroller Based Distance Relay. *European Journal of Electrical and ComputerEngineering.* 2 (5):17-21.

- Abyash Gautam, Deepak Rasaily, Sejal Dahal. (2016). Microcontroller Controlled Automated College Bell. *International Journal of Engineering Trends and Technology.* 32 (4): 184-187.

- Adedoyin M. A., Olopade o.I., Shoewu O. O.,Ogunlewe A.O. (2018). Design and Implementation of MicrocontrollerBased Calculator. *Journal of computation in biosciences and engineering. 1(2):1-4.*

- Burgoji Santhosh Kumar (2018). Implementation of Automatic College Bell System Using Arduino. *International Journal of Current Engineering and Scientific Research.* 5 (4): 367-371.

- Henry Ohiani Ohize, Elizabeth Nonye Onwuka and Ahmed Ibrahim. (2011). Design of Microcontroller-Based Automatic School Bell. *AU J.T*. 15(2): 121-128.

  https://learn.sparkfun.com/tutorials/what-is-an-arduino/
  https://www.electronicshub.org/arduino-introduction/

- Liping Guo, Paul Curtis, Andrew Barendregt, and Anthony Surillo (2009). Design and Implementation of A Sun Tracking Solar Power System. American Society for Engineering Education,:14.122.1- 14.122.11.

- Shabnum Rasool Shaikh, Snehal Suryakant Kale, Arti Ashokrao Suryawanshi, Prof. Chalkikar G.V. (2018). Arduinobased Real Time Clock with Ringing of Bell and National Anthem. *International Journal on Recent and Innovation Trends in Computing and Communication.* 6 (6): 33-35.

- Sheenu Choudhary, Shrikant and Priyanka Sharma. (2014). Automatic college bell system. *International Journal of scientific research and management*. 2 (3): 668-673.

- Syed Naveed Uddin, Mohd Omer Nawaz, T.Kalinga Raj, and Shaik MahammedRasool (2017). Automatic School Bell with User defined Time Schedule. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering.* 6 (2): 994-998.

- Vaishnavi.D.R, Neha Khanum, ApoorvaSingh A and Sumaya Afreen (2017). Automated College Bell System with Wireless Control. *InternationalJournal of Advanced Research in Electrical, Electronics and Instrumentation Engineering.* 6 (5): 3340-3345.

Appendix

Arduino IDE code:

```
#include <DS3231.h>//Memanggil RTC3231 Library
#include <Wire.h>  // i2C Conection Library
#include <LiquidCrystal.h> //Libraries
#include <EEPROM.h>

LiquidCrystal lcd(2, 3, 4, 5, 6, 7); //Arduino pins to lcd

#define bt_clock  A0
#define bt_up     A1
#define bt_down   A2
#define bt_timer  A3

#define relay 8
#define buzzer 13

// Init DS3231
DS3231  rtc(SDA, SCL);

// Init a Time-data structure
Time  t; //pencacah string time()

int hh = 0, mm = 0, ss = 0, dd = 0, bb = 0, set_day;
int yy = 0;
String Day = "  ";

int StartHH  = 0, StartMM  = 0, FinishHH  = 0, FinishMM  = 0, setMode = 0, setAlarm =
0, alarmMode=1;
int Start1HH, Start1MM, Finish1HH, Finish1MM;
int Start2HH, Start2MM, Finish2HH, Finish2MM;
int Start3HH, Start3MM, Finish3HH, Finish3MM;
```

```
int Start4HH, Start4MM, Finish4HH, Finish4MM;

int timer1, timer2, timer3, timer4;

int stop =0, mode=0, flag=0;

void setup(){
rtc.begin(); // memulai koneksi i2c dengan RTC

pinMode(bt_clock, INPUT_PULLUP);
pinMode(bt_up,    INPUT_PULLUP);
pinMode(bt_down,  INPUT_PULLUP);
pinMode(bt_timer, INPUT_PULLUP);

pinMode(relay, OUTPUT);
digitalWrite(relay, HIGH);

pinMode(buzzer, OUTPUT);

  lcd.begin(16, 2); // Configura lcd numero columnas y filas
  lcd.setCursor(0,0);  //Show "TIME" on the LCD

  lcd.setCursor (0,0);
  lcd.print(" Real Time Clock ");
  lcd.setCursor (0,1);
  lcd.print("  4Timer 1Relay  ");
  delay (2000);
  lcd.clear();

stop=EEPROM.read(50);
if(stop==0){
}else{
mode=1;WriteEeprom ();delay (20);
mode=2;WriteEeprom ();delay (20);
```

```
mode=3;WriteEeprom ();delay (20);
mode=4;WriteEeprom ();delay (20);
mode=0;
}
EEPROM.write(50,0);

ReadEeprom();
//Set RTC Untuk Pertama kali
//rtc.setDOW(2);     // Set Day-of-Week to SUNDAY
//rtc.setTime (00, 9, 50);
//rtc.setDate(12, 11, 2017);
}

void loop(){
t = rtc.getTime();
Day = rtc.getDOWStr(1);

if (setMode == 0){
hh = t.hour,DEC;
mm = t.min,DEC;
ss = t.sec,DEC;
dd = t.date,DEC;
bb = t.mon,DEC;
yy = t.year,DEC;
//if(t.hour>12){hh=t.hour-12;}// for removing 24 houre
//else{hh=t.hour;}
}

 if(setAlarm==0){
 lcd.setCursor(0,0);
 lcd.print((hh/10)%10);
 lcd.print(hh % 10);
 lcd.print(":");
 lcd.print((mm/10)%10);
```

```
 lcd.print(mm % 10);
 lcd.print(":");
 lcd.print((ss/10)%10);
 lcd.print(ss % 10);
 lcd.print("  T:");
 lcd.print(rtc.getTemp(),0);
 lcd.write(223);
 lcd.print("C");
 lcd.print("  ");

 lcd.setCursor(1,1);
 lcd.print(Day);
 lcd.print(" ");
 lcd.print((dd/10)%10);
 lcd.print(dd % 10);
 lcd.print("/");
 lcd.print((bb/10)%10);
 lcd.print(bb % 10);
 lcd.print("/");
 lcd.print((yy/1000)%10);
 lcd.print((yy/100)%10);
 lcd.print((yy/10)%10);
 lcd.print(yy % 10);
 }


setupClock();
setTimer();
delay (100);
blinking();

//Timer1 ON
if (timer1==1 && alarmMode==1 && hh==Start1HH && mm==Start1MM)
{digitalWrite(relay, LOW);}
```

```
//Timer1 OFF
if (timer1==1 && alarmMode==1 && hh==Finish1HH &&
mm==Finish1MM){digitalWrite(relay, HIGH);}

//Timer2 ON
if (timer2==1 && alarmMode==1 && hh==Start2HH && mm==Start2MM)
{digitalWrite(relay, LOW);}
//Timer2 OFF
if (timer2==1 && alarmMode==1 && hh==Finish2HH &&
mm==Finish2MM){digitalWrite(relay, HIGH);}

//Timer3 ON
if (timer3==1 && alarmMode==1 && hh==Start3HH && mm==Start3MM)
{digitalWrite(relay, LOW);}
//Timer3 OFF
if (timer3==1 && alarmMode==1 && hh==Finish3HH &&
mm==Finish3MM){digitalWrite(relay, HIGH);}

//Timer4 ON
if (timer4==1 && alarmMode==1 && hh==Start4HH && mm==Start4MM)
{digitalWrite(relay, LOW);}
//Timer4 OFF
if (timer4==1 && alarmMode==1 && hh==Finish4HH &&
mm==Finish4MM){digitalWrite(relay, HIGH);}

delay (100);
digitalWrite(buzzer, LOW);
}

void blinking (){
//BLINKING SCREEN
//Set Clock
if (setAlarm <2 && setMode == 1){lcd.setCursor(0,0);  lcd.print("  ");}
if (setAlarm <2 && setMode == 2){lcd.setCursor(3,0);  lcd.print("  ");}
```

```
if (setAlarm <2 && setMode == 3){lcd.setCursor(6,0);  lcd.print("  ");}
if (setAlarm <2 && setMode == 4){lcd.setCursor(1,1);  lcd.print("    ");}
if (setAlarm <2 && setMode == 5){lcd.setCursor(5,1);  lcd.print("  ");}
if (setAlarm <2 && setMode == 6){lcd.setCursor(8,1);  lcd.print("  ");}
if (setAlarm <2 && setMode == 7){lcd.setCursor(11,1); lcd.print("    "); }

//Set Timer
if (setMode == 0 && setAlarm == 1 && mode==0){lcd.setCursor(2,1); lcd.print("  "); }
if (setMode == 0 && setAlarm == 2 && mode==0){lcd.setCursor(6,1); lcd.print("  "); }
if (setMode == 0 && setAlarm == 3 && mode==0){lcd.setCursor(10,1); lcd.print("  "); }
if (setMode == 0 && setAlarm == 4 && mode==0){lcd.setCursor(13,1); lcd.print("  "); }

if (setMode == 0 && setAlarm == 1 && mode>0){lcd.setCursor(11,0); lcd.print("  "); }
if (setMode == 0 && setAlarm == 2 && mode>0){lcd.setCursor(14,0); lcd.print("  "); }
if (setMode == 0 && setAlarm == 3 && mode>0){lcd.setCursor(11,1); lcd.print("  "); }
if (setMode == 0 && setAlarm == 4 && mode>0){lcd.setCursor(14,1); lcd.print("  "); }
}

//Seting Jam ,Tanggal,Alarm/Timer
void setupClock (void) {

  if (setMode == 8){
  lcd.setCursor (0,0);
  lcd.print ("Set Time Finish ");
  lcd.setCursor (0,1);
  lcd.print ("Set Date Finish ");
  delay (1000);
  rtc.setTime (hh, mm, ss);
  rtc.setDate (dd, bb, yy);
  lcd.clear();
  setMode = 0;
  }

 if (setAlarm >0){alarmMode=0;}
```

```
if(digitalRead (bt_clock) == 0 && flag==0) {flag=1;
if(setAlarm>0){WriteEeprom(); setAlarm=1; mode =5;}
else{setMode = setMode+1;}
digitalWrite(buzzer, HIGH);
}

if(digitalRead (bt_timer) == 0 && flag==0){flag=1;
if(setMode>0){setMode=8;}
 else{
 setAlarm = setAlarm+1;
 if(setAlarm>4){setAlarm=1;
 WriteEeprom ();
 mode=mode+1;
 ReadEeprom();
 }
 }
 lcd.clear();
 digitalWrite(buzzer, HIGH);
 }

   if(setAlarm == 1 && mode==5){
   lcd.setCursor (0,0);
   lcd.print ("Set Timer Finish");
   lcd.setCursor (0,1);
   lcd.print ("-EEPROM Updated-");
   delay (2000);
   lcd.clear();
   setAlarm=0;
   mode =0;
   alarmMode=1;
   }

if(digitalRead (bt_clock) == 1 && digitalRead (bt_timer) == 1){flag=0;}
```

```
if(digitalRead (bt_up) == 0){
        if (setAlarm<2 && setMode==1)hh=hh+1;
        if (setAlarm<2 && setMode==2)mm=mm+1;
        if (setAlarm<2 && setMode==3)ss=ss+1;
        if (setAlarm<2 && setMode==4)set_day=set_day+1;
        if (setAlarm<2 && setMode==5)dd=dd+1;
        if (setAlarm<2 && setMode==6)bb=bb+1;
        if (setAlarm<2 && setMode==7)yy=yy+1;
        //Timer
        if (mode==0 && setMode==0 && setAlarm==1)timer1=1;
        if (mode==0 && setMode==0 && setAlarm==2)timer2=1;
        if (mode==0 && setMode==0 && setAlarm==3)timer3=1;
        if (mode==0 && setMode==0 && setAlarm==4)timer4=1;

        if (mode>0 && setMode==0 && setAlarm==1)StartHH=StartHH+1;
        if (mode>0 && setMode==0 && setAlarm==2)StartMM=StartMM+1;
        if (mode>0 && setMode==0 && setAlarm==3)FinishHH=FinishHH+1;
        if (mode>0 && setMode==0 && setAlarm==4)FinishMM=FinishMM+1;

if(hh>23)hh=0;
if(mm>59)mm=0;
if(ss>59)ss=0;
if(set_day>7)set_day=0;
if(dd>31)dd=0;
if(bb>12)bb=0;
if(yy>2030)yy=2000;
if(StartHH>23)StartHH=0;
if(StartMM>59)StartMM=0;
if(FinishHH>23)FinishHH=0;
if(FinishMM>59)FinishMM=0;
rtc.setDOW(set_day);
digitalWrite(buzzer, HIGH);
 }
```

```
if(digitalRead (bt_down) == 0){
        if (setAlarm<2 && setMode==1)hh=hh-1;
        if (setAlarm<2 && setMode==2)mm=mm-1;
        if (setAlarm<2 && setMode==3)ss=ss-1;
        if (setAlarm<2 && setMode==4)set_day=set_day-1;
        if (setAlarm<2 && setMode==5)dd=dd-1;
        if (setAlarm<2 && setMode==6)bb=bb-1;
        if (setAlarm<2 && setMode==7)yy=yy-1;
        //Timer
        if (mode==0 && setMode==0 && setAlarm==1)timer1=0;
        if (mode==0 && setMode==0 && setAlarm==2)timer2=0;
        if (mode==0 && setMode==0 && setAlarm==3)timer3=0;
        if (mode==0 && setMode==0 && setAlarm==4)timer4=0;

        if (mode>0 && setMode==0 && setAlarm==1)StartHH=StartHH-1;
        if (mode>0 && setMode==0 && setAlarm==2)StartMM=StartMM-1;
        if (mode>0 && setMode==0 && setAlarm==3)FinishHH=FinishHH-1;
        if (mode>0 && setMode==0 && setAlarm==4)FinishMM=FinishMM-1;

if(hh<0)hh=23;
if(mm<0)mm=59;
if(ss<0)ss=59;
if(set_day<0)set_day=7;
if(dd<0)dd=31;
if(bb<0)bb=12;
if(yy<0)yy=2030;
if(StartHH<0)StartHH=23;
if(StartMM<0)StartMM=59;
if(FinishHH<0)FinishHH=23;
if(FinishMM<0)FinishMM=59;
rtc.setDOW(set_day);
digitalWrite(buzzer, HIGH);
 }

}
```

```
void setTimer (){
//Timer
if (setMode == 0 && setAlarm >0 && mode>0){
 lcd.setCursor (0,0);
 lcd.print("Timer");
 lcd.print(mode);
 lcd.print(" On :");
 lcd.setCursor (11,0);
 lcd.print((StartHH/10)%10);
 lcd.print(StartHH % 10);
 lcd.print(":");
 lcd.print((StartMM/10)%10);
 lcd.print(StartMM % 10);

 lcd.setCursor (0,1);
 lcd.print("Timer");
 lcd.print(mode);
 lcd.print(" Off:");
 lcd.setCursor (11,1);
 lcd.print((FinishHH/10)%10);
 lcd.print(FinishHH % 10);
 lcd.print(":");
 lcd.print((FinishMM/10)%10);
 lcd.print(FinishMM % 10);
 }

if (setMode == 0 && setAlarm >0 && mode==0){
 lcd.setCursor (0,0);
 lcd.print(" T1  T2  T3  T4 ");
 lcd.setCursor (0,1);
   if(timer1==1){lcd.print("  A");}
 else{lcd.print("  D");}

  if(timer2==1){lcd.print("  A");}
```

```
 else{lcd.print("   D");}

   if(timer3==1){lcd.print("   A");}
 else{lcd.print("   D");}

   if(timer4==1){lcd.print("   A");}
 else{lcd.print("   D");}
}
}

void ReadEeprom() {
Start1HH=EEPROM.read(11);Start1MM=EEPROM.read(12);Finish1HH=EEPROM.read
(13);Finish1MM=EEPROM.read(14);
Start2HH=EEPROM.read(21);Start2MM=EEPROM.read(22);Finish2HH=EEPROM.read
(23);Finish2MM=EEPROM.read(24);
Start3HH=EEPROM.read(31);Start3MM=EEPROM.read(32);Finish3HH=EEPROM.read
(33);Finish3MM=EEPROM.read(34);
Start4HH=EEPROM.read(41);Start4MM=EEPROM.read(42);Finish4HH=EEPROM.read
(43);Finish4MM=EEPROM.read(44);

if(mode==1){StartHH=Start1HH, StartMM=Start1MM,
FinishHH=Finish1HH,FinishMM=Finish1MM;}
if(mode==2){StartHH=Start2HH, StartMM=Start2MM,
FinishHH=Finish2HH,FinishMM=Finish2MM;}
if(mode==3){StartHH=Start3HH, StartMM=Start3MM,
FinishHH=Finish3HH,FinishMM=Finish3MM;}
if(mode==4){StartHH=Start4HH, StartMM=Start4MM,
FinishHH=Finish4HH,FinishMM=Finish4MM;}

timer1=EEPROM.read(1);
timer2=EEPROM.read(2);
timer3=EEPROM.read(3);
timer4=EEPROM.read(4);
}
```

```
void WriteEeprom() {
if(mode==1){EEPROM.write(11,StartHH);EEPROM.write(12,StartMM);EEPROM.write(
13,FinishHH);EEPROM.write(14,FinishMM);}
if(mode==2){EEPROM.write(21,StartHH);EEPROM.write(22,StartMM);EEPROM.write(
23,FinishHH);EEPROM.write(24,FinishMM);}
if(mode==3){EEPROM.write(31,StartHH);EEPROM.write(32,StartMM);EEPROM.write(
33,FinishHH);EEPROM.write(34,FinishMM);}
if(mode==4){EEPROM.write(41,StartHH);EEPROM.write(42,StartMM);EEPROM.write(
43,FinishHH);EEPROM.write(44,FinishMM);}

EEPROM.write(1,timer1);
EEPROM.write(2,timer2);
EEPROM.write(3,timer3);
EEPROM.write(4,timer4);
}
```

Photographs