

# **Microcontroller Based Digital Clock With Bluetooth Module Controlled by Android Apps.**

This project report is submitted to the Department of Electrical & Electronic Engineering (EEE), Daffodil International University, Bangladesh, in Partial Fulfillment of the requirements for the Degree of “Bachelor of Science in Electrical & Electronic Engineering”.

## **Prepared By**

**Md. Ali Azam                      ID: 153-33-3005**

**Md. Abu Sufian                    ID: 153-33-3013**

## **Supervised By**

**Md. Mahmudur Rahman**

Assistant Professor

Department of Electrical & Electronic Engineering (EEE)

Daffodil International University



**Department of Electrical and Electronic Engineering**

**Faculty of Engineering**

**DAFFODIL INTERNATIONAL UNIVERSITY**

**December 2018**

# Certification

This is to certify that this Project report entitled “**Microcontroller Based Digital clock with Bluetooth module control by android apps**” is done by the following students under my direct supervision and this work has been carried out by them in the laboratories of the Department Of Electrical And Electronic Engineering under the Faculty of Engineering of Daffodil International University in partial fulfilment of the requirements for the degree of Bachelor of Science in Electrical And Electronic Engineering. The presentation of the work was held on .....

**Name: Md. Ali Azam**

ID: 153-33-3005

**Name: Md. Abu Sufian**

ID: 153-33-3013

**Supervised By**



**Md. Mahmudur Rahman**

**Assistant Professor**

**Faculty of Engineering**

**Daffodil International University**

December 2018  
Md. Mahmudur Rahman  
Assistant Professor  
Faculty of Engineering  
Daffodil International University

**Subject: Letter of Approval.**

Sir,

We are very glad to submit this thesis report titled “**Digital clock with Bluetooth module control by android apps**” It was a pleasure to work on such an important Project. We have worked hard and have tried our best to make this report properly according to your instruction. This Project has done according to the requirements of the Daffodil International University. We hope that you would kindly consider any mistake on our part in preparing this Project report. We believe that this Project report will be acceptable to you in partial fulfillment of the requirement for awarding the degree of Bachelor of Science in Electrical & Electronic Engineering.

Sincerely yours

Md. Ali Azam

ID: 153-53-3005

Md. Abu Sufian

ID: 153-33-3013

**Supervised By**



---

**Md. Mahmudur Rahman**

**Assistant Professor**

**Faculty of Engineering**

**Daffodil International University**

## ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty Allah for his divine Blessing makes us possible to complete this project successfully.

We are grateful and wish our profound indebtedness to supervisor **Md. Mahmudur Rahman, Assistant professor**, Department of EEE Daffodil International University, Dhaka. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. M. Shamsul Alam**, Professor, and Head, Department of EEE, for his kind help to finish our project and also to other faculty members of EEE department of Daffodil International University.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

-----*Author*

## **ABSTRACT**

A digital clock is a type of clock that displays the time digitally, i.e. in ciphers, as opposed to an analog clock, where the time is displayed by hands. Usually, digital clocks are associated with electronic drives.

Digital clocks typically use the 50 or 60 hertz oscillator of AC power or crystal oscillator as in a quartz clock to keep time. Internationally digital clocks display the hours of the day in 24 hour format but in our country a more commonly used hour sequence is 12 hour format (with some indication of AM or PM).

In this project to represent the time, Seven-segment Display has been used for each of four digit which shows hours and minutes and in order to represent seconds LED's are used for each and every seconds. Moreover, as they run on electricity, most digital clocks must be reset every time they are moved or the power is cut off. To reduce the problem, here a battery backup has been used to maintain the time during power outages. In this project we have used RTC (Real Time Clock).

Two switches have been used, one for entering the time and second is for time increment. This digital clock is inexpensive device that makes product designs more popular.

This project uses regulated 5V & 1A power supply. 7805 three terminal voltage regulator is used for voltage regulation. An adapter is used to rectify AC output and supply 10 V to the circuit input.

*DEDICATED*

*TO*

*OUR BELOVED PARENTS*

*&*

*HONORABLE TEACHERS*

## Table of Contents

<b>CHAPTERS</b>	<b>Titles</b>	<b>Pages</b>
<b>Chapter 1</b>	<b>Introduction</b>	<b>1-8</b>
1.1	Introduction	2-3
1.2	Objectives	3
1.3	Historical Background	4-5
1.4	Why Invented Digital Clock	5-6
1.5	Before Invention of Clock	7-8
<b>Chapter 2</b>	<b>Related and existed work</b>	<b>9-13</b>
2.1	Digital Clock Circuit using 8051 and DS12C887	10-12
2.2	Circuit Diagram Digital Clock Circuit using 8051 and DS1307	12-13
<b>Chapter 3</b>	<b>Literature Review Parts and Components</b>	<b>14-58</b>
3.1	Seven segment display	15-20
3.2	Liquid Crystal Displays(LCD)	21-29
3.3	ATMega328P Microcontroller IC	30-34
3.5	HC-05 Bluetooth Module	34-40

3.6	DHT11 basic temperature-humidity sensor	41-43
3.7	CD4511 7-Segment Driver IC	44-47
3.8	Voltage Regulator 7805 IC	47-49
3.9	RTC (Real Time Clock)	50-53
3.10	Crystal Oscillator	54-57
3.11	Summary of This Chapter	58
<b>Chapter 4</b>	<b>Methodology: Circuit Design and Simulation</b>	<b>59-65</b>
4.1	Design and Development	60-61
4.2	Flow Chart	62
4.3	Diagram of Seven Segment Clock Circuit	63
4.4	Diagram of Day and Date Circuit	64
4.5	Diagram of Temperature Circuit	65
4.6	PCB Design	65
4.7	Apps Design	66
4.8	Logic of Apps	67
<b>Chapter 5</b>	<b>Cost Analysis</b>	<b>68-69</b>
5.1	Cost Table	69



<b>Chapter 6</b>	<b>Discussion and Recommendations</b>	<b>70-72</b>
6.1	Conclusion	71
6.2	Limitations of The Work	71
6.3	Future Scopes	71
6.4	Advantages	72
	<b>References</b>	<b>73-73</b>
	<b>Appendices</b>	<b>74-96</b>

<b>List Of Figures</b>	<b>Pages</b>
2.1: Digital Clock Circuit Using 8051 And Ds12c887	10
2.2: Digital Clock Using 8051 And Ds12c887 Circuit Diagram	11
2.3: Digital Clock Using 8051 And Ds1307 Circuit Diagram	12
2.4: Digital Clock Using 8051 And Ds1307	13
3.1: 7-Segmentdisplay	15
3.2: The Common Cathode (Cc) 7 Segment Display.	17
3.3: The Common Anode (Ca) 7 Segment Display.	18
3.4: 7-Segment Display	20
3.5: Lcd Display	21
3.6: Working Of Lcd	26
3.7: Lcd With I2c Module	27
3.8: Interface Function Of Lcd With I2c Module	29
3.9 (A): Atmega328p Ic	30
3.9 (B): Pin Names	30
3.10: Hc-05 Bluetooth Module	35
3.11: Hardware Specification Of Hc-05	36
3.12: Hc-05 Connection	39
3.13: Dht11 Temperature-Humidity Sensor	41
3.14: Dht11 Temperature-Humidity Sensor With Module	42
3.15: Dht11 Temperature-Humidity Sensor Circuit	42
3.16: Cd4511 7-Segment Driver Ic	44
3.17: Voltage regulator 7805 Ic	47
3.18: Real Time Clock	50
3.19: Pin Description Of Ds1307	51
3.20: Rtc Module	53
3.21: Crystal Oscillator	54

3.22: Crystal Oscillator Working	55
3.23: Colpitts Crystal Oscillator	57
4.1: FLOWCHART	62
4.2: FLOWCHART	63
4.3: DIAGRAM OF DAY AND DATE CIRCUIT	64
4.4: DIAGRAM OF DAY AND DATE CIRCUIT	65
4.5: PCB DESIGN FOR 7 SEGMENT PART	65
4.6: PCB DESIGN FOR TEMP & DATE PART	66
4.7: APPS INTERFACE	66
4.8: LOGIC OF APPS	67

# Chapter 1

## Introduction

## 1.1 Introduction

A digital clock is a type of clock that displays the time digitally, i.e. in ciphers, as opposed to an analog clock, where the time is displayed by hands. Digital clocks are often associated with electronics drives, but the digital description refers only to the display, not to the drive mechanism (Both analog and digital clocks can be driven either mechanically or electronically, but clockwork mechanisms with digital displays are rare).

In other word a digital clock is a type of clock that displays the time digitally, i.e. in numerals or other symbols, as opposed to an analog clock, where the time is indicated by the positions of rotating hands.

In this project, we have built a digital clock with 12 hour count time. The clock runs from 00:00 to 11:59 and then back to 00:00. Our display has four digits, two digits for minutes and two for hour. The specialty of this clock is that it has very low power consumption and condensed layout.

Every 10 minutes, we add one on the ten's digit of minute. After every 6 such cycles, we add one on the unit's digit of hour, and every 10 hours, we add one on tens digit of hour. As soon as it reaches 1 on the ten's digit of hour and 2 on unit's digit of hour, both the ten's and the unit's digit of hour are reset to 0. So, in practice, we do not see 12:00 displayed. We rather see a transition from 11:59 to 00:00.

This clock has constructed based on the microcontroller Ic Atmega 328p. Temperature and humidity sensor are also been used to know about weather. In this project 7 segment display has used to show the time of this clock. Also used Lcd display of 20\*4 and 16\*2 to show the Temperature and Humidity, and date.

Also used 8MHz crystal oscillator, here we used AC power source so we take aback up power from a battery. The outputs would be connected to LEDs and would make the LED glow for a high output and vice-versa for a low output. We make a seven segment display by using LEDs.

Digital clocks typically use the 50 or 60 hertz oscillation of AC power or a 32.768kHz crystal oscillator. A more commonly used hour sequence option is 12 hour format (with

some indication of AM or PM). Emulations of analog-style faces often use an LCD screen, and these are also sometimes described as digital.

This clock is easy to use though this clock is to set up on the wall but it can be control with an android phone by an app. The time can be update by Bluetooth control also can be set alarm easily with the phone. This clock getting supply from an adapter of 230/10 v also used a rechargeable battery for backup.

## **1.2 Objectives**

- Design And Implementation Of Digital Clock,
- Design Temperature Sensor and program in Arduino,
- Design Humidity Display System,
- Design Digital Date System,
- Make A Control Apps.

## 1.3 Historical Background

The German factory Aktiengesellschaft für Uhrenfabrikation Lenzkirch made such digital clocks in 1893 and 1894. The earliest patent for a digital alarm clock was registered by D.E Protzmann and others on October 23, 1956, in the United States.

The next development in accuracy occurred after 1656 with the invention of the pendulum clock. Galileo had the idea to use a swinging bob to regulate the motion of a time-telling device earlier in the 17th century. Christiaan Huygens, however, is usually credited as the inventor.

The first digital pocket watch was the invention of Austrian engineer Josef Pallweber who created his "jump-hour" mechanism in 1883. Instead of a conventional dial, the jump-hour featured two windows in an enamel dial, through which the hours and minutes are visible on rotating discs. The second hand remained conventional. By 1885 Pallweber mechanism was already on the market in pocket watches by Cortébert and IWC; arguably contributing to the subsequent rise and commercial success of IWC. The principles of Pallweber jump-hour movement had appeared in wristwatches by the 1920s (Cortébert) and are still used today (Chronoswiss Digiteur). While the original inventor didn't have a watch brand at the time, his name has since been resurrected by a newly established watch manufacturer.

Plato clocks used a similar idea but a different layout. These spring-wound pieces consisted of a glass cylinder with a column inside, affixed to which were small digital cards with numbers printed on them, which flipped as time passed. The Plato clocks were introduced at the St. Louis World Fair in 1904, produced by Ansonia Clock Company. Eugene Fitch of New York patented the clock design in 1903. 13 years earlier Josef Pallweber had patented the same invention using digital cards (different from his 1885 patent using moving disks) in Germany (DRP No. 54093). The German factory Aktiengesellschaft für Uhrenfabrikation Lenzkirch made such digital clocks in 1893 and 1894.

The earliest patent for a digital alarm clock was registered by D.E Protzmann and others on October 23, 1956, in the United States. Protzmann and his associates also patented another digital clock in 1970, which was said to use a minimal amount of moving parts.

Two side-plates held digital numerals between them, while an electric motor and cam gear outside controlled movement.

In 1970, the first digital wristwatch with an LED display was mass-produced. Called the Pulsar, and produced by the Hamilton Watch Company, this watch was hinted at two years prior when the same company created a prototype digital watch for Kubrick's 2001: A Space Odyssey. Throughout the 1970s, despite the initial hefty cost of digital watches, the popularity of said devices steadily rose.

Over the years, many different types of digital alarm clocks have been developed.

In the Soviet Union, the 7-segment digital clocks were known as Elektronika 7.

## **1.4 Why Invented Digital Clock**

Have you ever been curious about how digital clocks work? Or would you just like to know more about digital clocks to help you make a future purchase? Let's take a look into the world of digital clocks and learn a bit about them. The first pocket style watch that was considered "digital" was introduced in 1883, it used what was called a jump-hour mechanism. This invention was created by Josef Pallweber which the jump-hour featured two windows in an enameled dial. Through the two windows, you can see the hours and minutes on rotating discs – the second hand had remained conventional in nature. Next came Plato clocks, which had a similar idea to Josef Pallweber but used a different layout structure. Spring-wound pieces use a glass cylinder with a column inside, which was then attached to small digital cards with numbers printed on them – flipping as time went on. The Plato clock was patented in 1903 and introduced in 1904 by Ansonia clock company. D.E. Protzmann and his associates submitted the first patent for a digital alarm clock in 1956 and later patented another digital clock in 1970 which was supposed to involve a minimal amount of moving parts.



The first digital watch that included an LED display called the Pulsar, was mass-produced in 1970 by the Hamilton Watch Company. Over the many years since, many various types of digital watches, clocks, alarm clocks have been developed. Digital clocks typically run off of an electrical power supply such as a 100-240 volt AC power cord to the wall. However, most digital clocks have battery backup in case of a power outage. An electronic time base that ticks away at an accurate rate, and digital clocks generally handle gearing with a component called a counter. The display on a digital clock either consists of LEDs or LCD lights. Unlike analog clocks which are based on 12 hours, digital clocks have two options. You can either use a 12-hour clock, often with an AM for morning and PM for afternoon and evening or; a 24 hour clock. Even though the clocks are useful, some digital clocks have to be changed manually when the time changes. Some of the newer model digital clocks are now able to have their time synchronized automatically by a radio signal or the internet. Digital clocks are generally smaller and can be easier to use and read than analog clocks. Digital clocks can also be made much larger. Because of the inexpensive design and size flexibility, digital clocks are often used in cars, microwave ovens, radios, and computers. Digital clocks help to make our lives a little bit easier – sometimes without us realizing it. The digital clock helps in our everyday lives.

## 1.5 Before Invention of Clock

Beginning in prehistoric days, humankind started reading the elements around them in order to measure time. Ancient humans used the power of simple observation, deductive reasoning, and the earth itself to complete their time tracking goals. These observations were all about reading the stars high up in the heavens, the changes in the seasons, and also by the presence of day and night. Humans learned to come up with very primitive methods to determine time. Telling time made it possible for prehistoric man to be able to plan any type of nomadic activity, any farming, sacred kind of feasts, or anything else that was important to him.

One of the earliest of all devices to tell time was the sundial. The sundial is looked on as being a form of sun-powered clock. Ancient civilizations such as the Sumerians did have this knowledge, but when the culture died, knowledge was lost along with many other aspects of the civilization unfortunately. One thing that is known is this. It was the ancient Egyptians who did come up with a form of system to divide the day up into parts. These parts were a whole lot like hours in description. The Egyptians constructed obelisks, which were four-sided and well tapered monuments, and they were geographically located in certain places. These obelisks were believed to have been constructed around 3500 BC and a shadow would be cast by the obelisk as the sun made it's way across the sky. This obelisk would be marked out in certain sections to see the two halves of the day. There was another more advanced shadow clock or sundial in use by the ancient Egyptians around 1500 BC. This shadow clock or sundial permitted one to measure the passage of hours within a day.

Another very early form of clock to tell the time was the water clock. The water clock was used by the ancient Greeks. It is considered to be one of the earliest forms of timekeeping devices that didn't take use the observance of the celestial bodies to help calculate the passage of time. It is believed that the ancient Greeks started using these early timekeeping devices around 325 BC. These clocks were mainly used to determine the hours of nighttime, but they may also have been used for daylight hours as well.

The hourglass is another form of early timekeeping device that was used in ancient times. The hourglass was constructed from two separate glass bulbs that were rounded. These two glass bulbs were connected by a neck of narrow glass that was situated between the bulbs. This hourglass contained particles of sand within it, and when it was turned upside down, a measured amount of sand particles would drop down from the top part of the glass to the bottom part of the glass. The hourglass of ancient times is said to be the early predecessor for the egg timer of modern times. Telling time before clocks did take place. It was just done in a different way from how it is done today.

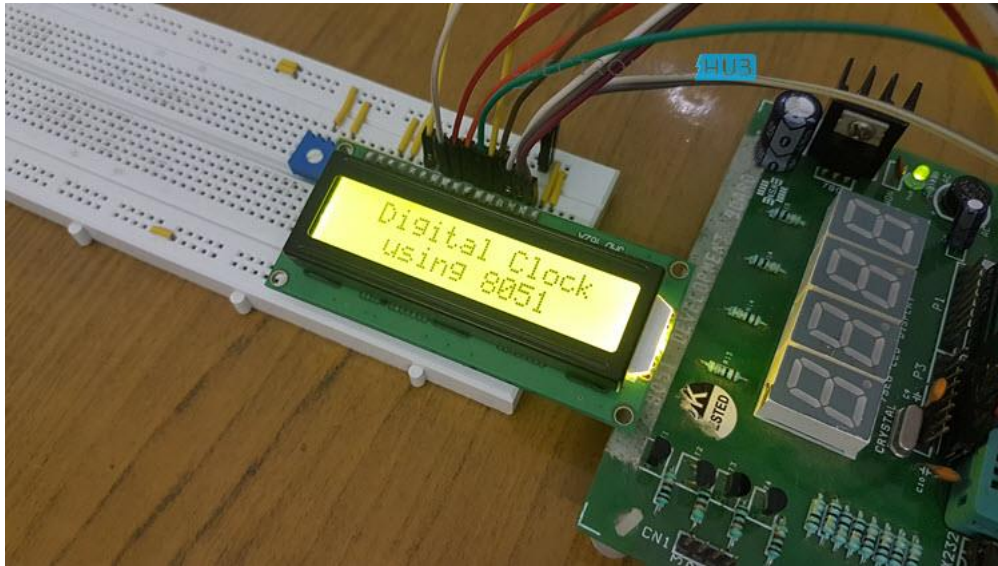
Égard Watches invites you to check out our quality luxury timepieces that do more than tell the time, they capture moments in time.

# Chapter 2

## Related & Existed Works

## 2.1 Digital Clock Circuit using 8051 and DS12C887

**Introduction:** This Project shown Digital Clock Circuit using 8051 and DS12C887 as well as DS1307 RTC Modules.



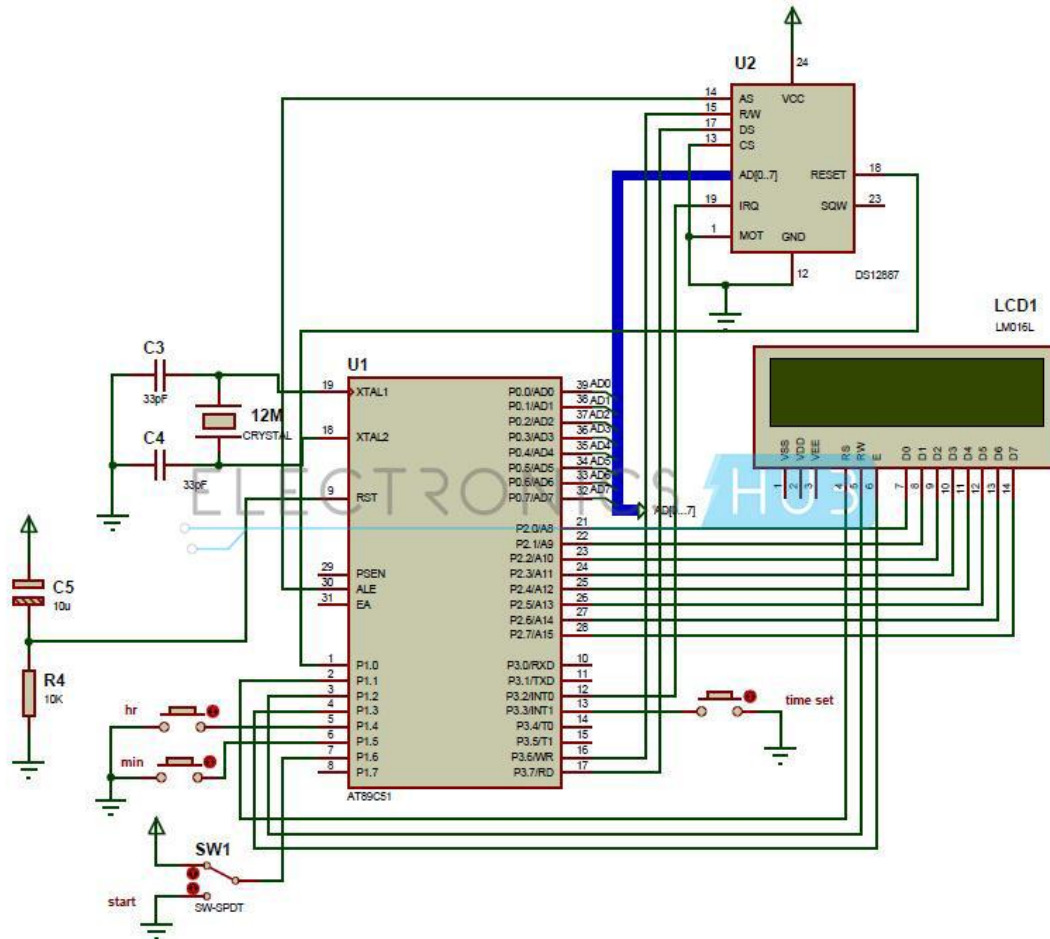
**Figure-2.1:** Digital Clock Circuit using 8051 and DS12C887

A digital clock displays the time using numbers and it has many applications like cars, railway stations, houses, offices, etc. in order to provide accurate time and date. In this type of applications, normally we use RTC (Real Time Clock) ICs to display the time and date accurately.

The circuit displays the time on LCD. For this clock, we can set the time at any instant. Here, the clock can work in either 24 hour mode or 12 hour mode and the RTC chip is configured by programming 8051 controller.

Here demonstrate two circuits of Digital Clocks using 8051 Microcontroller: one uses the RTC DS12C887 and the other uses the RTC DS1307

**Circuit Principle:** The main principle of both the circuits is the 8051 controller continuously reads the data from Real time clock IC's and process it in correct order to display the time on LCD.



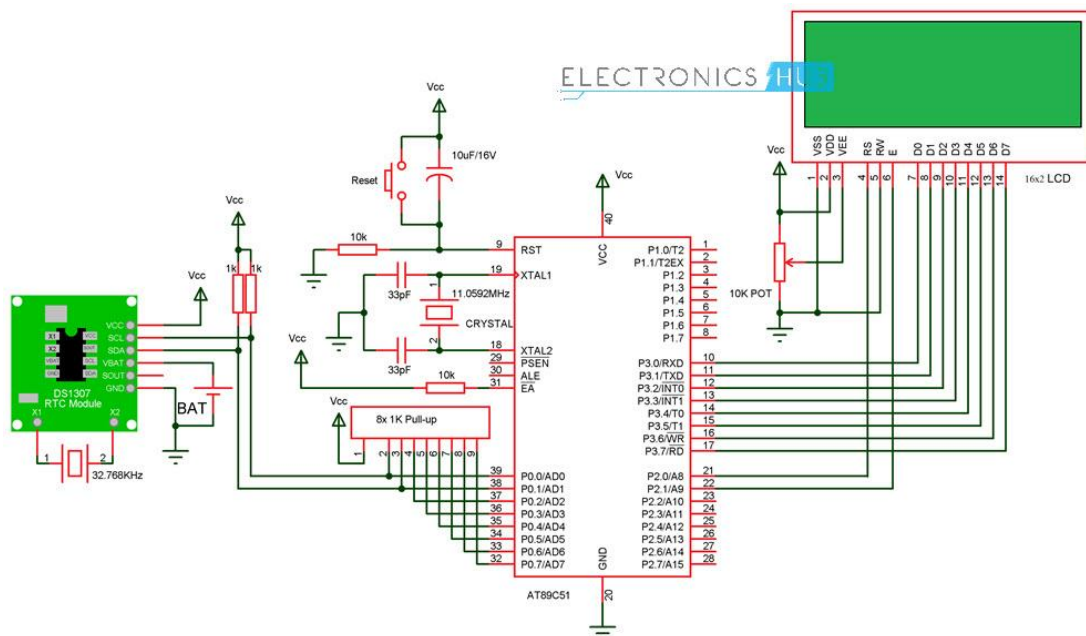
**Figure-2.2:** Digital Clock using 8051 and DS12C887 circuit diagram

**Circuit Design:** The circuit shows you how interface RTC IC to the 8051 controller. Port P0 is used as a data port of Real time clock. Port P2 of controller is connected to the data pins of LCD. Pins P1.1, P1.2 and P1.3 of controller are connected to the RS, RW, EN pins respectively. P1.0 is connected to the RESET of RTC. Push buttons are connected to the P1.4 and P1.5. These are used set the time. P1.6 is configured as START pin used to run the clock with time set by the user. P3.3 is connected to the push button used to call the set\_time function.

**DS12C887 Real Time Clock:** This IC is used most of the applications to provide accurate time and date. This IC provides the time in both 12 hour mode and 24 hour mode. This IC also provides calendar components day, month and year. This RTC uses internal lithium battery to keep the time and date updated when power is failed. This IC has 128 bytes of RAM memory. In these 128 bytes of RAM 14 bytes are used for time, date and registers. Remaining 114 bytes are used to store general purpose data.

The control registers of RTC are accessible only when the power is applied from external source. This IC requires more than 4.25V power supply and the control registers are accessible after 200ms when external power is supplied.

## 2.2 Circuit Diagram Digital Clock Circuit using 8051 and DS1307



**Figure-2.2:** Digital Clock using 8051 and DS1307 circuit diagram



**Figure-2.3:** Digital Clock using 8051 and DS1307



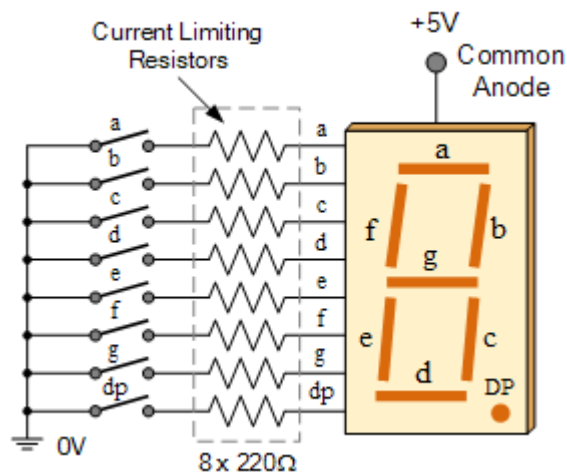
# Chapter 3

## Literature Review Parts & Components

### 3.1 Seven segment display

An LED or Light Emitting Diode, is a solid state optical pn-junction diode which emits light energy in the form of photons. The emission of these photons occurs when the diode junction is forward biased by an external voltage allowing current to flow across its junction, and in Electronics we call this process electroluminescence.

The actual color of the visible light emitted by an LED, ranging from blue to red to orange, is decided by the spectral wavelength of the emitted light which itself is dependent upon the mixture of the manifold impurities added to the semiconductor materials used to produce it.



**Figure-3.1:** 7-segment Display

The seven segment display, consists of seven LEDs arranged in a rectangular form as shown. Each of the seven LEDs is called a segment because when illuminated the segment forms part of a numerical digit (both Decimal and Hexadecimal) to be displayed. An additional 8th LED is sometimes used within the same package thus allowing the indication of a decimal point, (DP) when two or more 7-segment displays are connected together to display numbers greater than ten.

Each one of the seven LEDs in the display is given a positional segment with one of its connection pins being brought straight out of the rectangular plastic package. These individually LED pins are labelled from a through to g representing each individual LED. The other LED pins are connected together and wired to form a common pin.

So by forward biasing the appropriate pins of the LED segments in a particular order, some segments will be light and others will be dark allowing the desired character pattern of the number to be generated on the display. This then allows us to display each of the ten decimal digits 0 through to 9 on the same 7-segment display.

### **3.1.1 Advantages of light emitting diodes over traditional bulbs and lamps:**

- i) Being their small die size, several of them can be connected together within one small and compact package.
- ii) Long life,
- iii) Various colours,
- iv) Cheapness and are readily available,
- v) Being easy to interface with various other electronic components and digital circuits.

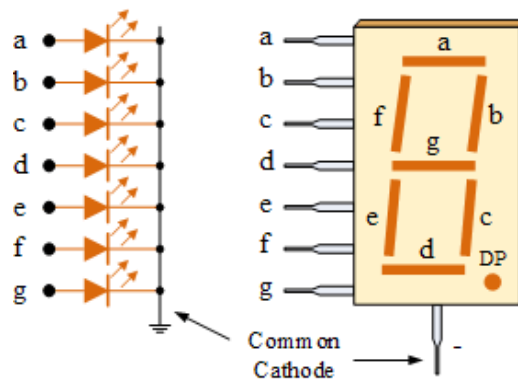
### 3.1.2 Classification of Seven segment

The displays common pin is generally used to identify which type of 7-segment display it is. As each LED has two connecting pins, one called the “Anode” and the other called the “Cathode”, there are consequently two types of LED 7-segment display called: **Common Cathode (CC)** and **Common Anode (CA)**.

### 3.1.3 Difference between two types segment display

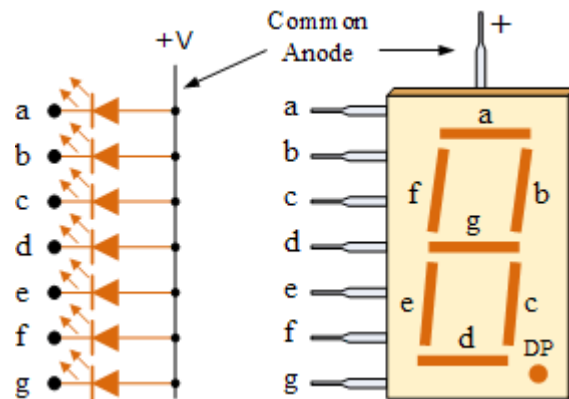
The difference between the two displays, as their name suggests, is that the common cathode has all the cathodes of the 7-segments connected directly together and the common anode has all the anodes of the 7-segments connected together and is illuminated as follows.

- i) **The Common Cathode (CC)** – In the common cathode display, all the cathode connections of the LED segments are joined together to logic “0” or ground. The individual segments are illuminated by application of a “HIGH”, or logic “1” signal via a current limiting resistor to forward bias the individual Anode terminals (a-g).



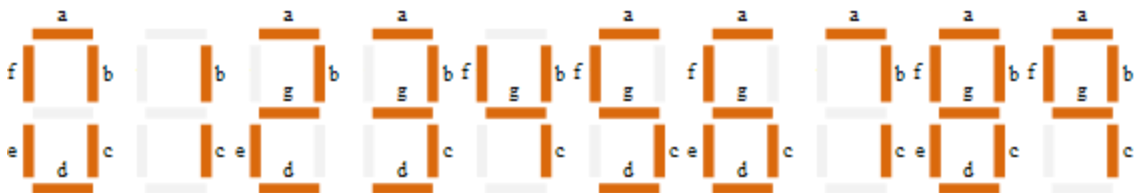
**Figure-3.2:** The Common Cathode (CC) – 7 segment display.

- ii) The Common Anode (CA) – In the common anode display, all the anode connections of the LED segments are joined together to logic “1”. The separate segments are illuminated by applying a ground, logic “0” or “LOW” signal via a suitable current limiting resistor to the Cathode of the particular segment (a-g).



**Figure-3.3:** The Common Anode (CA) – 7 segment display.

### 3.1.4 Seven Segment Display Segments for all Numbers.



Then for a 7-segment display, we can produce a truth table giving the individual segments that need to be illuminated in order to produce the required decimal digit from 0 through 9 as shown below.

### 3.1.5 Seven segment Display Truth Table

Decimal Digit	Individual Segment Illuminated						
	a	b	c	d	e	f	g
0	×	×	×	×	×	×	
1		×	×				
2	×	×		×	×		×
3	×	×	×	×			×
4		×	×			×	×
5	×		×	×		×	×
6	×		×	×	×	×	×
7	×	×	×				
8	×	×	×	×	×	×	×
9	×	×	×			×	×

**Table No: 3.1**

### 3.1.6 Driving a 7-segment Display

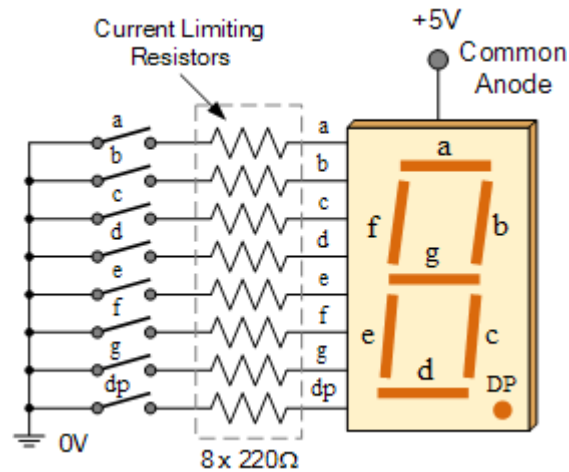
Although a 7-segment display can be thought of as a single display, it is still seven individual LEDs within a single package and as such these LEDs need protection from over current. LEDs produce light only when it is forward biased with the amount of light emitted being proportional to the forward current.

This means then that an LEDs light intensity increases in an approximately linear manner with an increasing current. So this forward current must be controlled and limited to a safe value by an external resistor to prevent damage to the LED segments.

The forward voltage drop across a red LED segment is very low at about 2-to-2.2 volts, (blue and white LEDs can be as high as 3.6 volts) so to illuminate accurately, the LED segments should be connected to a voltage source in excess of this forward voltage value with a series resistance used to limit the forward current to a desirable value.

Typically for a standard red colored 7-segment display, each LED segment can draw about 15 mA to illuminate correctly, so on a 5 volt digital logic circuit, the value of the current limiting resistor would be about  $200\Omega$  ( $5v - 2v$ )/15mA, or  $220\Omega$  to the nearest higher preferred value.

So to understand how the segments of the display are connected to a  $220\Omega$  current limiting resistor consider the circuit below



**Figure-3.4:** 7-segment Display

### 3.2 Liquid Crystal Displays(LCD)

The technology used for small computers and notebook. Like gas plasma technologies and led (light emitting diode). Lcd allow for display thinner than cathode ray tube technologies (crt). Lcds consume much less power than led and gas-display displays because they work on the principle of blocking light rather than emitting it. The lcd is made with either active matrix or passive matrix display grid. The active matrix display is also known as a thin film transistor (tft) display. Every intersection of passive matrix has grid of conductor with pixels. Current sent across two conductors on the grid to control light of any pixels. Every pixel intersection an active matrix has a transistor requiring less current to control the luminance of a pixel.

The current in an active display can be switched on and off more frequently improving the screen refresh time. Passive matrix lcds has dual scanning, meaning that they scan the grid twice with current in the same time that it took for me scan the original technology. However active matrix is still a superior technology.



**Figure-3.5:** LCD Display



### **3.2.1 LCD Working**

We always use devices made up of Liquid Crystal Displays (LCDs) like computers, digital watches and also DVD and CD players. They have become very common and have taken a giant leap in the screen industry by clearly replacing the use of Cathode Ray Tubes (CRT). CRT draws more power than LCD and are also bigger and heavier. LCD's have made displays thinner than CRT's. Even while comparing the LCD screen to an LED screen, the power consumption is lesser as it works on the basic principle of blocking light rather than dissipating. All of us have seen an LCD, but no one knows the exact working of it. Let us take a look at the working of an LCD.

### **3.2.2 Basics of LCD Displays**

The liquid-crystal display has the distinct advantage of having a low power consumption than the LED. It is typically of the order of microwatts for the display in comparison to the some order of mille watts for LEDs. Low power consumption requirement has made it compatible with MOS integrated logic circuit. Its other advantages are its low cost, and good contrast. The main drawbacks of **LCDs** are additional requirement of light source, a limited temperature range of operation (between 0 and 60° C), low reliability, short operating life, poor visibility in low ambient lighting, slow speed and the need for an ac drive.

### **3.2.3 Basic structure of an LCD**

A liquid crystal cell consists of a thin layer (about 10 u m) of a liquid crystal sandwiched between two glass sheets with transparent electrodes deposited on their inside faces. With both glass sheets transparent, the cell is known as transmittive type cell. When one glass is transparent and the other has a reflective coating, the cell is called reflective type. The LCD does not produce any illumination of its own. It, in fact, depends entirely on illumination falling on it from an external source for its visual effect.

### **3.2.4 Types of LCD/Liquid Crystal Displays.**

**Two types of display available are dynamic scattering display and field effect display.**

- i) When dynamic scattering display is energized, the molecules of energized area of the display become turbulent and scatter light in all directions. Consequently, the activated areas take on a frosted glass appearance resulting in a silver display. Of course, the un energized areas remain translucent.
- ii) Field effect LCD contains front and back polarizers at right angles to each other. Without electrical excitation, the light coming through the front polarizer is rotated  $90^\circ$  in the fluid.

### **3.2.5 Nematic Phase LCD**

The greatest advantage of a nematic phase liquid crystal substance is that it can bring about predictable controlled changes according to the electric current passed through them. All the liquid crystals are according to their reaction on temperature difference and also the nature of the substance.

Twisted nematic, a particular nematic substance is twisted naturally. When a known voltage is applied to the substance, it gets untwisted in varying degrees according to our requirement. This in turn is useful in controlling the passage of light. A nematic phase liquid crystal can be again classified on the basis in which the molecules orient themselves in respect to each other. This change in orientation mainly depends on the director, which can be anything ranging from a magnetic field to a surface with microscopic grooves. Classification includes Nematic and also cholesteric. Nematic can be again classified as nematic C, in which the molecules in each layer tilt at an angle from the previous layer. Cholesteric, on the other hand has molecules that twist slightly from one layer to the next, causing a spiral like design. There are also combinations of these two called Ferro-electric liquid crystals (FLC), which include cholesteric molecules in a nematic C type molecule so

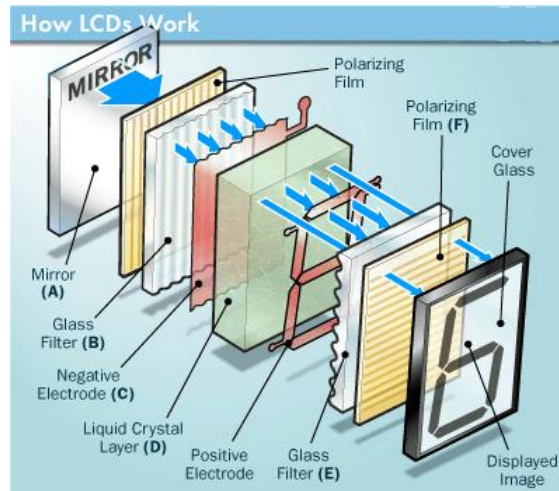
that the spiral nature of these molecules allows the microsecond switching response time. This makes FLCs to be of good use in advanced displays.

Liquid crystal molecules are further classified into thermochroic and lyotropic crystals. The former changes proportionally with respect to changes in pressure and temperature. They are further divided into smatic and isotropic. Sematic liquid crystals have a fixed order of pattern while isotropic liquid crystals are distributed randomly. The lyotropic crystal depends on the type of solvent they are mixed with. They are therefore useful in making detergents and soaps.

### 3.2.6 Making of LCD

- Though the making of LCD is rather simple there are certain facts that should be noted while making it.
- The basic structure of an LCD should be controllably changed with respect to the applied electric current.
- The light that is used on the LCD can be polarized.
- Liquid crystals should be able to both transmit and change polarized light.
- There are transparent substances that can conduct electricity.

To make an LCD, you need to take two polarized glass pieces. The glass which does not have a polarized film on it must be rubbed with a special polymer which creates microscopic grooves in the surface. It must also be noted that the grooves are on the same direction as the polarizing film. Then, all you need to do is to add a coating of smectic liquid crystals to one of the filters. The grooves will cause the first layer of molecules to align with the filter's orientation. At right angle to the first piece, you must then add a second piece of glass along with the polarizing film. Till the uppermost layer is at a 90-degree angle to the bottom, each successive layer of TN molecules will keep on twisting. The first filter will naturally be polarized as the light strikes it at the beginning. Thus the light passes through each layer and is guided on to the next with the help of molecules. When this happens, the molecules tend to change the plane of vibration of the light to match their own angle. When the light reaches the far side of the liquid crystal substance, it vibrates at the same angle as the final layer of molecules. The light is only allowed an entrance if the second polarized glass filter is same as the final layer. Take a look at the figure below.



**Figure-3.6:** working of Lcd

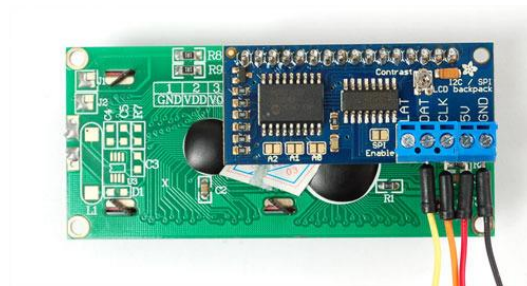
The main principle behind liquid crystal molecules is that when an electric current is applied to them, they tend to untwist. This causes a change in the light angle passing through them. This causes a change in the angle of the top polarizing filter with respect to it. So little light is allowed to pass through that particular area of LCD. Thus that area becomes darker comparing to others.

For making an LCD screen, a reflective mirror has to be setup in the back. An electrode plane made of indium-tin oxide is kept on top and a glass with a polarizing film is also added on the bottom side. The entire area of the LCD has to be covered by a common electrode and above it should be the liquid crystal substance. Next comes another piece of glass with an electrode in the shape of the rectangle on the bottom and, on top, another polarizing film. It must be noted that both of them are kept at right angles. When there is no current, the light passes through the front of the LCD it will be reflected by the mirror and bounced back. As the electrode is connected to a temporary battery the current from it will cause the liquid crystals between the common-plane electrode and the electrode shaped like a rectangle to untwist. Thus the light is blocked from passing through. Thus that particular rectangular area appears blank.

### 3.3 I2C\_LCD Module

I2C Lcd is an easy-to-use display module, It can make display easier. Using it can reduce the difficulty of make, so that makers can focus on the core of the work. We developed the Arduino library for I2C\_LCD, user just need a few lines of the code can achieve complex graphics and text display features. It can replace the serial monitor of Arduino in some place, you can get running information's without a computer.

More than that, we also develop the dedicated picture data convert software (bitmap converter) now is available to support PC platform of windows, Linux, Mac OS. Through the bitmap convert software you can get your favorite picture displayed on I2C\_LCD, without the need for complex programming. I2C\_LCD provided with a very convenient way of make.



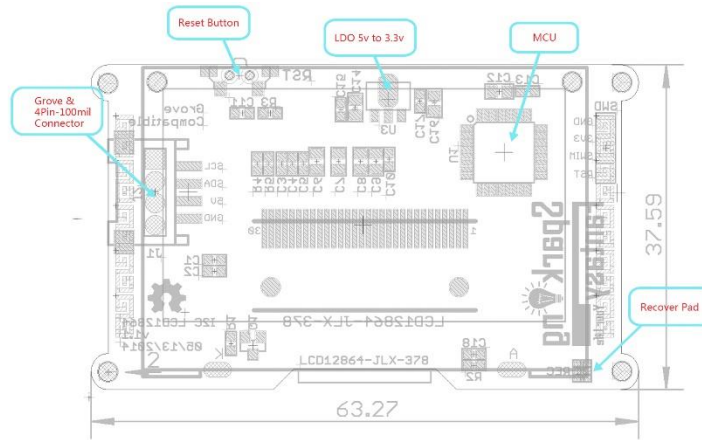
**Figure-3.7:** LCD with I<sub>2</sub>C Module

#### 3.3.1 Features of LCD with I<sub>2</sub>C Module

- Only 2 Arduino pins are occupied (Use I2C interface).
- Supports standard I2C mode (100Kbit/s) and fast I2C mode (400Kbit/s).
- Compatible with multiple communication logic levels: 2.8~5VDC.
- Arduino library supported, use a line of code to complete the display.
- Integrate 7 sizes of ASCII fonts, 5 graphics functions.

- Provide dedicated picture data convert software (Bitmap Converter).
- Most of the complex operation is processed by I2C\_LCD independent controller, saving user controller resources.
- Supports cursor function, can set up 16 cursor flicker frequency.
- Supports 128 level backlight lightness adjustment.
- Support 64 level screen contrast adjustment.
- Support device address modification.
- Supports 127 I2C\_LCD work in parallel.
- When debugging code, it can take the place of the serial monitor to monitor the program running state.
- Two abnormal recovery methods are provided: reset and restore the factory settings.
- Compatible with Grove interface and 4Pin-100mil interface (under the Grove socket).
- 4 symmetrical fixed hole design for easy user installation.
- China style unique appearance.

### 3.3.2 Interface Function



**Figure-3.8:** Interface Function of LCD with I<sub>2</sub>C Module



### 3.4 ATmega328P Microcontroller IC

ATMEGA328P is high performance, low power controller from Microchip.

ATMEGA328P is an 8-bit microcontroller based on AVR RISC architecture.

It is the most popular of all AVR controllers as it has used in PCB boards.

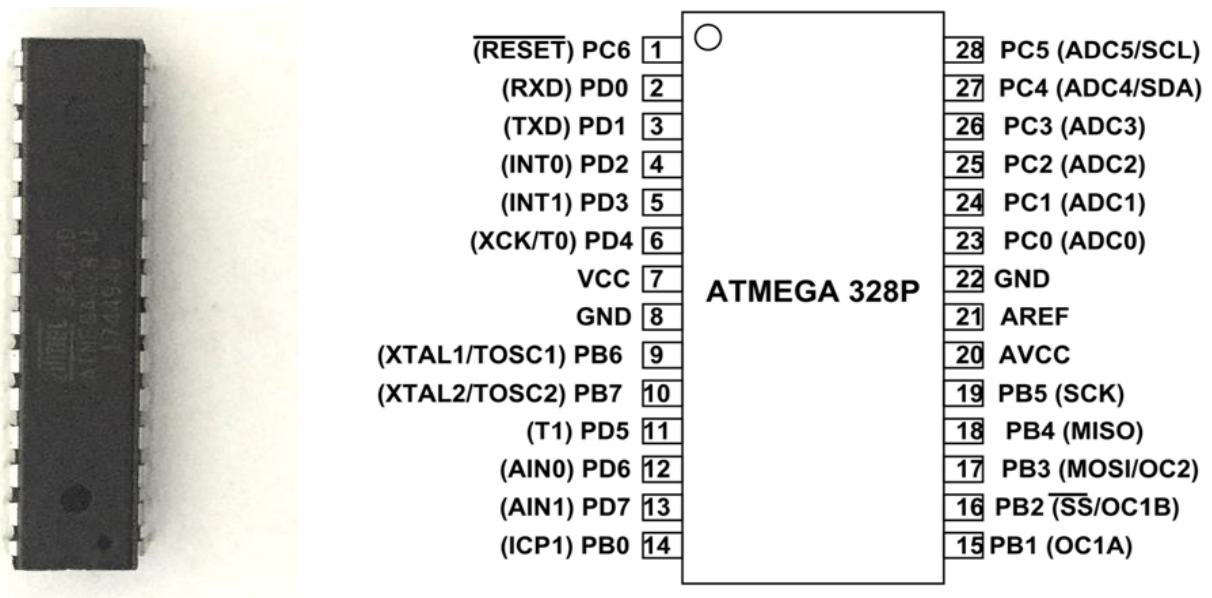


Figure 3.9 (a): Atmega328p Ic

Figure 3.9 (b): Pin Names

ATmega 328p has 1KB Electrically Erasable Programmable Read Only Memory (EEPROM). This property shows if the electric supply supplied to the micro-controller is removed, even then it can store the data and can provide results after providing it with the electric supply. Moreover, ATmega-328 has 2KB Static Random Access Memory (SRAM). ATmega 328p has several different features, which make it the most popular

device in today's market. These features consist of advanced RISC architecture, good performance, low power consumption, real timer counter having separate oscillator, 6 PWM pins, programmable Serial USART, programming lock for software security, throughput up to 20 MIPS etc. ATmega-328 is mostly used in Arduino.

### 3.4.1 ATmega328 Pin Configuration

Pin No.	Pin name	Description	
1	PC6 (RESET)	Pin6 of PORTC	Pin by default is used as RESET pin. PC6 can only be used as I/O pin when RSTDISBL Fuse has programmed.
2	PD0 (RXD)	Pin0 of PORTD	RXD (Data Input Pin for USART) USART Serial Communication Interface [Can be used for programming]
3	PD1 (TXD)	Pin1 of PORTD	TXD (Data Output Pin for USART) USART Serial Communication Interface [Can be used for programming] INT2( External Interrupt 2 Input)
4	PD2 (INT0)	Pin2 of PORTD	External Interrupt source 0

5	PD3 (INT1/OC2B)	Pin3 of PORTD	External Interrupt source1 OC2B (PWM - Timer/Counter2 Output Compare Match B Output)
6	PD4 (XCK/T0)	Pin4 of PORTD	T0( Timer0 External Counter Input) XCK ( USART External Clock I/O)
7	VCC		Connected to positive voltage
8	GND		Connected to ground
9	PB6 (XTAL1/TOSC1)	Pin6 of PORTB	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1)
10	PB7 (XTAL2/TOSC2)	Pin7 of PORTB	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2)
11	PD5 (T1/OC0B)	Pin5 of PORTD	
12	PD6 (AIN0/OC0A)	Pin6 of PORTD	AIN0(Analog Comparator Positive I/P) OC0A(PWM - Timer/Counter0 Output Compare Match A Output)
13	PD7 (AIN1)	Pin7 of PORTD	AIN1(Analog Comparator Negative I/P)
14	PB0 (ICP1/CLKO)	Pin0 of PORTB	ICP1(Timer/Counter1 Input Capture Pin) CLKO (Divided System Clock. The divided system clock can be output on the PB0 pin)

15	PB1 (OC1A)	Pin1 of PORTB	OC1A (Timer/Counter1 Output Compare Match A Output)
16	PB2 (SS/OC1B)	Pin2 of PORTB	SS (SPI Slave Select Input). This pin is low when controller acts as slave. [Serial Peripheral Interface (SPI) for programming] OC1B (Timer/Counter1 Output Compare Match B Output)
17	PB3 (MOSI/O C2A)	Pin3 of PORTB	MOSI (Master Output Slave Input). When controller acts as slave, the data is received by this pin. [Serial Peripheral Interface (SPI) for programming] OC2 (Timer/Counter2 Output Compare Match Output)
18	PB4 (MISO)	Pin4 of PORTB	MISO (Master Input Slave Output). When controller acts as slave, the data is sent to master by this controller through this pin. [Serial Peripheral Interface (SPI) for programming]
19	PB5 (SCK)	Pin5 of PORTB	SCK (SPI Bus Serial Clock). This is the clock shared between this controller and other system for accurate data transfer.
20	AVCC		
21	AREF		Analog Reference Pin for ADC
22	GND		GROUND

23	PC0 (ADC0)	Pin0 of PORTC	ADC0 (ADC Input Channel 0)
24	PC1 (ADC1)	Pin1 of PORTC	ADC1 (ADC Input Channel 1)
25	PC2 (ADC2)	Pin2 of PORTC	ADC2 (ADC Input Channel 2)
26	PC3 (ADC3)	Pin3 of PORTC	ADC3 (ADC Input Channel 3)
27	PC4 (ADC4/S DA)	Pin4 of PORTC	ADC4 (ADC Input Channel 4) SDA (Two-wire Serial Bus Data Input/output Line)
28	PC5 (ADC5/S CL)	Pin5 of PORTC	ADC5 (ADC Input Channel 5) SCL (Two-wire Serial Bus Clock Line)

**Table No: 3.2**

### **3.5 HC-05 Bluetooth Module**

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.



**Figure-3.10:** HC-05 Bluetooth Module

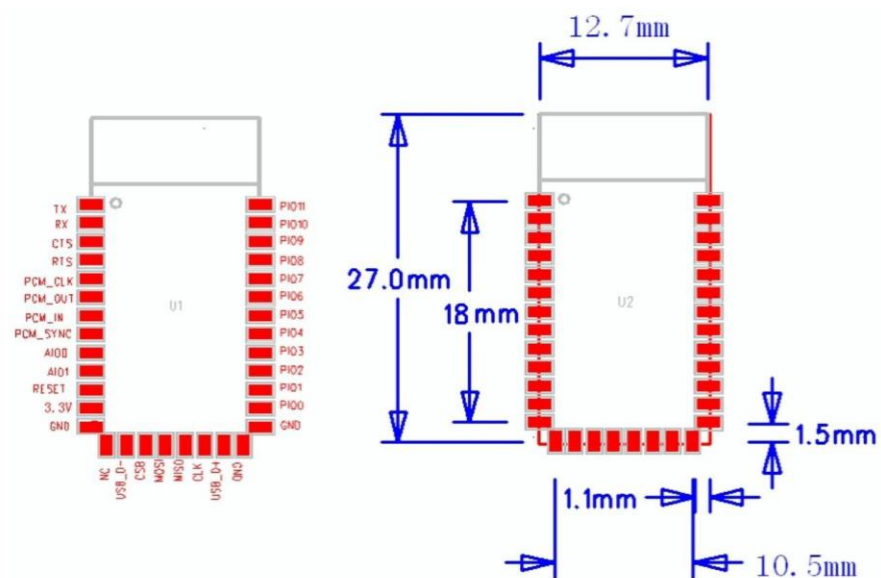
### **3.5.1 Hardware features**

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

### 3.5.2 Software features

- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has. Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:”0000” as default
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection

### 2.4.3 Hardware Specifications



**Figure-3.11:** Hardware Specification of HC-05

### 3.5.4 Pin Configuration

Pin Number	Pin Name	Description
1	Enable / Key	This pin is used to toggle between Data Mode (set low) and AT command mode (set high). By default it is in Data mode
2	Vcc	Powers the module. Connect to +5V Supply voltage
3	Ground	Ground pin of module, connect to system ground.
4	TX Transmitter	Transmits Serial Data. Everything received via Bluetooth will be given out by this pin as serial data.
5	RX – Receiver	Receive Serial Data. Every serial data given to this pin will be broadcasted via Bluetooth
6	State	The state pin is connected to on board LED, it can be used as a feedback to check if Bluetooth is working properly.
7	LED	Indicates the status of Module



		<ul style="list-style-type: none"> <li>• Blink once in 2 sec: Module has entered Command Mode</li> <li>• Repeated Blinking: Waiting for connection in Data Mode</li> <li>• Blink twice in 1 sec: Connection successful in Data Mode</li> </ul>
8	Button	Used to control the Key/Enable pin to toggle between Data and command Mode

**Table No: 3.2**

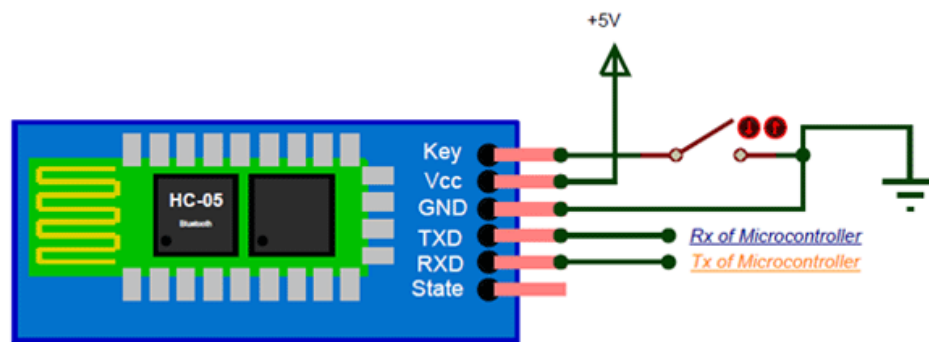
### **3.5.5 HC-05 Technical Specifications**

- Serial Bluetooth module for Arduino and other microcontrollers
- Operating Voltage: 4V to 6V (Typically +5V)
- Operating Current: 30mA
- Range: <100m
- Works with Serial communication (USART) and TTL compatible
- Follows IEEE 802.15.1 standardized protocol
- Uses Frequency-Hopping Spread spectrum (FHSS)
- Can operate in Master, Slave or Master/Slave mode
- Can be easily interfaced with Laptop or Mobile phones with Bluetooth
- Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.

### 3.5.6 How to Use the HC-05 Bluetooth module

The **HC-05** has two operating modes, one is the Data mode in which it can send and receive data from other Bluetooth devices and the other is the AT Command mode where the default device settings can be changed. We can operate the device in either of these two modes by using the key pin as explained in the pin description.

It is very easy to pair the HC-05 module with microcontrollers because it operates using the Serial Port Protocol (SPP). Simply power the module with +5V and connect the Rx pin of the module to the Tx of MCU and Tx pin of module to Rx of MCU as shown in the figure below 2.12



**Figure-3.12:** HC-05 Connection

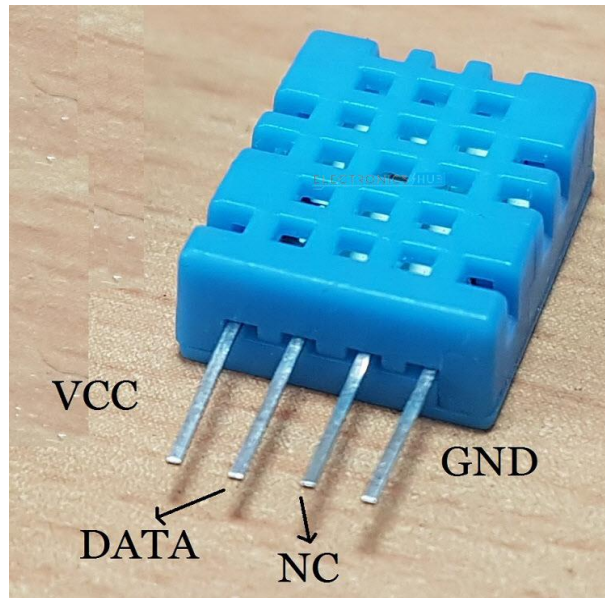
During power up the key pin can be grounded to enter into Command mode, if left free it will by default enter into the data mode. As soon as the module is powered you should be able to discover the Bluetooth device as “HC-05” then connect with it using the default password 1234 and start communicating with it. The name password and other default parameters can be changed.

### **3.5.7 Applications of HC-05 Bluetooth Module**

1. Wireless communication between two microcontrollers
2. Communicate with Laptop, Desktops and mobile phones
3. Data Logging application
4. Consumer applications
5. Wireless Robots
6. Home Automation

### 3.6 DHT11 basic temperature-humidity sensor

The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). Its fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using our library, sensor readings can be up to 2 seconds old.

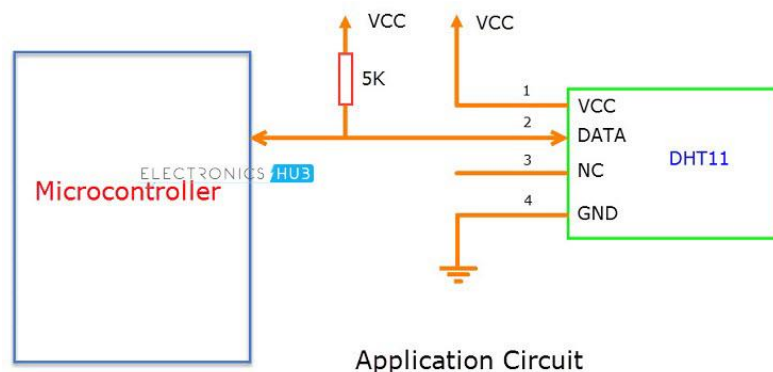


**Figure-3.13:** DHT11 temperature-humidity sensor



**Figure-3.14:** DHT11 temperature-humidity sensor with module

Each DHT11 sensors features extremely accurate calibration of humidity calibration chamber. The calibration coefficients stored in the OTP program memory, internal sensors detect signals in the process, we should call these calibration coefficients. The single-wire serial interface system is integrated to become quick and easy. Small size, low power, signal transmission distance up to 20 meters, enabling a variety of applications and even the most demanding ones. The product is 4-pin single row pin package. Convenient connection, special packages can be provided according to users need.



**Figure-3.15:** DHT11 temperature-humidity sensor circuit

### **3.6.1 Specification**

- Supply Voltage: +5 V
- Temperature range :0-50 °C error of  $\pm 2$  °C
- Humidity :20-90% RH  $\pm 5\%$  RH error
- Interface: Digital

### 3.7 CD4511 7-Segment Driver IC

This IC combines the low quiescent power dissipation and high noise immunity feature of CMOS with NPN bipolar transistor, which is capable of giving output current of 25mA. Mainly this IC is used for lighting up or driving various types of display like 7-segment, low voltage fluorescent display and incandescent display. But this IC is mainly used for driving the 7-segment display.

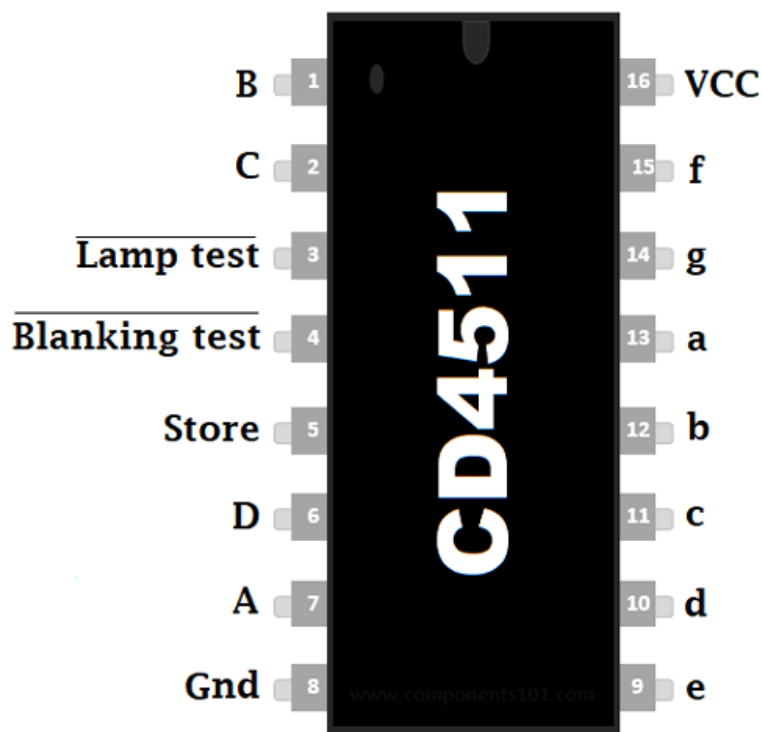


Figure-3.16: CD4511 7-Segment Driver IC

### 3.7.1 Features

- Lamping test and blanking capability.
- Low logic circuit power dissipation.
- Input latches for BCD code storage.
- Balanced propagation delays and transition time.
- High output sourcing capability.
- Lamp intensity modulation capability.
- Rigid structure.



### 3.7.2 Pin configuration

Pin no.	Pin name	Description
1,2,6,7	B,C,D,A	BCD input of the IC
3	Display test/Lamp test	To test the display LEDs
4	Blank input	To turn-off the LEDs of the display
5	Store	Store or strobe a BCD code
8	Gnd	Ground
9,10,11,12,13,14,15	e,d,c,b,a,g,f	7-segment outputs
16	Vcc	Positive supply input

**Table No: 3.3**

### 3.7.3 Technical Specification

- Supply voltage ranges from 3v to 18v
- Set-up-time is 150, 70, and 40ns at 5v, 10v, and 15v respectively
- Strobe pulse width is 400, 160, and 100ns at 5v, 10v, and 15v respectively
- Hold time is zero

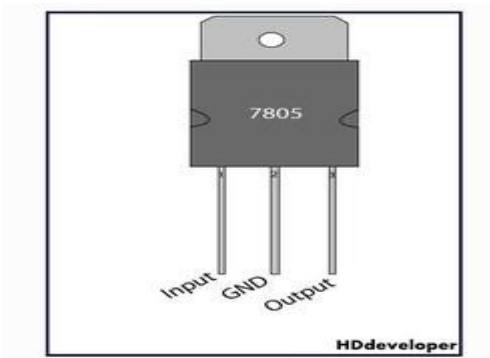
- 100% tested for quiescent current at 20v
- Maximum output current sourcing of 25mA
- Operating temperature ranges from -40 °C to 85 °C
- Storage temperature ranges from -65 °C to 150 °C

### 3.7.4 CD4511 Equivalent ICs

74LS47, 7446, 7447, SN5446A, SN5447A, SN5448, SN54LS48, SN54LS47, SN74LS48, SN74LS49, SN54LS49, SN7448

## 3.8 Voltage Regulator 7805 IC

Voltage sources in a circuit may have fluctuations resulting in not providing fixed voltage outputs. A voltage regulator IC maintains the output voltage at a constant value. 7805 IC, a member of 78xx series of fixed linear voltage regulators used to maintain such fluctuations, is a popular voltage regulator integrated circuit (IC). The xx in 78xx indicates the output voltage it provides. 7805 IC provides +5 volts regulated power supply with provisions to add a heat sink.



**Figure-3.17:** Voltage Regulator 7805 IC

In the below table 3.4 shown, there is a significant difference between the input voltage & the output voltage of the voltage regulator. This difference between the input and output voltage is released as heat. The greater the difference between the input and output voltage, more the heat generated. If the regulator does not have a heat sink to dissipate this heat, it can get destroyed and malfunction. Hence, it is advisable to limit the voltage to a maximum of 2-3 volts above the output voltage. So, we now have 2 options. Either design your circuit so that the input voltage going into the regulator is limited to 2-3 volts above the output regulated voltage or place an appropriate heatsink, that can efficiently dissipate heat.

### 3.8.1 Pin Details of 7805 IC

Pin No.	Pin	Function	Description
1	INPUT	Input voltage (7V-35V)	In this pin of the IC positive unregulated voltage is given in regulation.
2	GROUND	Ground(0V)	In this pin where the ground is given. This pin is neutral for equally the input and output.
3	OUTPUT	Regulated output; 5V (4.8V-5.2V)	The output of the regulated 5V volt has taken out at this pin of the IC regulator.

**Table No: 3.4**

### 3.8.2 Application areas for 7805 IC

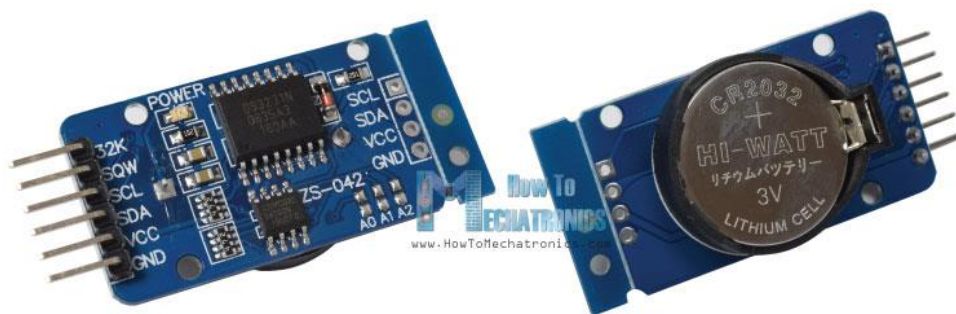
7805 IC is used in a wide range of circuits. The major ones being:

- Fixed-Output Regulator
- Positive Regulator in Negative Configuration
- Adjustable Output Regulator
- Current Regulator
- Adjustable DC Voltage Regulator
- Regulated Dual-Supply
- Output Polarity-Reversal-Protection Circuit
- Reverse bias projection Circuit.

### 3.9 RTC (Real Time Clock)

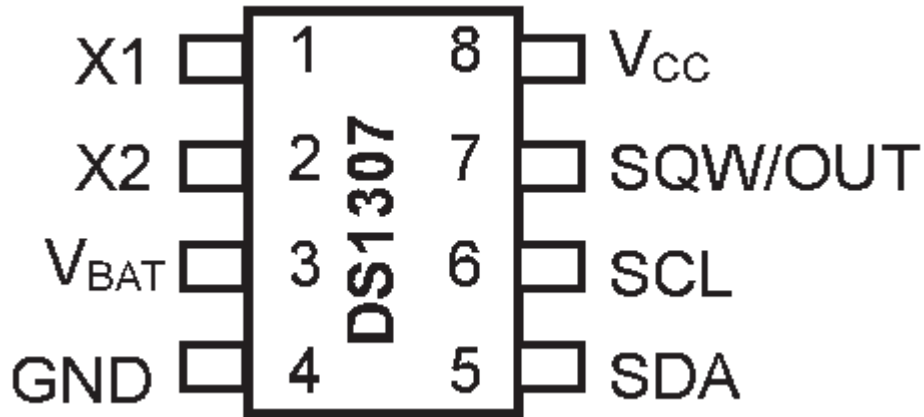
Real time clocks (RTC), as the name recommends are clock modules. The DS1307 real time clock (RTC) IC is an 8 pin device using an I2C interface. The DS1307 is a low-power clock/calendar with 56 bytes of battery backup SRAM. The clock/calendar provides seconds, minutes, hours, day, date, month and year qualified data. The end date of each month is automatically adjusted, especially for months with less than 31 days.

They are available as integrated circuits (ICs) and supervise timing like a clock and also operate date like a calendar. The main advantage of RTC is that they have an arrangement of battery backup which keeps the clock/calendar running even if there is power failure. An exceptionally little current is required for keeping the RTC animated. We can find these RTCs in many applications like embedded systems and computer mother boards, etc. In this article we are going to see about one of the real time clock (RTC), i.e. DS1307.



**Figure-3.18:** Real time clock

### 3.9.1 Pin Description of DS1307



**Figure-3.19:** Pin description of DS1307

**Pin 1, 2:** Connections for standard 32.768 kHz quartz crystal. The internal oscillator circuitry is intended for operation with a crystal having a specified load capacitance of 12.5pF. X1 is the input to the oscillator and can alternatively be connected to an external 32.768 kHz oscillator. The output of the internal oscillator, X2 is drifted if an external oscillator is connected to X1.

**Pin 3:** Battery input for any standard 3V lithium cell or other energy source. Battery voltage should be between 2V and 3.5V for suitable operation. The nominal write protect trip point voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V<sub>BAT</sub> nominal. A lithium battery with 48mAh or greater will backup the DS1307 for more than 10 years in the absence of power at 25°C. UL recognized to ensure against reverse charging current when utilized as a part of conjunction with a lithium battery.

**Pin 4:** Ground.

**Pin 5:** Serial data input/output. The input/output for the I2C serial interface is the SDA, which is open drain and requires a pull up resistor, allowing a pull up voltage upto 5.5V. Regardless of the voltage on VCC.

**Pin 6:** Serial clock input. It is the I2C interface clock input and is used in data synchronization.

**Pin 7:** Square wave/output driver. When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square-wave frequencies (1Hz, 4 kHz, 8 kHz, and 32 kHz). This is also open drain and requires an external pull-up resistor. It requires application of either Vcc or Vb at to operate SQW/OUT, with an allowable pull up voltage of 5.5V and can be left floating, if not used.

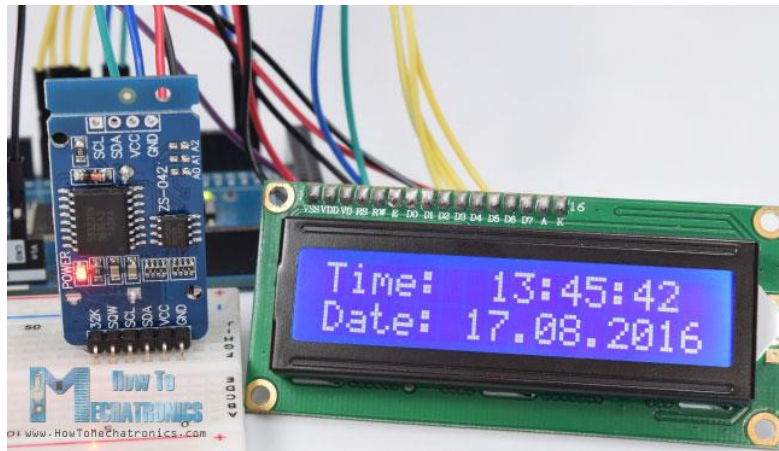
**Pin 8:** Primary power supply. When voltage is applied within normal limits, the device is fully accessible and data can be written and read. When a backup supply is connected to the device and VCC is below VTP, read and writes are inhibited. However at low voltages, the timekeeping function still functions.

### **3.9.2 Working Procedure of RTC Module**

There are two ways to work with the module. You can either use an Arduino library created by Rhys Weatherley, or use raw I2C commands. If you are new to the world of Arduino, using the library will be simpler. However if you enjoy working with I2C, wish to explore the DS3232 to the fullest, or you're using a different development platform - the I2C commands will be necessary.

There may be a small piece of plastic fitted between the battery and the holder, please remove this when you're ready to use your RTC module.

Now let's get started we will first demonstrate using the library which is simpler and allows you a fast start, and then examine the useful direct I2C commands which give you complete control over the real-time clock module.



**Fig 3.20:** RTCI Module

### 3.9.3 Features

- Programmable square wave output signal
- Automatic power-fail detect and switch circuitry
- Consumes less than 500nA in battery backup mode with oscillator running
- Available in 8-pin DIP or SOIC
- Underwriters Laboratory (UL) recognized
- Real-time clock (RTC) counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap-year compensation valid up to 2100
- 56-byte non-volatile RAM for data storage



### 3.10 Crystal Oscillator

A crystal oscillator is an electronic oscillator circuit which is used for the mechanical resonance of a vibrating crystal of piezoelectric material. It will create an electrical signal with a given frequency. This frequency is commonly used to keep track of time for example: wrist watches are used in digital integrated circuits to provide a stable clock signal and also used to stabilize frequencies for radio transmitters and receivers. Quartz crystal is mainly used in radio-frequency (RF) oscillators. Quartz crystal is the most common type of piezoelectric resonator, in oscillator circuits we are using them so it became known as crystal oscillators. Crystal oscillators must be designed to provide a load capacitance.

There are different types of oscillator electronic circuits which are in use they are namely: Linear oscillators – Hartley oscillator, Phase-shift oscillator, Armstrong oscillator, Clapp oscillator, Colpitts oscillator. Relaxation oscillators – Royer oscillator, Ring oscillator, Multivibrator and Voltage Controlled Oscillator (VCO). Soon we are going to discuss in detail about crystal oscillator like, working and applications of a crystal oscillator.



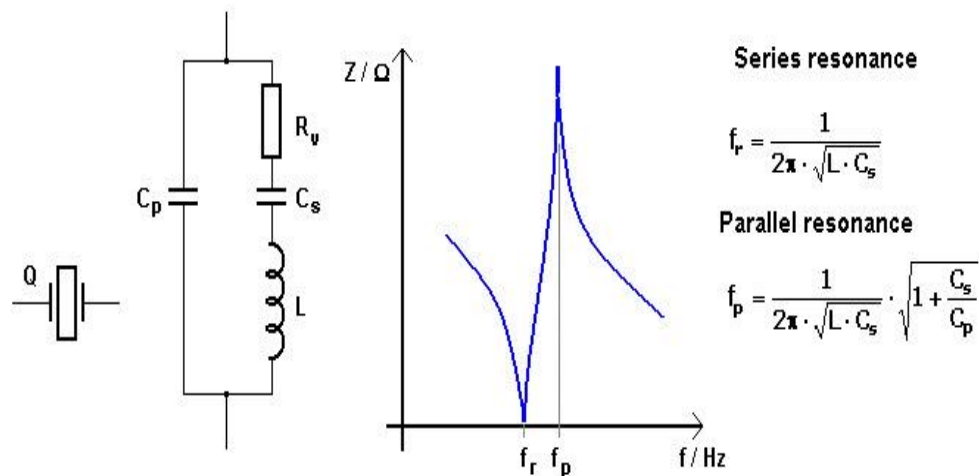
**Figure 3.21:** Crystal Oscillator

### 3.10.1 Crystal Oscillator Working

Crystal oscillator circuit usually works on the principle of the inverse piezoelectric effect. The applied electric field will produce a mechanical deformation across some materials. Thus, it utilizes the vibrating crystal's mechanical resonance, that is made with a piezoelectric material for generating an electrical signal of a particular frequency.

Usually quartz crystal oscillators are highly stable, consists of good quality factor(Q), they are small in size, and are economically related. Hence, quartz crystal oscillator circuits are more superior compared to other resonators like LC circuits, turning forks. Generally in Microprocessors and Micro controllers we are using an 8MHz crystal oscillator.

The equivalent electrical circuit is also describes the crystal action of the crystal. Just look at the equivalent electrical circuit diagram shown in the above. The basic components used in the circuit, inductance L represents crystal mass, capacitance C2 represents compliance, and C1 is used to represent the capacitance that is formed because of crystal's mechanical moulding, resistance R represents the crystal's internal structure friction, The quartz crystal oscillator circuit diagram consists of two resonances such as series and parallel resonance, i.e., two resonant frequencies.



**Figure 3.22:** Crystal Oscillator Working

The series resonance occurs when the reactance produced by capacitance  $C$  is equal and opposite to the reactance produced by inductance  $L$ . The  $f_r$  and  $f_p$  represents series and parallel resonant frequencies respectively, and the values of ' $f_r$ ' and ' $f_p$ ' can be determined by using the following equations shown in the figure below.

The above diagram describes an equivalent circuit, plot graph for Resonant frequency, Formulae for Resonant frequencies.

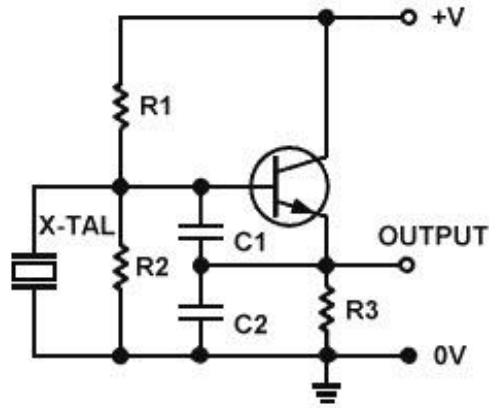
### **3.10.2 Uses of Crystal Oscillator**

In general, we know that, in the design of microprocessors and microcontrollers, crystal oscillators are used for the sake of providing the clock signals. For instance, let us consider 8051 microcontroller, in this particular controller an external crystal oscillator circuit will work with 12MHz that is essential, even though this 8051 microcontroller (based on model) is capable to work at 40 MHz (max) have to provide 12MHz in most of the cases because for a machine cycle 8051 requires 12 clock cycles, so that to give effective cycle rate at 1MHz (taking 12MHz clock) to 3.33MHz (taking the maximum 40MHz clock). This particular crystal oscillator which is having cycle rate at 1MHz to 3.33MHz is used to generate clock pulses which are required for the synchronization of all the internal operations.

### **3.10.3 Application of Crystal Oscillator**

There are various applications for crystal oscillator in various fields and some of crystal oscillator applications are given below.

Colpitts Oscillator is used to generate a sinusoidal output signal at very high frequencies. This oscillator can be used as the different types of sensors such as temperature sensor Because of SAW device which we are using in Colpitts circuit it senses directly from its surface.



**Fig-3.23:** Colpitts Crystal Oscillator

The applications of the Colpitts oscillators mainly involve where the wide range of frequencies is used. Also used in undamped and continuous oscillation condition. Using some devices in Colpitts circuit, we can achieve greater temperature stability and high frequency.

Colpitts used for the development of mobile communications and radio communication

### **3.11 Summary of this chapter**

Those above parts have been used in our project directly. Some alternatives can be used of above components. But these are very much available in market.

# Chapter 4

## Methodology: Circuit Design and Development

## 4.1 Design and Development

This project creates a clock which displays the time on seven segment displays. The project can befits in a small box. The circuit uses seven outputs and four common connections (com1, com2, com3, com4) for the segments. To allow the setting of the time two buttons are provided. The first selects the digit and the second increments the digits. The order is hours, minutes, second. When the setting mode is active only the digits being set are displayed, all the others are shown as '--'.

Before creating the program we listed the steps the program requires:

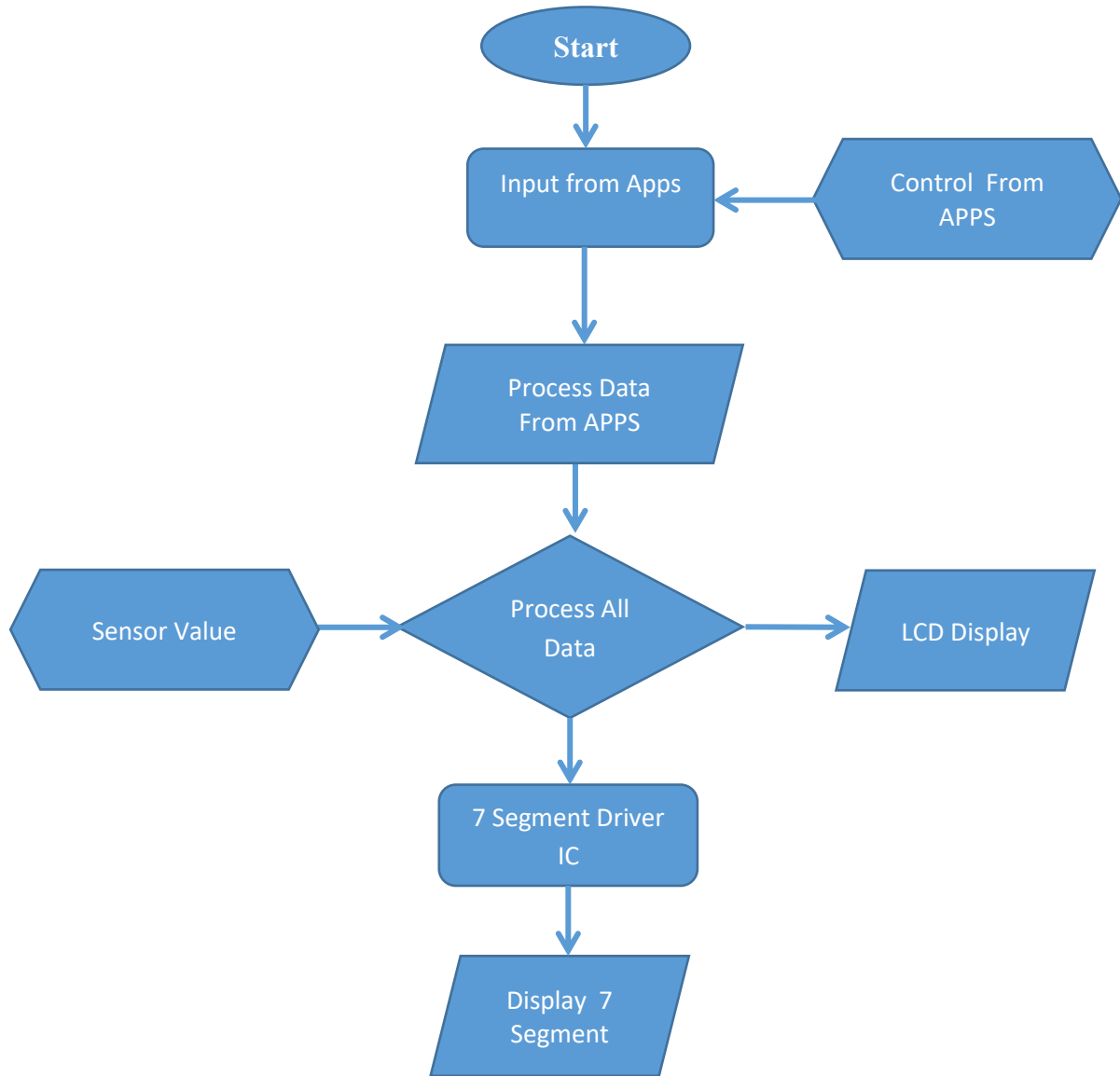
- Initialize the PIC and set all digits to zero.
- Start the timer interrupts to count the seconds elapsed.
- When seconds reach 60 increment minutes and reset seconds.
- When minutes reach 60 increment hours and reset minutes.
- When hours reach 12 increments and reset hours.
- When the first button is pressed stop updating the clock, reset all digits to – apart from the hours. If the button is pressed again set hours to – and display the minutes. Each press moves through the digits.
- After the first button has been pressed the second button increments the number on the digits. When the digit reaches its intended maximum it returns back to its lowest value. For example, hours range from 00 to 12. When the digit increments to 12 it returns to 00.

First version only in the program and this is reflected in the flow charts below. Based on these steps we created a flow diagram to help create the program. The Main Flow Diagram constitutes the main sub program. After setting up the ports and starting the interrupt the main sub program polls the buttons.

The interrupt flow diagram now shows only the seconds being updated. Not shown in the flow charts, code is also required to display the digits and to read the inputs. We found in use that the clock gained time as the day went on. After a few days the time difference was noticeable. I calculated that every day 16 seconds were being gained. I decided to setup a new counter that was incremented each second and it would be used to lose a second at the correct moment. I needed to lose 1 second every  $24 / 16 = 1.5$  hours. This was 5400 seconds ( $1.5 \times 60 \times 60$ ). When the counter reached 5400 seconds the seconds counter was decreased by 1 and the counter was reset. The clock now keeps very good time. Outputs are used to turn the 7 segment and decimal points on and off. The displays are common four anodes so a port set low turns the segment on and a port set high turns it off. Each of the digits is turned on sequentially. The segments for that digit are only on while that digit is on.



## 4.2 Flowchart



**Fig-4.1:** Flowchart

### 4.3 Diagram of Seven Segment Clock Ckt

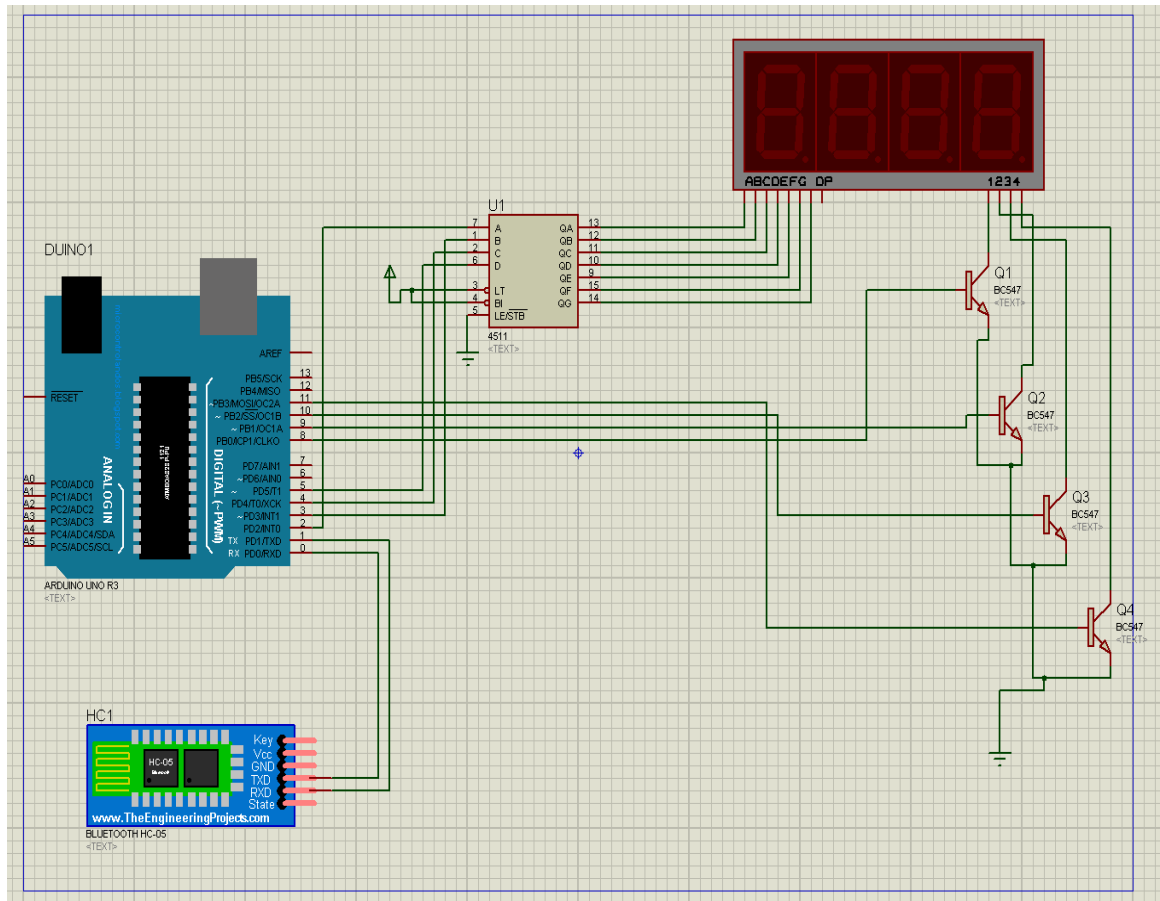
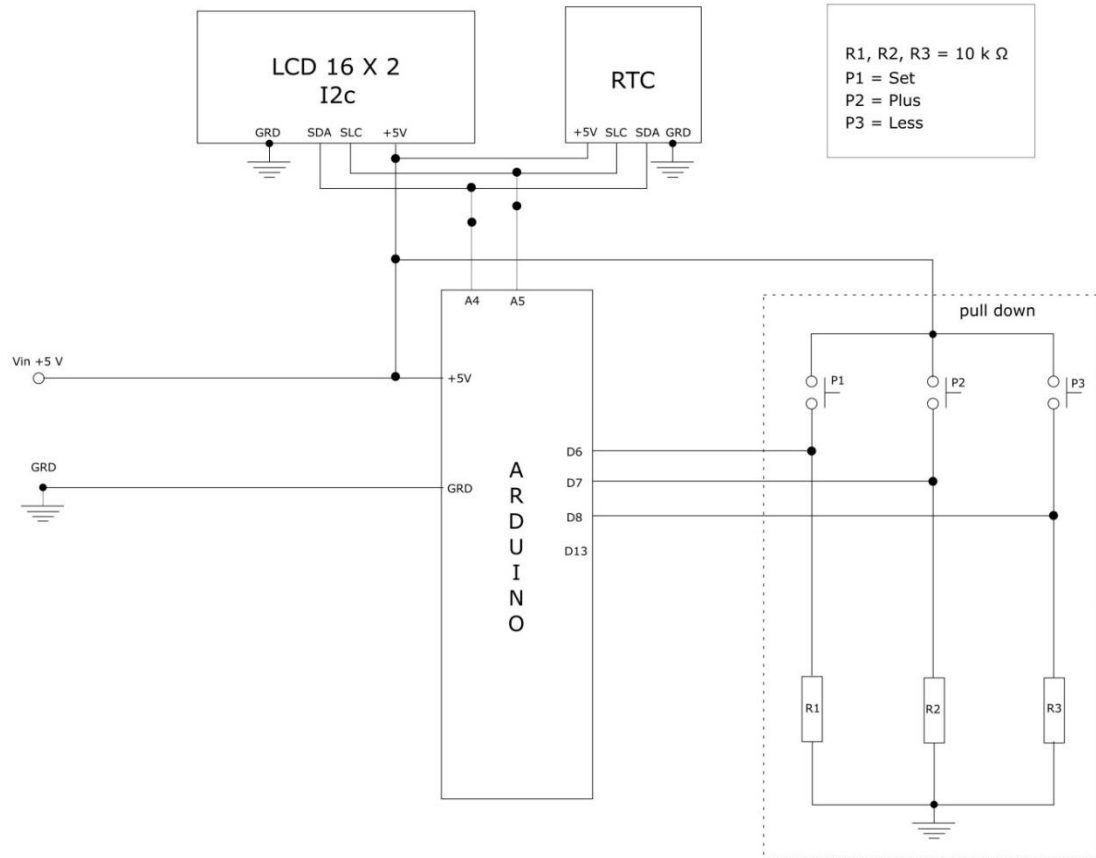


Fig-4.2: Flowchart

## 4.4 Diagram of Day and Date Circuit



**Fig-4.3:** Diagram of Day and Date Circuit

## 4.5 Diagram of Temperature Circuit

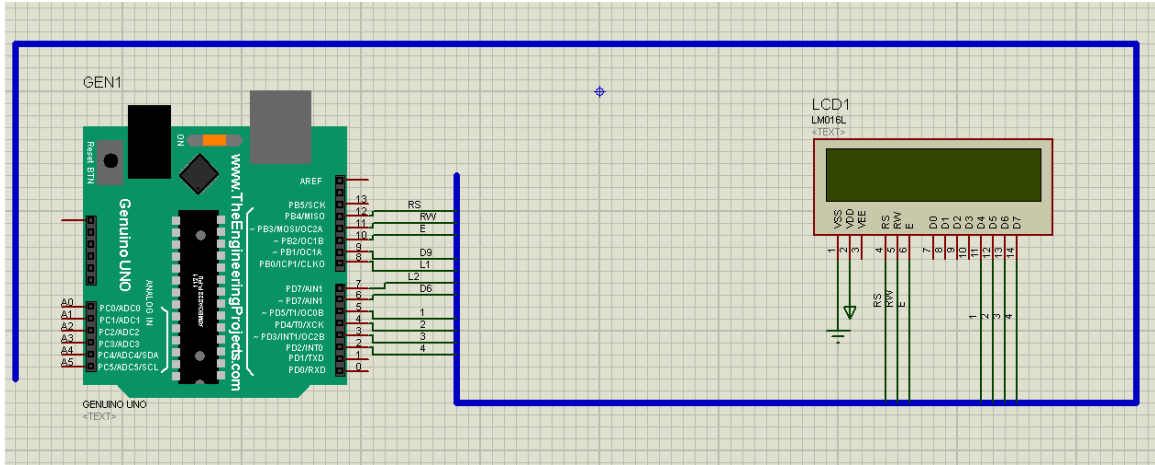


Fig-4.4: Diagram of Day and Date Circuit

## 4.6 PCB Design

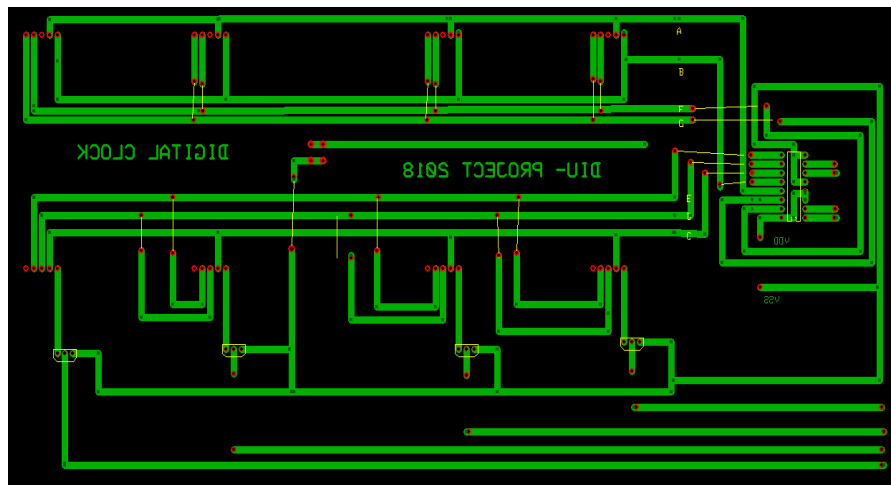
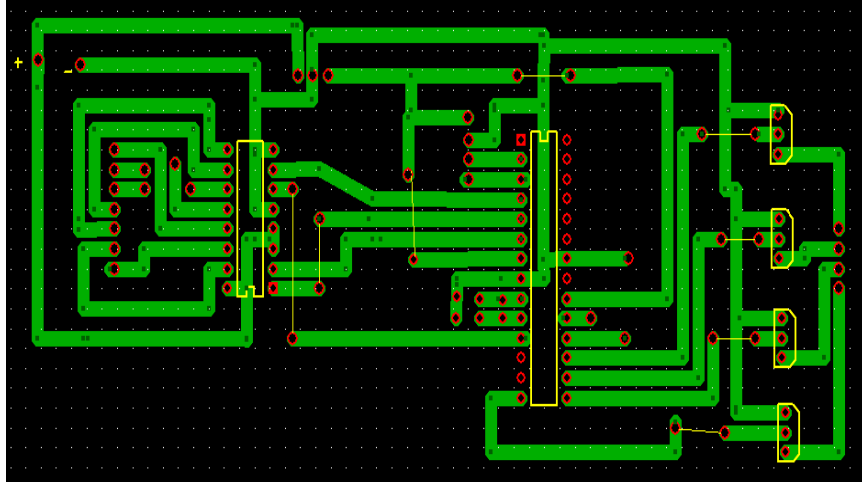
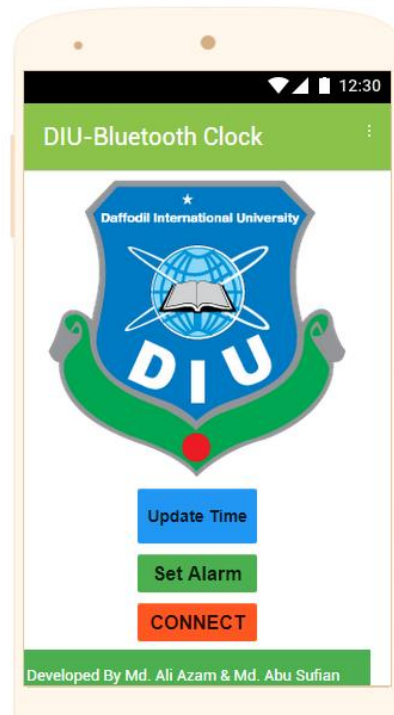


Fig-4.5: PCB design for 7 segment part



**Fig-4.6:** PCB design for Temp & date part.

## 4.7 Apps Design



**Fig-4.7:** Apps Interface

## 4.8 Logic of Apps

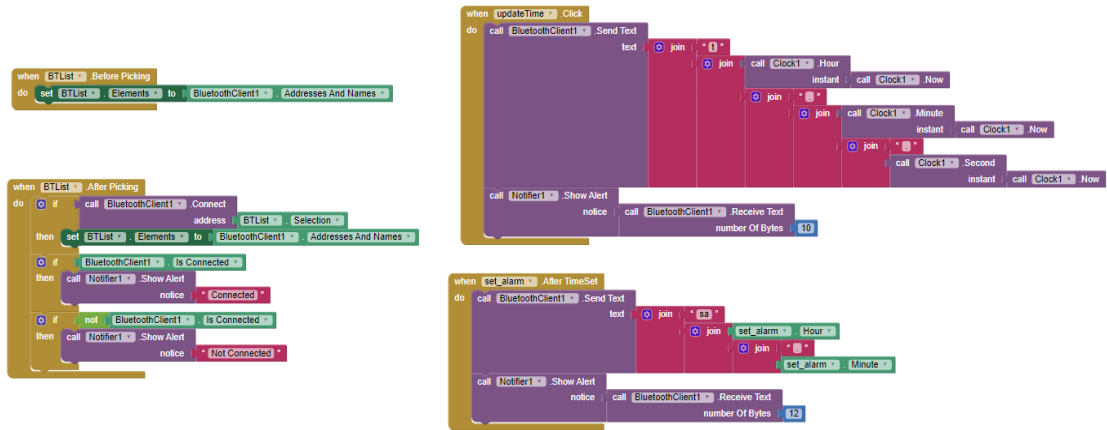


Fig-4.8: Logic of Apps

# Chapter 5

## Cost Analysis

## 5.1 Cost Table

SL No.	Parts Name	Unit	Total Price in Tk
1	Seven Segment display 2.3"	4	240
2	ATMega328p	3	525
3	Bluetooth Module	1	350
4	Voltage Regulator IC L7805	2	220
5	Driver IC CD4511	2	520
6	PCB Board	2	400
7	NPN Transistor	10	20
8	Piezo Electric Buzzer	1	30
9	IC Base 28 Pin	3	30
10	IC Base 16 Pin	2	20
11	9v Rechargeable Battery	1	280
12	DHT Temperature & Humidity Sensor	1	300
13	I2C LCD Display 16*2	1	388
14	I2C LCD Display 20*4	1	525
15	16 MHz Crystal Oscillator	3	27
16	Ceramic Capacitor (20pF)	10	10
17	RTC Module	1	165
18	Push Switch	4	16
19	Jumper wire	50	100
20	9V Ac to Dc Adapter	1	130
		<b>Total</b>	<b>= 4296</b>

**Table No. 5.1**



# Chapter 6

## Discussion & Recommendation

## **6.1 Conclusion**

We have designed a unique PIC and seven segments based intelligent Digital clock successfully. Its required power is very low. To make it user friendly we have kept different button for second, minute, hour to operate the device.

The accuracy of the clock is dependent upon the accuracy of the crystal and the accuracy of the match between the capacity load of the oscillator circuit and the capacitive load for which the crystal was trimmed. Additional error will be added by crystal frequency drift caused by temperature shifts. External circuit noise coupled into the oscillator circuit may result in the running fast.

This project provided significant information that will enable to build a complete Digital Clock. Overall we are very hopeful about our proposed device.

## **6.2 Limitations of the work**

Microchips has not been used in this project because lack of machine we can not make microchips at home. That why this project is bigger in size that the clocks available in the market.

## **6.3 Future Scope**

- In Future we want to add Alarm System
- Auto Wake up System

## **6.4 Advantages**

- We can use big Size Display in our Digital Clock.
- Power Consumption is very low.
- Temperature and humidity Display System.
- Controlled by Apps.

## References:

- [1] [wineyard.in/wp-content/uploads/2012/12/WYARD9.pdf](http://wineyard.in/wp-content/uploads/2012/12/WYARD9.pdf)
- [2] [http://www.pelnet.co.uk/elect/projects/clock/Clock\\_a.pdf](http://www.pelnet.co.uk/elect/projects/clock/Clock_a.pdf)
- [3] [http://en.wikipedia.org/wiki/Digital\\_clock](http://en.wikipedia.org/wiki/Digital_clock)
- [4] [www.ee.duke.edu/~jmorizio/ece261/F09/projects/clk.pdf](http://www.ee.duke.edu/~jmorizio/ece261/F09/projects/clk.pdf)
- [5] [https://www.hni.uni-paderborn.de/uploads/media/doc2486\\_01.pdf](https://www.hni.uni-paderborn.de/uploads/media/doc2486_01.pdf)
- [6] [datasheets.maximintegrated.com/en/ds/DS1307.pdf](http://datasheets.maximintegrated.com/en/ds/DS1307.pdf)
- [7] <https://www.electronics-tutorials.ws/blog/7-segment-display-tutorial.html>

# Appendix 1

```
#include <Wire.h>
#include <RTClib.h>
#include <LiquidCrystal_I2C.h>

//*****//
LiquidCrystal_I2C lcd(0x27,20,4); // Display I2C 16 x 2
RTC_DS1307 RTC;

//*****Button*****//
int P1=6; // Button SET MENU'
int P2=7; // Button +
int P3=8; // Button -

//*****Variables*****//
int hourupg;
int minupg;
int yearupg;
int monthupg;
int dayupg;
int menu =0;

void setup()
{

  lcd.begin();
  lcd.backlight();
```

```

lcd.clear();

pinMode(P1,INPUT);
pinMode(P2,INPUT);
pinMode(P3,INPUT);

Serial.begin(9600);
Wire.begin();
RTC.begin();

if (! RTC.isrunning()) {
  Serial.println("RTC is NOT running!");
  // Set the date and time at compile time
  RTC.adjust(DateTime(__DATE__, __TIME__));
}
// RTC.adjust(DateTime(__DATE__, __TIME__)); //removing "/" to adjust the time
// The default display shows the date and time
int menu=0;
}

void loop()
{

// check if you press the SET button and increase the menu index
if(digitalRead(P1))
{
  menu=menu+1;
}
// in which subroutine should we go?

```

```
if (menu==0)
{
    DisplayDateTime(); // void DisplayDateTime

}
if (menu==1)
{
    DisplaySetHour();
}
if (menu==2)
{
    DisplaySetMinute();
}
if (menu==3)
{
    DisplaySetYear();
}
if (menu==4)
{
    DisplaySetMonth();
}
if (menu==5)
{
    DisplaySetDay();
}
if (menu==6)
{
    StoreAgg();
    delay(500);
}
```

```

    menu=0;
  }
  delay(100);
}

void DisplayDateTime ()
{
// We show the current date and time
DateTime now = RTC.now();

  lcd.setCursor(0, 1);
  lcd.print("Hour:");
  if (now.hour()<=9)
  {
    lcd.print("0");
  }
  lcd.print(now.hour(), DEC);
  hourupg=now.hour();
  lcd.print(":");
  if (now.minute()<=9)
  {
    lcd.print("0");
  }
  lcd.print(now.minute(), DEC);
  minupg=now.minute();
  lcd.print(":");
  if (now.second()<=9)
  {
    lcd.print("0");
  }
}

```



```

}
lcd.print(now.second(), DEC);

lcd.setCursor(0, 0);
lcd.print("Date: ");
if (now.day()<=9)
{
  lcd.print("0");
}
lcd.print(now.day(), DEC);
dayupg=now.day();
lcd.print("/");
if (now.month()<=9)
{
  lcd.print("0");
}
lcd.print(now.month(), DEC);
monthupg=now.month();
lcd.print("/");
lcd.print(now.year(), DEC);
yearupg=now.year();
}

void DisplaySetHour()
{
// time setting
  lcd.clear();
  DateTime now = RTC.now();
  if(digitalRead(P2)==HIGH)

```

```

{
  if(hourupg==23)
  {
    hourupg=0;
  }
  else
  {
    hourupg=hourupg+1;
  }
}
if(digitalRead(P3)==HIGH)
{
  if(hourupg==0)
  {
    hourupg=23;
  }
  else
  {
    hourupg=hourupg-1;
  }
}
lcd.setCursor(0,0);
lcd.print("Set time:");
lcd.setCursor(0,1);
lcd.print(hourupg,DEC);
delay(200);
}

void DisplaySetMinute()

```

```

{
// Setting the minutes
lcd.clear();
if(digitalRead(P2)==HIGH)
{
if (minupg==59)
{
minupg=0;
}
else
{
minupg=minupg+1;
}
}
if(digitalRead(P3)==HIGH)
{
if (minupg==0)
{
minupg=59;
}
else
{
minupg=minupg-1;
}
}
}
lcd.setCursor(0,0);
lcd.print("Set Minutes:");
lcd.setCursor(0,1);
lcd.print(minupg,DEC);

```

```

    delay(200);
}

void DisplaySetYear()
{
// setting the year
    lcd.clear();
    if(digitalRead(P2)==HIGH)
    {
        yearupg=yearupg+1;
    }
    if(digitalRead(P3)==HIGH)
    {
        yearupg=yearupg-1;
    }
    lcd.setCursor(0,0);
    lcd.print("Set Year:");
    lcd.setCursor(0,1);
    lcd.print(yearupg,DEC);
    delay(200);
}

```

```

void DisplaySetMonth()
{
// Setting the month
    lcd.clear();
    if(digitalRead(P2)==HIGH)
    {
        if (monthupg==12)

```

```

    {
        monthupg=1;
    }
    else
    {
        monthupg=monthupg+1;
    }
}
if(digitalRead(P3)==HIGH)
{
    if (monthupg==1)
    {
        monthupg=12;
    }
    else
    {
        monthupg=monthupg-1;
    }
}
}
lcd.setCursor(0,0);
lcd.print("Set Month:");
lcd.setCursor(0,1);
lcd.print(monthupg,DEC);
delay(200);
}

```

```

void DisplaySetDay()

```

```

{
// Setting the day

```

```
lcd.clear();
if(digitalRead(P2)==HIGH)
{
  if (dayupg==31)
  {
    dayupg=1;
  }
  else
  {
    dayupg=dayupg+1;
  }
}
if(digitalRead(P3)==HIGH)
{
  if (dayupg==1)
  {
    dayupg=31;
  }
  else
  {
    dayupg=dayupg-1;
  }
}
lcd.setCursor(0,0);
lcd.print("Set Day:");
lcd.setCursor(0,1);
lcd.print(dayupg,DEC);
delay(200);
}
```

```

void StoreAgg()
{
// Variable saving
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("SAVING IN");
  lcd.setCursor(0,1);
  lcd.print("PROGRESS");
  RTC.adjust(DateTime(yearupg,monthupg,dayupg,hourupg,minupg,0));
  delay(200);
}

```

## Appendix 2

```

#include <Wire.h>
#include <RTCLib.h>
#include <LiquidCrystal_I2C.h>

//*****//
LiquidCrystal_I2C lcd(0x3F,16,2); // Display I2C 16 x 2
RTC_DS1307 RTC;

//*****Button*****//
int P1=6; // Button SET MENU'
int P2=7; // Button +
int P3=8; // Button -

```

```

//*****Variables*****//

int hourupg;
int minupg;
int yearupg;
int monthupg;
int dayupg;
int menu =0;

void setup()
{

  lcd.begin();
  lcd.backlight();
  lcd.clear();

  pinMode(P1,INPUT);
  pinMode(P2,INPUT);
  pinMode(P3,INPUT);

  Serial.begin(9600);
  Wire.begin();
  RTC.begin();

  if (! RTC.isrunning()) {
    Serial.println("RTC is NOT running!");
    // Set the date and time at compile time
    RTC.adjust(DateTime(__DATE__, __TIME__));
  }
  // RTC.adjust(DateTime(__DATE__, __TIME__)); //removing "/" to adjust the time

```



```

    // The default display shows the date and time
    int menu=0;
}

void loop()
{

// check if you press the SET button and increase the menu index
if(digitalRead(P1))
{
    menu=menu+1;
}
// in which subroutine should we go?
if (menu==0)
{
    DisplayDateTime(); // void DisplayDateTime

}
if (menu==1)
{
    DisplaySetHour();
}
if (menu==2)
{
    DisplaySetMinute();
}
if (menu==3)
{
    DisplaySetYear();
}
}

```

```

    }
    if (menu==4)
    {
        DisplaySetMonth();
    }
    if (menu==5)
    {
        DisplaySetDay();
    }
    if (menu==6)
    {
        StoreAgg();
        delay(500);
        menu=0;
    }
    delay(100);
}

void DisplayDateTime ()
{
// We show the current date and time
    DateTime now = RTC.now();

    lcd.setCursor(0, 1);
    lcd.print("Hour:");
    if (now.hour()<=9)
    {
        lcd.print("0");
    }
}

```

```
lcd.print(now.hour(), DEC);  
hourupg=now.hour();  
lcd.print(":");  
if (now.minute()<=9)  
{  
    lcd.print("0");  
}  
lcd.print(now.minute(), DEC);  
minupg=now.minute();  
lcd.print(":");  
if (now.second()<=9)  
{  
    lcd.print("0");  
}  
lcd.print(now.second(), DEC);
```

```
lcd.setCursor(0, 0);  
lcd.print("Date: ");  
if (now.day()<=9)  
{  
    lcd.print("0");  
}  
lcd.print(now.day(), DEC);  
dayupg=now.day();  
lcd.print("/");  
if (now.month()<=9)  
{  
    lcd.print("0");  
}
```

```
lcd.print(now.month(), DEC);  
monthupg=now.month();  
lcd.print("/");  
lcd.print(now.year(), DEC);  
yearupg=now.year();  
}
```

```
void DisplaySetHour()  
{  
// time setting  
lcd.clear();  
DateTime now = RTC.now();  
if(digitalRead(P2)==HIGH)  
{  
if(hourupg==23)  
{  
hourupg=0;  
}  
else  
{  
hourupg=hourupg+1;  
}  
}  
if(digitalRead(P3)==HIGH)  
{  
if(hourupg==0)  
{  
hourupg=23;  
}  
}
```

```

else
{
    hourupg=hourupg-1;
}
}
lcd.setCursor(0,0);
lcd.print("Set time:");
lcd.setCursor(0,1);
lcd.print(hourupg,DEC);
delay(200);
}

void DisplaySetMinute()
{
// Setting the minutes
lcd.clear();
if(digitalRead(P2)==HIGH)
{
    if (minupg==59)
    {
        minupg=0;
    }
    else
    {
        minupg=minupg+1;
    }
}
if(digitalRead(P3)==HIGH)
{

```

```

if (minupg==0)
{
  minupg=59;
}
else
{
  minupg=minupg-1;
}
}
lcd.setCursor(0,0);
lcd.print("Set Minutes:");
lcd.setCursor(0,1);
lcd.print(minupg,DEC);
delay(200);
}

```

```

void DisplaySetYear()
{
// setting the year
lcd.clear();
if(digitalRead(P2)==HIGH)
{
  yearupg=yearupg+1;
}
if(digitalRead(P3)==HIGH)
{
  yearupg=yearupg-1;
}
lcd.setCursor(0,0);

```

```

lcd.print("Set Year:");
lcd.setCursor(0,1);
lcd.print(yearupg,DEC);
delay(200);
}

void DisplaySetMonth()
{
// Setting the month
lcd.clear();
if(digitalRead(P2)==HIGH)
{
if (monthupg==12)
{
monthupg=1;
}
else
{
monthupg=monthupg+1;
}
}
if(digitalRead(P3)==HIGH)
{
if (monthupg==1)
{
monthupg=12;
}
else
{

```

```

    monthupg=monthupg-1;
}
}
lcd.setCursor(0,0);
lcd.print("Set Month:");
lcd.setCursor(0,1);
lcd.print(monthupg,DEC);
delay(200);
}

```

```

void DisplaySetDay()
{
// Setting the day
lcd.clear();
if(digitalRead(P2)==HIGH)
{
if (dayupg==31)
{
dayupg=1;
}
else
{
dayupg=dayupg+1;
}
}
if(digitalRead(P3)==HIGH)
{
if (dayupg==1)
{

```



```

    dayupg=31;
}
else
{
    dayupg=dayupg-1;
}
}
lcd.setCursor(0,0);
lcd.print("Set Day:");
lcd.setCursor(0,1);
lcd.print(dayupg,DEC);
delay(200);
}

void StoreAgg()
{
// Variable saving
lcd.clear();
lcd.setCursor(0,0);
lcd.print("SAVING IN");
lcd.setCursor(0,1);
lcd.print("PROGRESS");
RTC.adjust(DateTime(yearupg,monthupg,dayupg,hourupg,minupg,0));
delay(200);
}

```

## Appendix 3

```
#include <dht.h>
```

```

#define dht_apin A0 // Analog Pin sensor is connected to
dht DHT;
void setup(){
  Serial.begin(9600);
  delay(500);//Delay to let system boot
  Serial.println("DHT11 Humidity & temperature Sensor\n\n");
  delay(1000);//Wait before accessing Sensor

} //end "setup()"

void loop(){
  //Start of Program

  DHT.read11(dht_apin);

  Serial.print("Current humidity = ");
  Serial.print(DHT.humidity);
  Serial.print("% ");
  Serial.print("temperature = ");
  Serial.print(DHT.temperature);
  Serial.println("C ");

  delay(5000);//Wait 5 seconds before accessing sensor again.
  //Fastest should be once every two seconds.

} // end loop()
void loop(){
  //Start of Program
  DHT.read11(dht_apin);

```

```
Serial.print("Current humidity = ");  
Serial.print(DHT.humidity);  
Serial.print("% ");  
Serial.print("temperature = ");  
Serial.print(DHT.temperature);  
Serial.println("C ");  
delay(5000);//Wait 5 seconds before accessing sensor again.
```