AUTO INTENSITY CONTROL OF STREET LIGHTS

MINIPROJECT REPORT

SUBMITTED BY:

ADITHYA HEGDE

CHANDAN S

SHREEKANTH

In partial fulfilment for the award of the degree of

B.E

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

ABSTRACT

Auto Intensity Control of Street Lights is designed to automatically switch ON the Street Light alongside the roads (or path) or the light lamp just outside our house on the onset of dark weather or at dusk & switch them off automatically after sunrise or during the light hours.

This system would help in reducing the electricity consumption & also save from unprecedented accidents & thus will increase the safety of the pedestrians & prevent mishaps.

Auto Intensity Control of Street Lights is a simple project where the intensity of the streetlights is automatically controlled based on the sunlight conditions. Generally, streetlights are turned on during evening time and will continue to glow till morning.

This might result is unnecessary usage of power as the lights will be glowing at full intensity all the times. But using the Auto Intensity Control of Street Lights using Arduino project, you can control the intensity based on the ambient lighting conditions.

As an additional power saving feature, I have used LEDs for streetlights.

1.Components Required

- Arduino UNO
- DS3231 RTC Module
- LDR
- 16×2 LCD Display
- LED
- 10KΩ Potentiometer
- 10KΩ Resistor
- Push Button
- Connecting Wires
- Breadboard

2.Circuit Description

In this project we have used LDR and RTCDS3231.The LDR is used for detecting the intensity of the light and the 1 pin of the LDR and 10kohm resistor is connected in voltage divider form and is connected to the analog pin of the Arduino uno that is A3 pin, as for the RTC module the SCL and the SDA pins of the RTC module is connected to the A4 and A5 pins of the Arduino. So, to switch between two modes that is LDR mode or RTC mode there is switch button placed so when pressed they switch between two modes. The switch button is connected to the digital pin D2 of the Arduino uno.

Since there is a RTC module to show the time there must be output, so we have connected a lcd display to the Arduino uno and digital pins of the LCD D4 to D7 is connected to the d3 to d6 pins of the Arduino uno respectively, the enable pin is connected to the digital pin D7 of the Arduino, the register select is connected to the digital D8 pin on the Arduino uno.

The LED is connected to the digital out of the Arduino that is the D11 pin. When the mode is in RTC the LED is switched off and on based on time whereas for the LDR the intensity of the light is high the lights are switched off and when the intensity is very low as low as zero percent the LED is turned off. This is the basic connections of this project.

3.Circuit Diagram

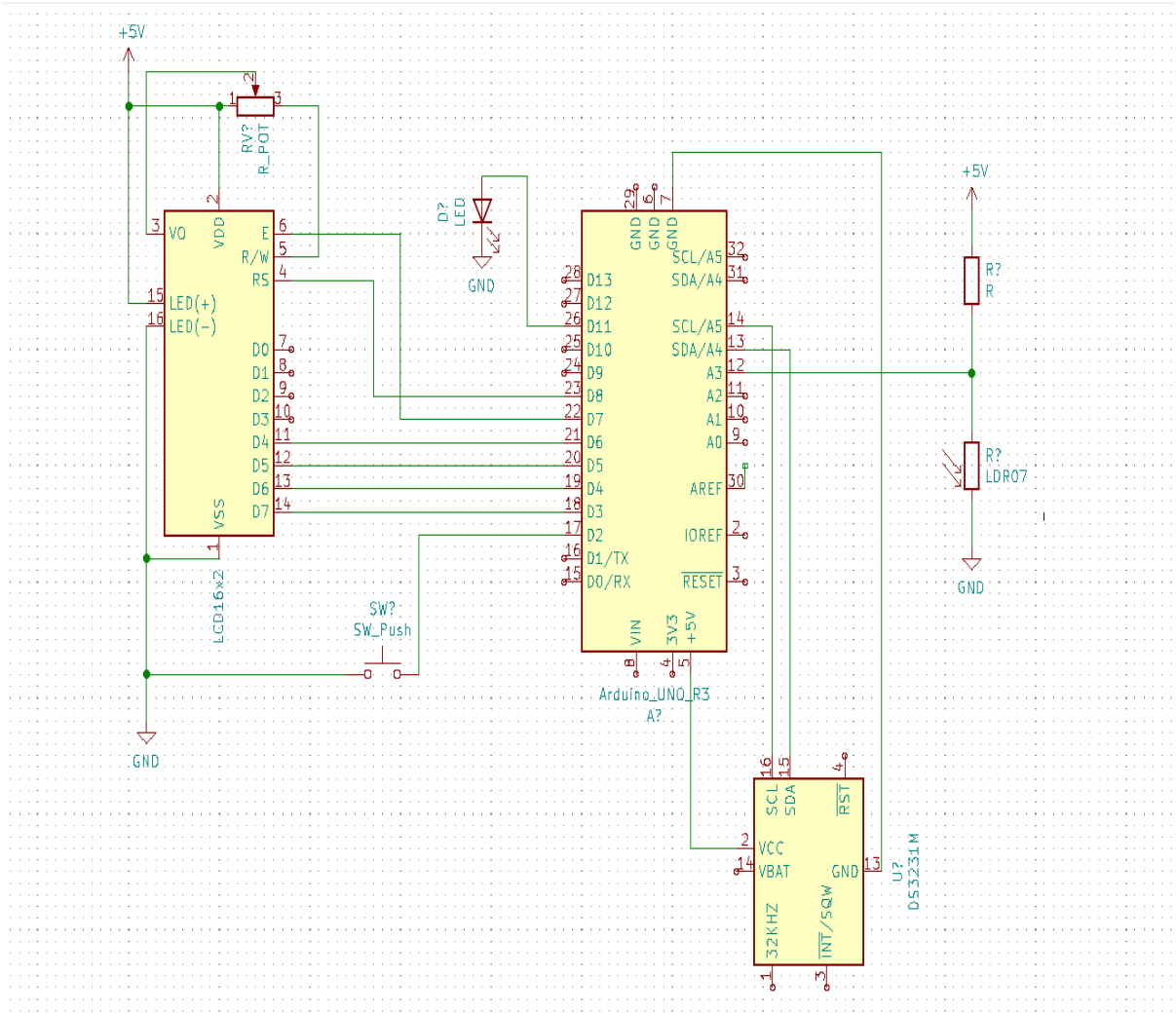

Fig 3.1 Circuit Diagram

4.Components Description

4.1Arduino Uno

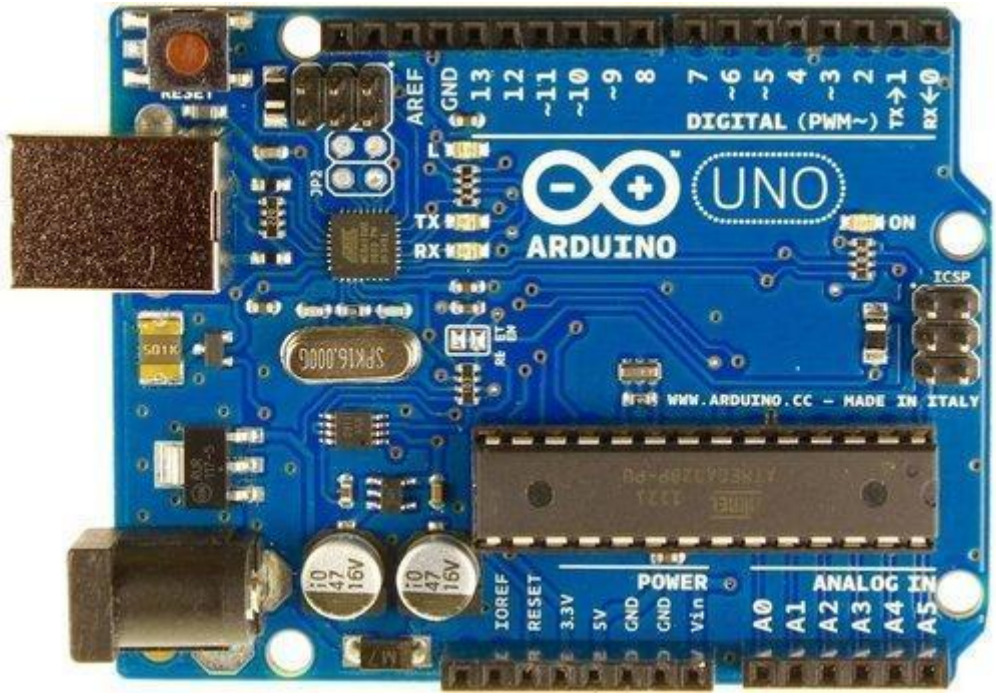

Fig 4.1 UNO board

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 computerized input/yield pins (of which 6 can be utilized as PWM yields), 6 simple sources of info, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB association, a power jack, an ICSP header and a reset button. It contains all that expected to help the microcontroller; basically, associate it to a PC with a USB link or power it with an AC-to-DC connector or battery to begin. You can dabble with your Uno without stressing a lot over accomplishing something incorrectly, most dire outcome imaginable you can substitute the chip for a couple of dollars and begin once more.

The Arduino Uno is an open-source microcontroller board reliant on the Microchip ATmega328P microcontroller and made by Arduino.cc. The board is outfitted with sets of cutting edge and straightforward information/yield (I/O) sticks that may be interfaced to various augmentation sheets (shields) and different circuits. The board has 14digital I/O pins (six prepared for PWM output), 6 basic I/O sticks, and is programmable with the Arduino IDE (Integrated Development Environment), through a sort of B USB link. It tends to be powered by the USB interface or by an external 9-volt battery, anyway it recognizes voltages some place in the scope of 7 and 20 volts. It resembles the Arduino Nano and Leonardo. The

equipment reference configuration is appropriated under a Creative Commons Attribution Share-Alike 2.5 permit and is accessible on the Arduino site. Format and creation records for certain variants of the equipment are likewise accessible.

"Uno" signifies "one" in Italian and was picked to stamp the underlying arrival of Arduino Software. The Uno board is the first in a progression of USB-based Arduino boards; it and form 1.0 of the Arduino IDE were the reference adaptations of Arduino, which have now advanced to more current releases. The ATmega328 on the board comes preprogrammed with a bootloader that permits transferring new code to it without the utilization of an outer equipment programmer.

While the Uno conveys utilizing the first STK500 protocol, it contrasts from all former sheets in that it doesn't utilize the FTDI USB-to-chronic driver chip. All things considered; it utilizes the Atmega16U2 (Atmega8U2 up to form R2) customized as a USB-to-sequential converter.

4.1.1 Specifications of Arduino uno

- Microcontroller: Microchip ATmega328P
- Operating Voltage: 5 Volts
- Input Voltage: 7 to 20 Volts
- Digital I/O Pins: 14 (of which 6 can provide PWM output)
- UART: 1
- I2C: 1
- SPPI: 1
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader.
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz

4.1.2 Pin out of Arduino Uno

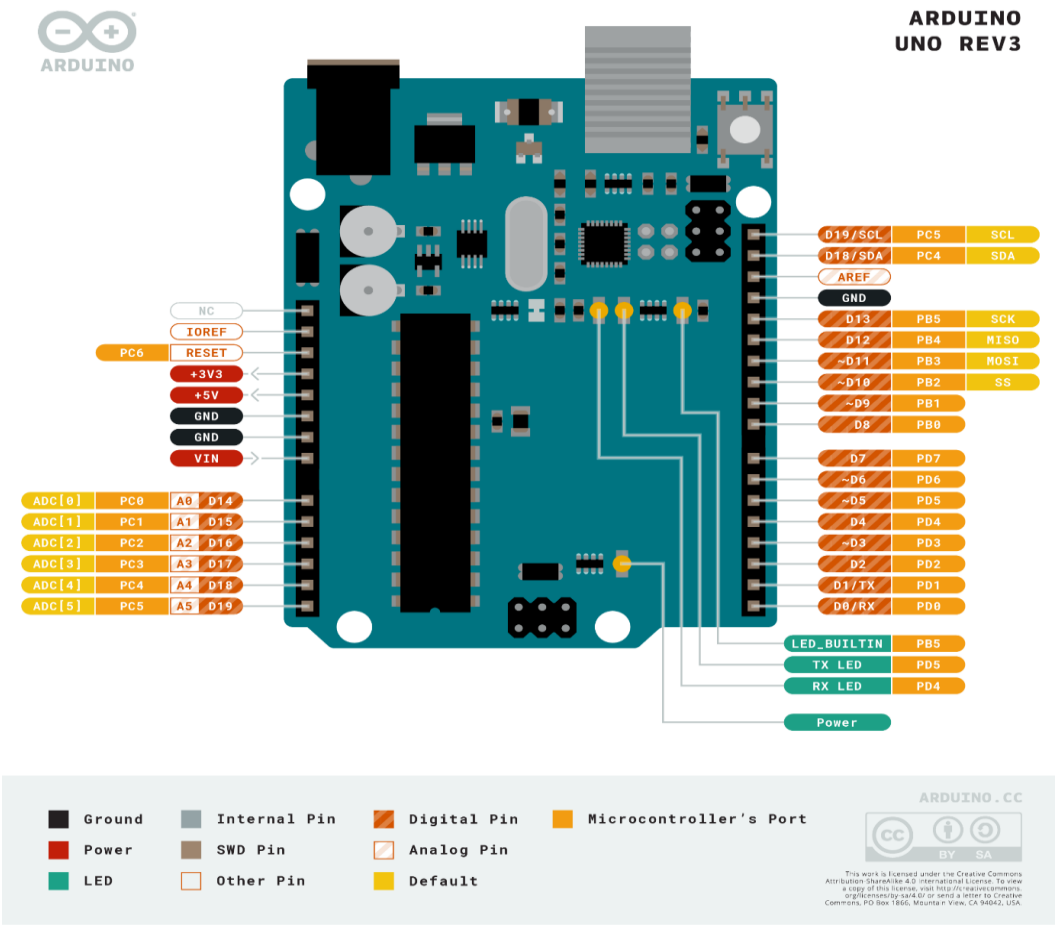

Fig4.2 Pin Diagram For UNO

There are a few I/O digital and analog pins put on the board which works at 5V. These pins accompany standard working evaluations running between 20mA to 40mA. Internal pull up resistors are utilized in the board that restricts the current surpassing from the given working conditions. Nonetheless, an excess of expansion in current makes these resisters pointless and harms the device.

LED. Arduino Uno accompanies built in LED which is associated through pin 13. Offering HIGH benefit to the pin will turn it ON and LOW will turn it OFF.

Vin. It is the information voltage gave to the Arduino Board. It is unique in relation to 5 V provided through a USB port. This pin is utilized to supply voltage. On the off chance that a voltage is given through power jack, it very well may be gotten to through this pin.

5V. This board accompanies the capacity to give voltage regulations. 5V pin is utilized to give yield regulated voltage. The block is controlled utilizing three different ways for example USB,

Vin pin of the board or DC power jack. USB underpins voltage around 5V while Vin and Power Jack uphold a voltage range between 7V to 20V. It is prescribed to work the board on 5V. It is imperative to take note of that, if a voltage is provided through 5V or 3.3V pins, they bring about bypassing the voltage guideline that can harm the board if voltage outperforms from its limit.

GND. These are ground pins. More than one ground pins are given on the board which can be utilized according to requirement.

Reset. As opposed to requiring an actual press of the reset button before an upload, the Arduino/Genuino Uno board is planned in a way that permits it to be reset by programming running on an associated computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is associated with the reset line of the ATmega328 by means of a 100 nano-farad capacitor. At the point when this line is asserted (taken low), the reset line drops adequately long to reset the chip.

This arrangement has different ramifications. At the point when the Uno is associated with a PC running Mac OS X or Linux, it resets each time an association is made to it from programming (through USB). For the accompanying half-second or something like that, the bootloader is running on the Uno. While it is modified to disregard deformed information (for example anything but a transfer of new code), it will block the initial not many bytes of information shipped off the board after an association is opened.

IOREF. This pin is helpful for giving voltage reference to the board. A shield is utilized to peruse the voltage across this pin which at that point select the appropriate power source.

PWM. PWM is given by 3,5,6,9,10, 11pins. These pins are arranged to give 8-bit yield PWM.

SPI. It is known as Serial Peripheral Interface. Four pins 10(SS), 11(MOSI), 12(MISO), 13(SCK) give SPI correspondence the assistance of SPI library.

AREF. It is called Analog Reference. This pin is utilized for giving a reference voltage to the analog inputs.

TWI. It is called Two-wire Interface. TWI correspondence is gotten to through Wire Library. A4 and A5 pins are utilized for this purpose.

Serial Communication. Sequential correspondence is brought out through two pins called Pin 0 (Rx) and Pin 1 (Tx). Rx pin is utilized to get information while Tx pin is utilized to send data.

External Interrupts. Pin 2 and 3 are utilized for giving outer interferes. A hinder is called by offering LOW or changing values.

4.1.3 Communications and Programming.

Arduino Uno accompanies a capacity of interfacing with other Arduino boards, micro-controllers and computers. The Atmega328 set on the board gives serial communication utilizing pins like Rx and Tx. The Atmega16U2 incorporated on the board gives a pathway to

serial communication utilizing USB com drivers. Serial monitor is given on the IDE programming which is utilized to send or get text information from the board. If LEDs set on the Rx and Tx pins will flash, they demonstrate the transmission of data. Arduino Uno is customized utilizing Arduino Software which is a cross-stage application called IDE written in Java. The AVR microcontroller Atmega328 spread out on the base accompanies built in bootloader that liberates you from utilizing a different burner to transfer the program on the board.

4.1.4 Applications of Arduino Uno.

Arduino Uno accompanies a wide scope of utilizations. A bigger number of individuals are utilizing Arduino board for creating sensors and instruments that are utilized in logical exploration. Following are some fundamental uses of the board.

- Embedded System
- Security and Defense System
- Digital Electronics and Robotics
- Parking Lot Counter
- Weighing Machines
- Traffic Light Count Down Timer
- Medical Instrument
- Emergency Light for Railways
- Home Automation
- Industrial Automation

4.2RTC DS3231

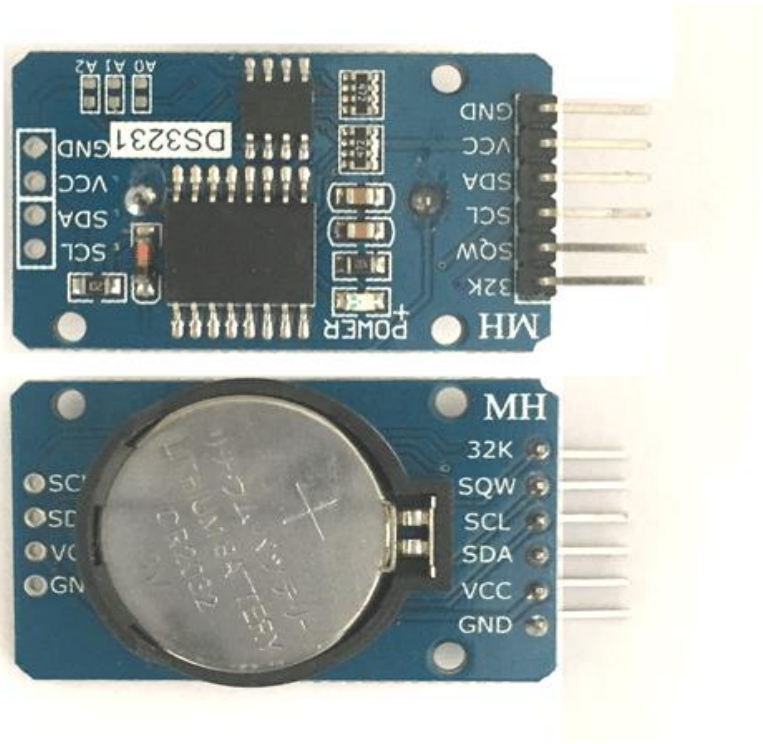

Fig 4.3 RTC DS3231 Module

RTC implies to Real Time Clock. RTC modules are basically TIME and DATE recalling frameworks which contain battery, this is like the CMOS present in the computers to remember the date and the time of the system. Similarly, the RTC module doesn't require any external power and the battery keeps the module running and keeps the time and date up to date. So, we can have precise time and date from the RTC module whenever we want.

At the core of the module is a low cost, incredibly exact RTC chip from Maxim – DS3231. It deals with all timekeeping function and highlights a basic two-wire I2C interface which can be effectively interfaced with any microcontroller of your decision.

The chip looks after seconds, minutes, hours, day, date, month, and year data. The date toward the month's end is consequently changed for quite a long time with less than 31 days, including rectifications for jump year (substantial up to 2100).

The clock works in either the 24-hour or 12-hour design with an AM/PM marker. It additionally gives two programmable season of-day alarms.

The other element of this board accompanies SQW pin, which outputs a decent square wave at either 1Hz, 4kHz, 8kHz or 32kHz and can be dealt with programming. This can additionally be utilized as a hinder because of caution condition in many time sensitive applications.

4.2.1 DS 3231 RTC MODULE PIN CONFIGURATION

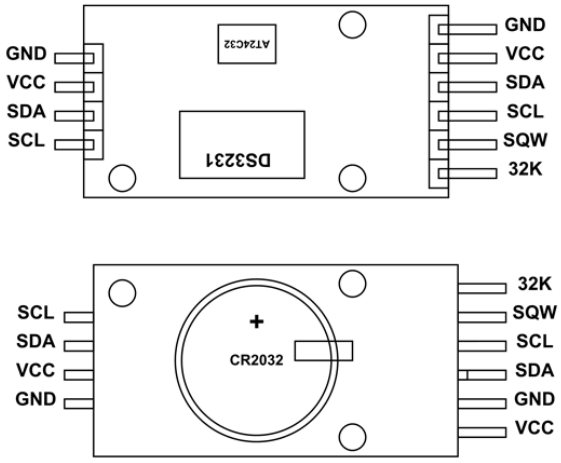

Fig 4.4 Pin Diagram For RTC Module

VCC: -pin supplies power for the module. It can be anywhere between 3.3V to 5.5V.

GND: -Connected to ground pin of the Arduino

SDA: -Serial Data pin (I2C interface)

SCL: -Serial Clock pin (I2C interface)

SQW: -pin outputs a nice square wave at either 1Hz, 4kHz, 8kHz or 32kHz and can be handled programmatically. This can further be used as an interrupt due to alarm condition in many time-based applications.

32K: -32K oscillator output the stable (temperature compensated) and accurate reference clock.

4.2.2 FEATURES of DS3231 RTC module

1.Temperature Compensated Crystal Oscillator (TCXO)

Most RTC modules accompany an external 32kHz crystal oscillator for timekeeping. In any case, the issue with these precious stones is that outside temperature can influence their oscillation frequency. This adjustment in recurrence can be unimportant yet it clearly adds up.

To keep away from such slight floats in crystal, DS3231 is driven by a 32kHz temperature compensated oscillator (TCXO). It's exceptionally safe to the outside temperature changes.

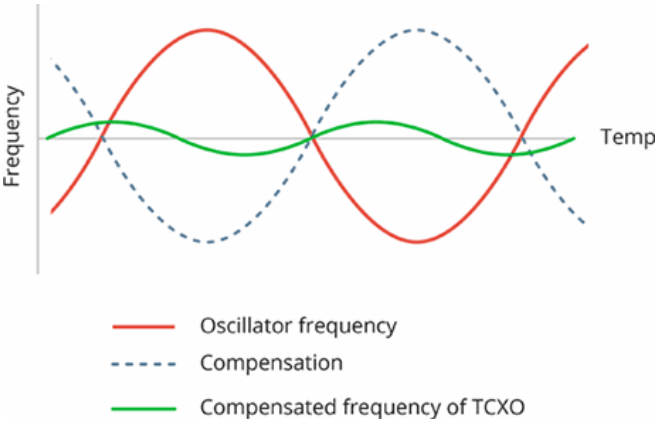

Fig4.4.1 Freq & Temp Graph

TCXO is bundled inside the RTC chip, making the entire bundle massive. Directly close to the incorporated crystal is a temperature sensor.

This sensor repays the frequency changes by adding or eliminating clock ticks with the goal that the timekeeping remains on target.

That is the explanation TCXO gives a steady and exact reference clock and keeps up the RTC to inside ±2 minutes of the year exactness.

2.Battery Backup

The DS3231 incorporates a battery input and keeps up exact timekeeping when main power to the gadget is interrupted.

The built-in power sense circuit constantly monitors the status of VCC to distinguish power failures and naturally changes to the backup power supply. Along these lines, you need not worry about power blackouts, your MCU can at present monitor time.

Fig 4.5 RTC battery Holder

The bottom side of the board holds a battery holder for 20mm 3V lithium coin cells. Any CR2032 battery can fit well.

If a fully charged coin cell is placed in the module the battery can last up to 8 years.so the power usage of the module will be around 3µA.

3.On Board EPROM

DS3231 RTC module likewise accompanies a 32 bytes 24C32 EEPROM chip from Atmel having limitless read write cycles. It tends to be utilized to save settings or truly anything.

The 24C32 EEPROM utilizes I2C interface for communication and offers a similar I2C bus as DS3231.

The I2C address of the EEPROM can be changed effectively with the three A0, A1 and A2 bind jumpers at the back. Each one of these is utilized to hardcode in the address. On the off chance that a jumper is shorted with solder, that sets the address.

According to the 24C32's datasheet, these 3 pieces are put toward the finish of the 7-piece I2C address, not long before the Read/Write bit.

As there are 3 location inputs, which can take 2 states HIGH/LOW, we can consequently make 8 (23) diverse combinations(addresses).

4.3LDR (LIGHT DEPENDENT RESISTOR):

Fig 4.6 LDR

A photoresistor or light dependent resistor is an electronic component whose value is sensitive to light. At the point when light falls upon it, the resistance changes. Estimations of the resistance of the LDR may change over numerous significant degrees the estimation of the obstruction falling as the degree of light increments.

It isn't remarkable for the estimations of resistance of a LDR or photoresistor to be a few megohms in dimness and afterward to tumble to two or three hundred ohms in bright light. With quite a wide variety in resistance, LDRs are easy to utilize and there are numerous LDR circuits accessible. The affectability of light dependent resistors or photoresistors additionally changes with the wavelength of the incident light.

LDRs are produced using semiconductor materials to empower them to have their light delicate properties. Numerous materials can be utilized, however one famous material for these photoresistors is cadmium sulfide, CdS, even though the utilization of these cells is presently confined in Europe on account of environmental issues with the utilization of cadmium.

Likewise, cadmium CdSe is additionally confined. Different materials that can be utilized incorporate lead sulfide, PbS and indium antimonide, InSb.

Although a semiconductor material is utilized for these photoresistors, they are simply latent gadgets since they don't have a PN junction, and this isolates them from other photodetectors like photodiodes and phototransistors.

4.3.1LDR / photoresistor symbol

The LDR symbol utilized in electronic circuits is based around the resistor circuit symbol, yet shows the light, as arrows falling on it. In this manner it follows a similar show utilized for photodiode and phototransistor circuit symbol where arrows are utilized to show the light falling on these parts.

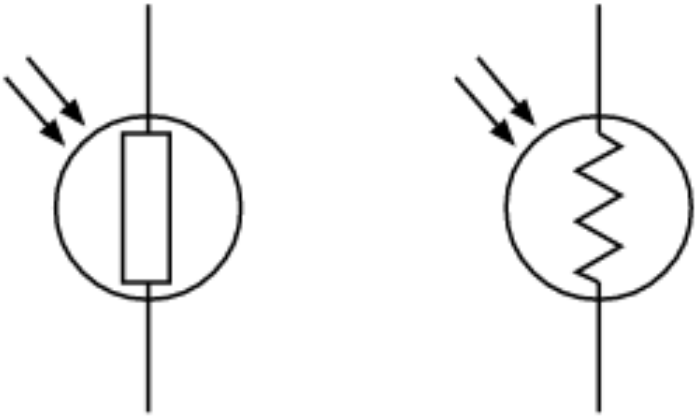

Fig 4.7 LDR Symbol

The light dependent resistor/photoresistor circuit images are appeared for both the more up to date style resistor image, for example a rectangular box and the more established crisscross line resistor circuit symbols.

4.3.2Photoresistor / LDR structure

Basically, the photoresistor is a light touchy resistor that has an even body that is presented to light.

The essential organization for a photoresistor is that demonstrated as follows:

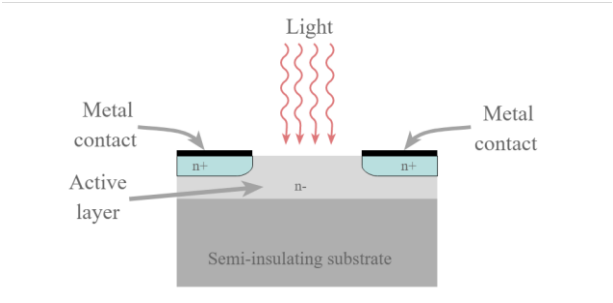

Fig 4.8 LDR Structure

The dynamic semiconductor area is ordinarily stored onto a semi-protecting substrate and the dynamic district is typically delicately doped.

In numerous discrete photoresistor gadgets, an interdigital design is utilized to expand the region of the photoresistor that is presented to light. The example is cut in the metallization on the outside of the dynamic region and this lets the light through. The two metallize territories go about as the two contacts for the resistor. This territory must be made generally huge on the grounds that the opposition of the contact to the dynamic region should be limited. This type of structure is widely used for many small photoresistors or light dependent resistors that are seen. The interdigital pattern is quite recognisable.

4.3.3 Types of photoresistor

Light needy resistors, LDRs or photoresistors can be categorized as one of two sorts or classifications:

Intrinsic photoresistors: Intrinsic photoresistors utilize un-doped semiconductor materials including silicon or germanium. Photons fall on the LDR energize electrons moving them from the valence band to the conduction band. Thus, these electrons can direct power. The more light that falls on the gadget, the more electrons are freed and the more prominent the degree of conductivity, and this outcomes in a lower level of opposition.

Extrinsic photoresistors: Extrinsic photoresistors are fabricated from semiconductor of materials doped with contaminations. These pollutants or dopants make another energy band over the current valence band. Accordingly, electrons need less energy to move to the conduction band considering the more modest energy hole.

Despite the sort of light ward resistor or photoresistor, the two kinds display an expansion in conductivity or fall in obstruction with expanding levels of occurrence light.

4.3.4 LDR frequency dependence

The sensitivity of photoresistors is appeared to change with the wavelength of the light that is affecting the delicate territory of the device. The impact is extremely checked, and it is discovered that on the off chance that the frequency is outside a given reach, at that point there is no recognizable impact.

Device produced using various materials react distinctively to light of various wavelength, and this implies that the diverse hardware parts can be utilized for various applications.

It is likewise discovered that extraneous photoresists will in general be touchier to longer wavelength light and can be utilized for infrared. Anyway, when working with infrared, care should be taken to evade heat develop caused however he is exhilarating impact of the radiation.

4.3.5Photoresistor applications

Photoresistors are found in various applications and can be seen in a wide range of electronic circuit plans. They have an exceptionally straightforward structure, and they are ease and rough gadgets. They are broadly utilized in a wide range of things of electronic hardware and circuit plans including photographic light meters, fire or smoke cautions just as robber alerts, and they additionally discover utilizes as lighting controls for streetlights.

Extraneous photoresistors are given affectability to longer wavelength and therefore they are famous in different electronic circuit plans as data red photodetectors. Photoresistors can likewise be utilized to identify atomic radiation.

4.4 16x2 LCD

The term LCD represents Liquid Crystal Display. It is one sort of electronic display module utilized in a broad scope of uses like different circuits and devices like cell phones, mini-computers, PCs, TV sets, and so on These presentations are predominantly favored for multi-fragment light-transmitting diodes and seven sections. The primary advantages of utilizing this module are reasonable; basically programmable, liveliness, and there are no constraints for showing custom characters, exceptional and even activities, and so forth

4.4.1LCD 16×2 Pin Diagram

The 16×2 LCD pinout is shown below.

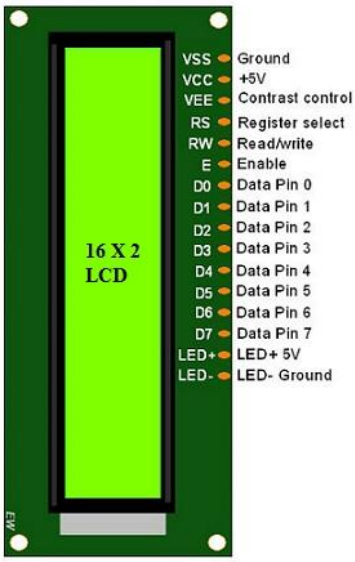

Fig 4.9 LCD Pin out

- Pin1 (Ground/Source Pin): This is a GND pin of show, used to associate 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 interface the supply pin of the power source.
- Pin3 (V0/VEE/Control Pin): This pin directs the distinction of the display, used to interface an variable POT that can supply 0 to 5V.
- Pin4 (Register Select/Control Pin): This pin switches among command or data register, used to associate a microcontroller unit pin and acquires either 0 or 1(0 = data mode, and 1 = command mode).
- Pin5 (Read/Write/Control Pin): This pin switches the display among the read or composes activity, and it is associated with 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 cycle, and it is associated with the microcontroller unit and continually held high.
- Pins 7-14 (Data Pins): These pins are utilized to send data to the display. These pins are associated in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, just four pins are associated with the microcontroller unit like 0 to 3, though in 8-wire mode, 8-pins are associated with 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.

4.4.2 Features of LCD16x2

- The operating voltage of this LCD is 4.7V-5.3V.
- It includes two rows where each row can produce 16-characters.
- The utilization of current is 1mA with no backlight.
- Every character can be built with a 5×8-pixel box.
- The alphanumeric LCDs alphabets & numbers.
- Is display can work on two modes like 4-bit & 8-bit.
- These are obtainable in Blue & Green Backlight.
- It displays a few custom generated characters.

4.4.3 Registers of LCD

A 16×2 LCD has two registers like data register and command register. The RS (register select) is essentially used to change starting with one register then onto the next. At the point when the register set is '0', at that point it is known as command register. Likewise, when the register set is '1', at that point it is known as data register.

Command Register: The principle capacity of the order register is to store the directions of order which are given to the presentation. So that predefined errands can be performed, for example, clearing the display, instating, set the cursor spot, and display control. Here command processing can happen inside the register.

Data Register: The primary capacity of the information register is to store the data which is to be shown on the LCD screen. Here, the ASCII estimation of the character is the data which is to be shown on the screen of LCD. At whatever point we send the data to LCD, it communicates to the information register, and afterward the cycle will be beginning there. At the point when register set =1, at that point the information register will be chosen.

5.Working of the Project

After making the connections and uploading the code to the Arduino using the software from the computer. The Arduino loads the program with RTC mode ON to check this we can use the LCD.

When the mode is RTC the LCD displays the time because of the coin cell present in the RTC module even if the whole circuit is switched off the module will store the time and date so when the circuit is turned on the Arduino will retrieve the data from the RTC module.

Since RTC module is time based the program compares the time with the RTC module is the time is matched the LEDs are turned ON if the time is not matching then the LEDs remain OFF.

If the user wants to make it fully automated there is a push button connected to the Arduino external interrupt pin 2. when the button is pushed the Arduino will automatically switch to LDR mode. When it's triggered the mode switch can be seen on the LCD.

When the Arduino is in LDR mode basically it reads the values of the LDR from A3 pin based on the values of the LDR the Arduino will adjust the brightness of the LED. The brightness of the LED is shown on the LCD.

In order to switch back to RTC mode the user must just push the button again. This lets the user to have the freedom to choose between any mode since both produce similar accurate results.

SOFTWARE REQUIREMENTS:

- Arduino IDE for writing the code and uploading.

6.Code for the Project

```
#include <Wire.h>
#include <LiquidCrystal.h>
#include "RTClib.h"
#define ON  0
#define OFF 1
DateTime now;

RTC_DS3231 rtc;
LiquidCrystal lcd(8, 7, 6, 5, 4, 3); // (rs, e, d4, d5, d6, d7)

const int buttonPin = 2;
const int led=11;
int nob = A3;
int val = 0;
int val1 = 0;
int path=1;
int a=1;
int previousState = HIGH;
unsigned int previousPress;
volatile int buttonFlag;
int buttonDebounce = 20;

int on_hour=1;
int on_minute=1;
int on_second=10;

int off_hour=23;
```

```
int off_minute=57;
int off_second=10;

int c_hour=0;
int c_minute=0;
int c_second=0;

int onOrOffFlag = ON;

void showDate(void);
void showTime(void);
void showDay(void);

void loadHandler(int, int , int , int , int , int ,  int , int , int );

typedef struct userTime
{
   int temp_hour;
   int temp_minute;
   int temp_second;
}userTime_t;
unsigned char checkLessThanOrEqual(userTime_t , userTime_t);

void setup ()
{
  Serial.begin(9600);
  lcd.begin(16,2);

  pinMode(buttonPin, INPUT_PULLUP);
```

```
 pinMode(led,OUTPUT);
 attachInterrupt(digitalPinToInterrupt(buttonPin), button_ISR, CHANGE);
 if (! rtc.begin())
 {
  Serial.println("Couldn't find RTC Module");
  while (1);
 }
 if (rtc.lostPower())
 {
  Serial.println("RTC lost power, let's set the time!");
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
 }
 rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
}
void loop ()
{
 if(path)
  {
   if(a==1)
   {
     lcd.setCursor(0,0);
      lcd.print("      RTC     ");
      lcd.setCursor(0,1);
      lcd.print("    MODE ON    ");
      delay(2000);
      a=0;
   }
   now = rtc.now();
   showTime();
```

```
   c_hour=now.hour();
   c_minute=now.minute();
   c_second=now.second();
   loadHandler( on_hour, on_minute, on_second, off_hour, off_minute, off_second,
c_hour, c_minute, c_second);
   delay(1000);
  }
 else
 {
     if(a==0)
   {
     lcd.setCursor(0,0);
      lcd.print("      LDR      ");
      lcd.setCursor(0,1);
      lcd.print("     MODE ON    ");
      delay(2000);
      a=1;
   }
     val = analogRead(nob);
     if(val>300 && val<450)
     {
      lcd.setCursor(0,0);
      lcd.print("      30%      ");
      lcd.setCursor(0,1);
      lcd.print("   Brightness   ");
      analogWrite(led, 400);
     }
     else if(val>450 && val<550)
     {
      lcd.setCursor(0,0);
```

```
      lcd.print("      60%     ");
      lcd.setCursor(0,1);
      lcd.print("   Brightness   ");
      analogWrite(led, 600);
     }
     else if(val>550 && val<600)
     {
      lcd.setCursor(0,0);
      lcd.print("     100%     ");;
      lcd.setCursor(0,1);
      lcd.print("   Brightness   ");
      analogWrite(led, 1023);
     }
     else if(val<300)
     {
      lcd.setCursor(0,0);
      lcd.print("      0%     ");
      lcd.setCursor(0,1);
      lcd.print("   Brightness   ");
      analogWrite(led, 0);
     }
   }
}
 void showTime()
 {
  lcd.setCursor(0,0);
  lcd.print("   Time:");
  lcd.print(now.hour());
  lcd.print(':');
```

```
  lcd.print(now.minute());
  lcd.print(':');
  lcd.print(now.second());
  lcd.print("    ");
 }
void button_ISR()
{
  buttonFlag = 1;
  if((millis() - previous Press) > buttonDebounce && buttonFlag)
  {
    previousPress = millis();
    if(digitalRead(buttonPin) == LOW && previousState == HIGH)
    {
      path =! path;
      previousState = LOW;
    }
    else if(digitalRead(buttonPin) == HIGH && previousState == LOW)
    {
      previousState = HIGH;
    }
    buttonFlag = 0;
  }
}
unsigned char checkLessThanOrEqual(userTime_t a, userTime_t b)
{
    if(a.temp_hour < b.temp_hour)
        return true;
    else
    {
```

```
if ((a.temp_hour == b.temp_hour) && (a.temp_minute < b.temp_minute))
{
   return true;
}
else
{
   if(a.temp_hour > b.temp_hour)
      return false;
   else
   {
      if((a.temp_minute == b.temp_minute) && (a.temp_second < b.temp_second))
      {
         return true;
      }
      else
      {
         if(a.temp_minute > b.temp_minute)
            return false;
         else
         {
            if(a.temp_second == b.temp_second)
            {
               return true;
            }
            else
            {
               return false;
            }
         }
```

```
            }
          }
        }
      }
}
void loadHandler(int onTimeHr, int onTimeMin, int onTimeSec, int offTimeHr, int offTimeMin, int offTimeSec,  int rtcTimeHr, int rtcTimeMin, int rtcTimeSec)
{
   userTime_t in1 = {onTimeHr, onTimeMin, onTimeSec}, in2 = {offTimeHr, offTimeMin, offTimeSec}, rtc_hr = {rtcTimeHr, rtcTimeMin, rtcTimeSec}, a = {}, b = {};

   if(checkLessThanOrEqual(in1, in2))
   {
      onOrOffFlag = ON;
      memcpy(&a, &in1, sizeof(userTime_t));
      memcpy(&b, &in2, sizeof(userTime_t));
   }
   else
   {
      onOrOffFlag = OFF;
      memcpy(&a, &in2, sizeof(userTime_t));
      memcpy(&b, &in1, sizeof(userTime_t));

   }

   if((checkLessThanOrEqual(a, rtc_hr)) && (checkLessThanOrEqual(rtc_hr, b)))
   {
      if(onOrOffFlag == ON)
      {
         // Switch on the load
```

```
        digitalWrite(led,HIGH);
        lcd.setCursor(0,1);
        lcd.print("OffTime:");
        lcd.print(off_hour);
        lcd.print(':');
        lcd.print(off_minute);
        lcd.print(':');
        lcd.print(off_second);
    }
    else
    {
        // Switch off the load
         digitalWrite(led,LOW);
        lcd.setCursor(0,1);
        lcd.print(" OnTime:");
        lcd.print(on_hour);
        lcd.print(':');
        lcd.print(on_minute);
        lcd.print(':');
        lcd.print(on_second);

    }
}
else
{
    if(onOrOffFlag == ON)
    {
        // Switch off the load
        digitalWrite(led,LOW);
```

```
            lcd.setCursor(0,1);
            lcd.print(" OnTime:");
            lcd.print(on_hour);
            lcd.print(':');
            lcd.print(on_minute);
            lcd.print(':');
            lcd.print(on_second);
        }
        else
        {
            // Switch on the load
            digitalWrite(led,HIGH);
            lcd.setCursor(0,1);
            lcd.print("OffTime:");
            lcd.print(off_hour);
            lcd.print(':');
            lcd.print(off_minute);
            lcd.print(':');
            lcd.print(off_second);
        }
    }
}
```

7.Applications of the Project

- A simple project for saving power is implemented using Auto Intensity Control of Street Lights using Arduino. With slight modifications and enhancements, this project can be applicable for real time use.
- The solution to energy conservation is to eliminate time slot and introduce a system that could sense brightness environment and act accordingly so that seasonal change would not affect the intensity of streetlights. Also, LEDs should replace HID lamps due to their dimming feature, another reason are that they are more reliable.

S