

**AUTOMATION OF STREET LIGHTS USING LDR, PIR AND RTC**

##### A PROJECT REPORT

###### *Submitted by*

##### VINOTHIN T 211414104308

**YENUGU VENKATESH 211414104313**

**RITESH ASWIN RAAM S G 211414104221**

***In partial fulfilment for the award of the degree***

***Of***

**BACHELOR OF ENGINEERING**

**in**

# COMPUTER SCIENCE AND ENGINEERING

**PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.**

**ANNA UNIVERSITY: CHENNAI 600 025**

##### APRIL 2018

**ANNA UNIVERSITY: CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report“**AUTOMATION OF STREET LIGHTS USING LDR, PIR AND RTC” is** the bonafide work of **VINOTHIN T (211414104308) YENUGU VENKATESH (211414104313) RITESH ASWIN RAAM S G (211414104221)** who carried out the project work under my supervision.

**SIGNATURE                                   SIGNATURE**

Dr. S. MURUGAVALLI, M.E., Ph.D.,Mr. S.A.K. JAINULABUDEEN, M.Tech.,

**PROFESSOR     SUPERVISOR**

**HEAD OF THE DEPARTMENT          ASSISTANT PROFESSOR**

DEPARTMENT OF CSE,      DEPARTMENT OF CSE

PANIMALAR ENGINEERING COLLEGE,      PANIMALAR ENGINEERING COLLEGE,

NASARATHPETTAI,      NASARATHPETTAI,

POONAMALLEE,      POONAMALLEE,

CHENNAI-600 123.      CHENNAI-600 123.

Certified that the above candidate(s) was examined in the Anna University Project Viva-Voice Examination held on ………………..

**INTERNAL EXAMINER              EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

My sincere, humble and heartfelt thanks and praises to our honorable **Founder and Chairman** Late **Dr. JEPPIAAR, M.A., B.L., Ph.D.,** for his sincere Endeavor in educating us in his reputed institution.

We would like to express our deep gratitude to our respected **Secretary and Correspondent Dr. P. CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our **Directors Mrs. C. VIJAYA RAJESWARI, Thiru. C. SAKTHI KUMAR, M.E., and Mrs. SARANYA SREE SAKTHI KUMAR B.E.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our **Principal Dr. K. MANI, M.E., Ph.D.,** who helped us in completing the project.

We thank the **Head of the CSE Department, Dr. S. MURUGAVALLI, M.E., Ph.D.,** for the support extended throughout the project.

We would like to thank our Supervisor **Mr. S.A.K. JAINULABUDEEN, M.Tech., Assistant professor,** Project co-ordinator **Dr. N. PUGHAZENDHI, M.E., Ph.D.,** and all the faculty members of the department of CSE for their advice and encouragement for the successful completion of the project.

We take this opportunity to thank our beloved parents, friends and teachers for their constant support and encouragement.

**VINOTHIN T**

**YENUGU VENKATESH**

**RITESH ASWIN RAAM S G**

**ABSTRACT**

Smart City-based electronic product applications are gaining importance nowadays. It contains efficient urban mobility, efficient public transportation, e-governance, safety and security, smart lighting system, etc. The main focus of this work is to present a design of street light controller to provide a reduction in power consumption and wireless control which eventually will provide a reduction in the required budget of electricity for street lights. Reducing power consumption leads to a reduction in brightness of lamps. The system will work as adaptive street lighting. The main focus of this work is to present a design of street light controller to provide a reduction in power consumption and wireless control which eventually will provide a reduction in required budget of electricity for street lights. The complete system will work on RTC (Real Time  Clock) where lights will be running at 100 % intensity in peak traffic time and with reduced intensity after peak traffic time. Lights will be obscured when no action is detected and will light up when some movement is detected using PIR (Passive Infrared Sensor).During day time the LDR (Light Dependent Resistor) checks the atmospheric brightness and if it is low it makes the lights glow with full intensity.

In case of any failures on street lamps GSM (Global System for Mobile Communications ) automatically sends the message to the service department using a mobile network.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ACKNOWLEDGEMENT** | **iii** |
|  | **ABSTRACT** | **iv** |
|  | **LIST OF FIGURES** | **vii** |
|  | **LIST OF TABLES** | **viii** |
|  | **LIST OF ABBREVIATIONS** | **viii** |
| **1.** | **INTRODUCTION** | **1** |
|  | 1.4 Specification of Component | **5** |
|  | 1.4.1 LDR | **5** |
|  | 1.4.2 PIR | **6** |
|  | 1.4.3 RTC | **7** |
|  | 1.4.4 Arduino | **8** |
|  | 1.4.5 GSM Module | **17** |
| **2.** | **LITERATURE SURVEY** | **21** |
| **3.** | **SYSTEM ANALYSIS** | **31** |
|  | 3.1 Existing system | **32** |
|  | 3.2 Drawback of Existing System | **32** |
|  | 3.3 Proposed System | **32** |
|  | 3.4 Hardware Requirement | **32** |
|  | 3.5 Software Requirement | **33** |
| **4.** | **SYSTEM ARCHITECTURE** | **34** |
|  | 4.1 System Architecture Diagram | **35** |
|  | 4.2 Algorithm | **35** |
|  | 4.3 Pseudocode | **36** |
| **5.** | **CONCLUSION** | **37** |
|  | 5.1 Conclusion | **38** |
|  | 5.2 Future Enhancement | **38** |
| **6.** | **APPENDICES** | **39** |
|  | A1 Coding | **40** |
|  | A2 Screenshots | **54** |
| **7.** | **REFERENCES** | **57** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PAGE NO.** |
| 1.1 | Block diagram | 3 |
| 1.3 | Flowchart | 4 |
| 1.4.1.1 | LDR sensor | 6 |
| 1.4.2.1 | PIR sensor | 7 |
| 1.4.3.1 | RTC | 8 |
| 1.4.4.1 | Arduino | 10 |
| 1.4.4.2 | Screenshot of Arduino IDE | 12 |
| 1.4.5.1 | GSM module | 20 |
| 4.1 | System Architecture diagram | 35 |
| A3 | Lamp status between 22 and 06 | 54 |
| A4 | Lamp status between 18 and 22 | 55 |
| A5 | Screenshot of message received by Authority | 55 |
| A6 | Lamp status between 06 and 18 | 56 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PAGE NO.** |
| 1.2 | List of components | 4 |
| 2.1 | Literature Survey | 22 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **SYMBOL** | **DESCRIPTION** |
| LED | Light Emitting Diode |
| LDR | Light Dependent Resistor |
| PIR | Passive Infrared Sensor |
| RTC | Real Time Clock |
| GSM | Global System for Mobile communication |

**CHAPTER 1**

**INTRODUCTION**

**CHAPTER 1**

**INTRODUCTION**

Automatic Street Light Control System is a simple yet powerful concept. By using this system manual works are 100% removed. It automatically switches ON lights when the sunlight goes below the visible region of our eyes. This is done by a sensor called Light Dependant Resistor (LDR) which senses the light actually like our eyes. It automatically switches OFF lights whenever the sunlight comes, visible to our eyes. By using this system energy consumption is also reduced because nowadays the manually operated street lights are not switched off even the sunlight comes and also switched on earlier before sunset. The working of relay is also known. **Street light controllers** are smarter versions of the mechanical or electronic timers previously used for [street light](http://en.wikipedia.org/wiki/Street_light) ON-OFF operation. They come with [energy conservation](http://en.wikipedia.org/wiki/Energy_conservation) options like twilight saving, staggering or [dimming.](http://en.wikipedia.org/wiki/Dimmer) Also many street light controllers come with an real time clock to give the best ON-OFF time and energy saving. Automatic Street Light Control System is a simple and powerful concept, which uses sensors as a switch to switch ON and OFF the street light automatically. By using this system manual works are removed. It automatically switches ON lights when the sunlight goes below the visible region of our eyes. It automatically switches OFF lights under illumination by sunlight. This is done by a sensor called Light Dependant Resistor (LDR) which senses the light actually like our eyes .By using this system energy consumption is also reduced because now-a-days the manually operated street lights are not switched off properly even the sunlight comes and also not switched on earlier before sunset. In sunny and rainy days, ON time and OFF time differ significantly which is one of the major disadvantage of using timer circuits or manual .

A **street light**, **lamppost**, **street lamp**, **light standard**, or **lamp standard** is a raised source of [light](http://en.wikipedia.org/wiki/Light) on the edge of a [road](http://en.wikipedia.org/wiki/Road) or walkway, which is turned on or lit at a certain time every night. Modern lamps may also have light-sensitive [photocells](http://en.wikipedia.org/wiki/Photocell) to turn them on at [dusk,](http://en.wikipedia.org/wiki/Dusk) off at [dawn,](http://en.wikipedia.org/wiki/Dawn) or activate automatically in dark [weather.](http://en.wikipedia.org/wiki/Weather) In older lighting this function would have been performed with the aid of a [solar dial.](http://en.wikipedia.org/wiki/Solar_dial) It is not uncommon for street lights to be on poles which have wires strung between them, or mounted on [utility poles](http://en.wikipedia.org/wiki/Utility_pole) The complete system will work on RTC (Real Time  Clock) where lights will be running at 100 % intensity in peak traffic time and with reduced intensity after peak traffic time.Lights will be obscured when no action is detected and will light up when some movement is detected using PIR (Passive Infrared Sensor).In case of any failures on street lamps GSM (Global System for Mobile Communications ) automatically sends the message to the service department using a mobile network.Automatic Streetlight needs no manual operation of switching ON and OFF. The system itself detects whether there is need for light or not. When darkness rises to a certain.

**1.1 BLOCK DIAGRAM**

Power Supply

PIR Sensor

LED array

Arduino UNO

GSM

Mobile network

LDR Sensor

RTC Module

# 

# 1.2 LIST OF COMPONENTS

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **PARTS** | **RANGE** | **QUANTITY** |
| 1. | LDR |  | 4 |
| 2. | PIR |  | 4 |
| 3. | RTC |  | 1 |
| 4. | GSM MODULE |  | 1 |
| 5. | ARDUINO BOARD |  | 1 |
| 6. | RESISTORS | 1K ohm | 4 |
| 7. | LED |  | 8 |
| 8. | POWER SUPPLY | 5V | 1 |

**1.3FLOWCHART**

if(22>hour<=6)

if(18>hour<=22)

if(06>hour<=18)

if vehicle detected

]

PIR

[

Lamp glows on full

intensity

if lamp doesn"t

glows

A message is send

using GSM module

if cloud is dark

]

[

LDR

Lamp glows on partia

intensty

Lamp does not glows

lamp glows on full

intensity

lamp glows on 20%

intensity

RTC

else

else

**1.4 SPECIFICATION OF COMPONENT**

**1.4.1 LDR**

Type : Passive

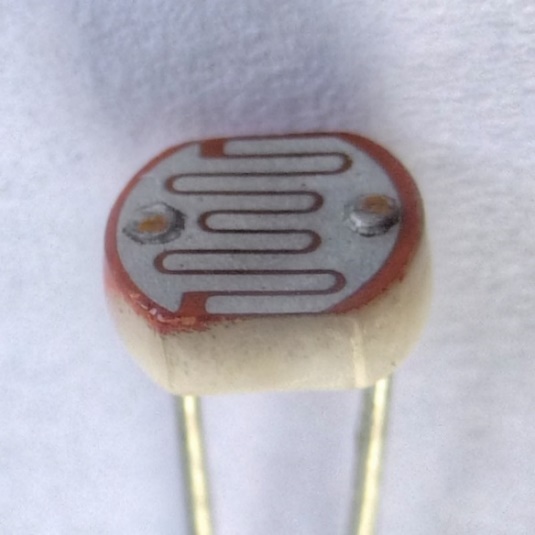
Working Principle : Photoconductivity

Electronic Symbol

[](https://en.wikipedia.org/wiki/File:Photoresistor_symbol.svg)

A **photoresistor** (or **light-dependent resistor**, **LDR**, or **photo-conductive cell**) is a light-controlled variable [resistor](https://en.wikipedia.org/wiki/Resistor). The [resistance](https://en.wikipedia.org/wiki/Electrical_resistance) of a photoresistor decreases with increasing incident light intensity; in other words, it exhibits [photoconductivity](https://en.wikipedia.org/wiki/Photoconductivity). A photoresistor can be applied in light-sensitive detector circuits, and light-activated and dark-activated switching circuits.

A photoresistor is made of a high resistance [semiconductor](https://en.wikipedia.org/wiki/Semiconductor). In the dark, a photoresistor can have a resistance as high as several megaohms (MΩ), while in the light, a photoresistor can have a resistance as low as a few hundred ohms. If incident light on a photoresistor exceeds a certain [frequency](https://en.wikipedia.org/wiki/Frequency), [photons](https://en.wikipedia.org/wiki/Photon) absorbed by the semiconductor give bound [electrons](https://en.wikipedia.org/wiki/Electron) enough energy to jump into the [conduction band](https://en.wikipedia.org/wiki/Conduction_band). The resulting free electrons (and their [hole](https://en.wikipedia.org/wiki/Electron_hole) partners) conduct electricity, thereby lowering [resistance](https://en.wikipedia.org/wiki/Electrical_resistance). The resistance range and sensitivity of a photoresistor can substantially differ among dissimilar devices. Moreover, unique photoresistors may react substantially differently to photons within certain wavelength bands.



LDR 1.4.1.1

**1.4.2 PIR**

A PIR-based motion detector is used to sense movement of people, animals, or other objects. They are commonly used in automatically-activated [lighting](https://en.wikipedia.org/wiki/Lighting) systems. They are commonly called simply "PIR", or sometimes "PID", for "passive infrared detector”.

An individual PIR sensor detects changes in the amount of infrared radiation impinging upon it, which varies depending on the temperature and surface characteristics of the objects in front of the sensor. When an object, such as a [human](https://en.wikipedia.org/wiki/Human), passes in front of the background, such as a [wall](https://en.wikipedia.org/wiki/Wall), the temperature at that point in the sensor's field of view will rise from [room temperature](https://en.wikipedia.org/wiki/Room_temperature) to [body temperature](https://en.wikipedia.org/wiki/Body_temperature), and then back again. The sensor converts the resulting change in the incoming infrared radiation into a change in the output voltage, and this triggers the detection. Objects of similar temperature but different surface characteristics may also have a different infrared emission pattern, and thus moving them with respect to the background may trigger the detector as well.



**PIR 1.4.2.1**

**1.4.3 RTC**

A **real-time clock** (**RTC**) is a [computer](https://en.wikipedia.org/wiki/Computer) [clock](https://en.wikipedia.org/wiki/Clock) (most often in the form of an [integrated circuit](https://en.wikipedia.org/wiki/Integrated_circuit)) that keeps track of the current [time](https://en.wikipedia.org/wiki/Time) .

Although the term often refers to the devices in [personal computers](https://en.wikipedia.org/wiki/Personal_computer), [servers](https://en.wikipedia.org/wiki/Server_(computing)) and [embedded systems](https://en.wikipedia.org/wiki/Embedded_system), RTCs are present in almost any electronic device which needs to keep accurate time. A common RTC used in [single-board computers](https://en.wikipedia.org/wiki/Single-board_computer) is the DS1307.

RTCs often have an alternate source of power, so they can continue to keep time while the primary source of power is off or unavailable. This alternate source of power is normally a [lithium battery](https://en.wikipedia.org/wiki/Lithium_battery) in older systems, but some newer systems use a [supercapacitor](https://en.wikipedia.org/wiki/Supercapacitor), because they are rechargeable and can be [soldered](https://en.wikipedia.org/wiki/Soldering). The alternate power source can also supply power to [battery backed RAM](https://en.wikipedia.org/wiki/Nonvolatile_BIOS_memory).

Most RTCs use a [crystal oscillator](https://en.wikipedia.org/wiki/Crystal_oscillator), but some use the [power line frequency](https://en.wikipedia.org/wiki/Utility_frequency). In many cases, the oscillator's frequency is 32.768 kHz. This is the same frequency used in [quartz clocks and watches](https://en.wikipedia.org/wiki/Quartz_clock), and for the same reasons, namely that the frequency is exactly 215 cycles per second, is a convenient rate to use with simple binary counter circuits.

Many commercial RTC ICs are accurate to less than 5 parts per million.[[10]](https://en.wikipedia.org/wiki/Real-time_clock#cite_note-10) In practical terms, this is good enough to perform [celestial navigation](https://en.wikipedia.org/wiki/Celestial_navigation), the classic task of a [chronometer](https://en.wikipedia.org/wiki/Marine_chronometer). In 2011, [Chip-scale atomic clocks](https://en.wikipedia.org/wiki/Chip-scale_atomic_clock) were invented. Although more expensive, they keep time within 100 nanoseconds.



**RTC 1.4.3.1**

**1.4.4 ARDUINO**

A microcontroller often serves as the “brain” of a mechatronic system. Like a mini, self-contained computer, it can be programmed to interact with connected hardware and/or a user, much like a PC connected to a small network of hardware. As the computer industry has evolved, so has the technology associated with microcontrollers. Every year microcontrollers become much faster, have more memory, and extend their input and output feature sets, all the while becoming even cheaper and easier to use.

The development board used in this course is called the *Arduino*. The name Arduino is used to denote the hardware board, the software development environment, its library of easy-to-use software functions, and/or the layout standard of the original version's connection headers, depending on the context.

The name “Arduino” is a copyright held by the original team based in Italy that originally built the hardware, the IDE (integrated development environment) and the software libraries. It was designed to be a platform for *physical computing* usable primarily by non-programmers (artists, hobbyists, teachers, students, etc.). Another goal of the project was that all of the hardware designs (except those for the chips themselves) as well as the software tools are open-source – free to copy and distribute without any licensing restrictions. By making an easy-to-use, easy-to-make platform, they hoped that it would enable a wide variety of interactive and creative uses by an even greater variety of users. Its popularity and use has spread much wider than originally anticipated.

Another attraction of the Arduino platform is that there is almost no configuration necessary to get started. Once you have downloaded and installed the software on your computer, you can load, compile, download, and run a sample program.

Arduino programs are called *sketches*, reflecting the vocabulary of the artist community. You will still often hear them called *programs* (including here) and not *sketches*, but at least now when you hear “sketches

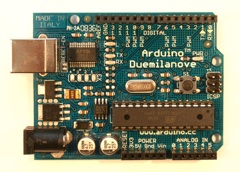
Besides its ease-of-use, the Arduino was one of the first complete development systems to cost around $30. The current, standard Arduino is still priced around $30, but small, barebones Arduino-compatible boards can be purchased for $11. In fact, the hardware platform is simple enough that even cheaper, barebones versions can be handmade using a variety of construction methods.

Another advantage is that the Arduino development environment can be run on Windows, Linux and MacOS for no cost other than for the hardware. All of the software is freely downloadable in one bundle from [www.arduino.cc](http://www.arduino.cc/), the website that is ground-zero for all-things-Arduino. This website should earn a prominent bookmark in your web browser.

The Arduino boards that we are using are from the group of “standard” Arduinos, in particular, the Arduino Duemilanove and the current Arduino Uno.

What characterizes this class of Arduino boards is the layout of the four rows of female headers across the top and bottom, and the single 2x3 header along the right edge. Any Arduino following this standard layout can connect to a large number of expansion PCBs (printed circuit boards) that plug on top of the Arduino called *shields*. Most Arduino-family boards follow this classic layout, but some do not, mostly for size or cost reasons.

The analog output pins are located on the bottom-right six-pin female



ATmega328P MCU

Reset button

Analog pins header

Digital pins header

Power-Ground header

Pin 13 LED

USB connector

Barrel jack

ARDUINO 1.4.4.1

The Arduino board comes with a single LED, often called the *Pin 13 LED* because it is electrically connected to Digital Pin 13. This LED is the board's only built-in indicator accessible to programs. As you will soon see, you may bring more of your own input and output devices to the party and connect them yourself. A virgin Arduino board comes pre-programmed with a sample sketch called *Blink* that flashes the Pin 13 LED once per second. This feature is handy because you can easily check the general health of your new board simply by plugging it into USB, and the flashing LED demonstrates that the board is mostly working.On the board is also a single reset button that causes the program to restart when pressed, and two more LEDs that flash as data is passed back-and-forth over the USB connection.

### General precautions

As with any piece of non-consumer-grade electronics equipment, you have to be somewhat careful to avoid frying electronic components. Some types of components, mostly semiconductor ICs (integrated circuits) *can* be delicate and easily damaged by electrical shorts and static discharges. In general, before touching any circuit board with integrated circuits (ICs), make sure that you have dissipated any static charge that you may have accumulated on your body. The best way to do this is by using an ESD wrist strap that has been connected to a good earth ground and by placing your circuit board on a grounded ESD mat.

If you don’t have this or similar ESD equipment (we don't), it's generally sufficient to touch a grounded metal surface like the metal frame of the lab bench before you handle the circuit board. Static charge is especially easy to accumulate during the dry winter months, simply by walking across the floor.

Handle all PCBs by their edges, not by grasping the faces of the printed circuit board or its components. Finally, always wash your hands after handling electronics because lead and other heavy metals in the components and solder connecting them are hazardous in sufficient quantities.

### Getting to know the Arduino website:

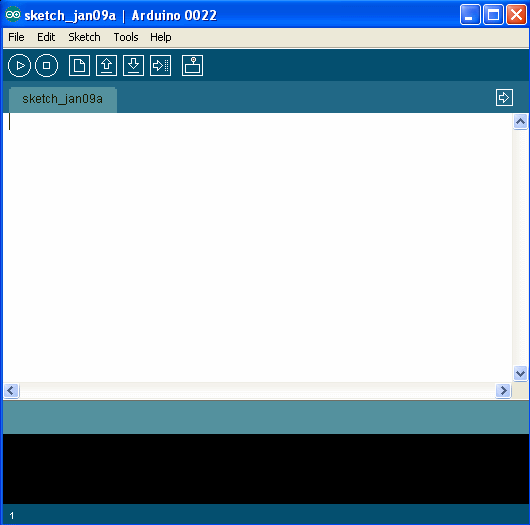
The main Arduino website is [www.arduino.c](http://www.arduino.com/)c. From here you can [download the software bundle](http://arduino.cc/en/Main/Software) that runs on your PC, find the [reference pages for programming](http://arduino.cc/en/Reference/HomePage) with the Arduino libraries, [learn to use the IDE](http://arduino.cc/en/Reference/HomePage), and search the central repository for community-contributed software modules in the [Playground](http://www.arduino.cc/playground/). The [Learning](http://arduino.cc/en/Tutorial/HomePage) section of the website is probably the first place you should start exploring after [Getting Started](http://arduino.cc/en/Guide/HomePage).

### Downloading and installing the Arduino software

Depending on which version of the Arduino board you are using, the installation procedures vary slightly, but are very easy,*if you follow the installation instructions.*

Note that different drivers are required for the most recent Arduino Uno versus the ArduinoDuemilanove, so if you're planning on using both on one computer, both driver installation procedures will have to be performed.

### Configuring the Arduino IDE

At this point you should connect the Arduino to the computer with the USB cable. After initial set-up, the computer should issue its ascending-beep to acknowledge that the board has been connected. After launching the Arduino IDE,

1.4.4.2 Screenshot of Arduino IDE

Now you have to configure the IDE so that it can communicate with the Arduino board.

Please note that *every time that you connect an Arduino over USB, you may have to go through the same configuration procedure* (at least with pre-Uno boards). Additionally, if you cause the Arduino to hard-reset (for instance, because you created a temporary short), that will also reset the USB connection. Hard resets are functionally equivalent to disconnecting and reconnecting the cable or board. Fortunately, pressing the reset button on the Arduino board (or shields) will *not* reset the USB connection.

There are only two settings that need to be configured, and one of them rarely changes. The first setting is the *board type*. If you are using an Uno board, under Tools->Board select *Arduino Uno.* This option is probably already the default, and this setting is retained from the last time the Arduino IDE was run.

If you are using a Duemillanove board, under Tools->Board select *ArduinoDuemilanove or Nano w/ATmega328*. Not setting the board type will cause program downloads to the board to fail.

The other option, under Tools->Serial Port, specifies the COM port that the USB driver should use to communicate with the Arduino. This one is annoying because every time that your USB configuration changes, your Arduino board may move to another COM port. There are three ways to figure out which COM port to use:

1) try to download to each of them  
 2) For Windows only – grope around in *Start->Control Panel->System->Hardware->Device Manager->Ports*, looking for an entry labeled *USB Serial Port[[1]](#footnote-1),*

3) remember which USB connections aren't connected to the Arduino (but to mice, etc.) and use the one that changes

You should see the following two areas in the bottom part of the IDE window, a blue-gray status message section, and a detailed message/log section:

In succession, you should see the following series of messages:

1. During the compilation process, the status message “Compiling...”
2. After successful compilation, the logged message “*Binary sketch size: ...*”
3. During upload, the status message “*Uploading to I/O Board...*” followed by a few-seconds-long flashing of the TX/RX LEDs on the Arduino (next to the Pin 13 LED). If either your COM port or board selection is incorrect, the upload will time-out, and you will likely see an error from the “avrdude” program in the logged messages window.
4. After uploading, the status message “*Done uploading*.”

If all goes well you should see the Pin 13 LED flashing with a period of two seconds.

**SUMMARY**

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

**POWER**

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

**MEMORY**

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](http://www.arduino.cc/en/Reference/EEPROM)).

**INPUT AND OUTPUT**

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode()](http://arduino.cc/en/Reference/PinMode), [digitalWrite()](http://arduino.cc/en/Reference/DigitalWrite), and [digitalRead()](http://arduino.cc/en/Reference/DigitalRead) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms.

**COMMUNICATION**

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer.

The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, [on Windows, a .inf file is required](http://arduino.cc/en/Guide/Windows#toc4). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board.

The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](http://www.arduino.cc/en/Reference/SoftwareSerial) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](http://arduino.cc/en/Reference/Wire) for details. For SPI communication, use the [SPI library](http://arduino.cc/en/Reference/SPI).

**1.4.5 GSM MODULE**

This GSM Modem can accept any GSM network operator SIM card and act just like a mobile phone with its own unique phone number. Advantage of using this modem will be that you can use its RS232 port to communicate and develop embedded applications. Applications like SMS Control, data transfer, remote control and logging can be developed easily.

GSM/GPRS MODEM is a class of wireless MODEM devices that are designed for communication of a computer with the GSM and GPRS network. It requires a SIM (Subscriber Identity Module) card just like mobile phones to activate communication with the network. Also they have IMEI(International Mobile Equipment Identity) number similar to mobile phones for their identification. A GSM/GPRS MODEM can perform the following operations:

1. Receive, send or delete SMS messages in a SIM.

2. Read, add, search phonebook entries of the SIM.

3. Make, Receive, or reject a voice call.

The MODEM needs AT commands, for interacting with processor or controller, which are communicated through serial communication. These commands are sent by the controller/processor. The MODEM sends back a result after it receives a command. Different AT commands supported by the MODEM can be sent by the processor/controller/computer to interact with the GSM and GPRS cellular network.

#### Booting the GSM Module!

**1.** Insert the SIM card to GSM module and lock it.

**2.** Connect the adapter to GSM module and turn it ON!

**3.** Now wait for some time (say 1 minute) and see the blinking rate of ‘status LED’  or ‘network LED’ (GSM module will take some time to establish connection with mobile network)

**4.** Once the connection is established successfully, the status/network LED will blink continuously every 3 seconds. You may try making a call to the mobile number of the sim card inside GSM module. If you hear a ring back, the gsm module has successfully established network connection.

Okay! Now let’s see how to connect a gsm module to Arduino!

#### Connecting GSM Module to Arduino

There are two ways of connecting GSM module to arduino. In any case, the communication between Arduino and GSM module is serial. So we are supposed to use serial pins of Arduino (Rx and Tx). So if you are going with this method, you may connect the Tx pin of GSM module to Rx pin of Arduino and Rx pin of GSM module to Tx pin of Arduino. You read it right ? **GSM Tx –> Arduino Rx** and **GSM Rx –> Arduino Tx**. Now connect the ground pin of arduino to ground pin of gsm module! So that’s all! You made 3 connections and the wiring is over! Now you can load different programs to communicate with gsm module and make it work.

 The problem with this connection is that, while programming Arduino uses serial ports to load program from the Arduino IDE. If these pins are used in wiring,  the program will not be loaded successfully to Arduino. So you have to disconnect wiring in Rx and Tx each time you burn the program to arduino. Once the program is loaded successfully, you can reconnect these pins and have the system working!

To avoid this difficulty, I am using an alternate method in which two digital pins of arduino are used for serial communication. We need to select two**PWM enabled pins of arduino** for this method. So I choose pins**5** and **6** (which are PWM enabled pins). This method is made possible with the **[SoftwareSerial Library](http://arduino.cc/en/Reference/softwareSerial)** of Ardunio. SoftwareSerial is a library of Arduino which enables serial data communication through other digital pins of Arduino. The library replicates hardware functions and handles the task of serial communication.

I hope you understood so far!  Lets get to the circuit diagram! So given below is the circuit diagram to connect gsm module to arduino – and hence use the circuit to send sms and receive sms using arduino and gsm modem.



**GSM MODULE 1.4.5.1**

**CHAPTER 2**

**LITERATURE SURVEY**

**CHAPTER 2**

**LITERATURE SURVEY**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Authors** | **Name** | **Description** |
| 1. | [Noriaki Yoshiura](https://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Noriaki%20Yoshiura.QT.&newsearch=true)  [Yusaku Fujii](https://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Yusaku%20Fujii.QT.&newsearch=true)  [Naoya Ohta](https://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Naoya%20Ohta.QT.&newsearch=true) | Smart street light system looking like usual street lights based on sensor networks | In this case light turns on when its needed otherwise not Whenever someone see street lights, they turn on whenever no one see street light turned off. The lights turn on before pedestrians and vehicles come and turn off or reduce brightness when there is no one. It will be difficult for pedestrians and drivers of vehicles to distinguish our smart street lights and the conventional street lights because our street lights all turn on before they come. |
| 2. | Chetna Badgaiyan , Palak Sehgal | Smart Street Lighting System | New method has been proposed in which Zigbee and WSN devices and PIR sensors in which in case of failure message is sent using zigbee devices and pir detects the moment of people. Proposed system has high expense and low cost wiring. |
| 3. | Samir A. Elsagheer Mohamed | Smart Street Lighting Control and Monitoring System for Electrical Power Saving by Using VANET | New method of system is used in this Vanet technology is used in this where presence of vehicles, their locations, their directions and their speeds in real time , they have concluded that its more efficient It can extend the lifetime of the lamps and warn the maintenance traffic authority upon failure detection in any place of the streets. |
| 4. | Prof. K.Y.Rajput , Gargeyee Khatav , Monica Pujari, Priyanka Yadav | “Intelligent Street Lighting System Using Gsm | In this paper Intelligent street lighting system is described that integrates new technologies offering ease of maintenance and energy savings. The proposed system is appropriate for street lighting in remote as well as urban areas where traffic is low at times. Along with energy saving it also tackles with the problem of power theft .It is capable of taking corrective actions in case of unprecedented events of climatic changes. |
| 5. | Power control research group | Power Control Research Group  Smart Street Light System | They present a new approach based on IR sensor and the idea of this project can be implemented in a large scale in many big cities,where most of the street lights consuming useful power, It is a step forward to allot the power generated in a much better fashion. |
| 6. | Abdul Latif Saleem,  Raja Sagar R,  Sachin Datta NS,  Sachin H,  Usha M S | Street Light Monitoring And Control System | This paper aims at designing and executing the advanced development I embedded system for energy saving of street lights, in this 2 sensors LDR and photoelectric sensor is used on the street. The microcontroller PIC16F877A is used as brain to control the street light system. |
| 7. | Jayesh Patel,  Sumit Tayade,  Manjusha Pattadkal,  Neelambari Nawale,  Harsha Bhute | Smart Street Light Control System | In this model LDR is placed at the top of the lamp in case of failure microcontroller communicates with the XBEE devices and then LDR is used to check the day night status and status of the lamp is checked to check whether it is working properly. |
| 8. | R Abinaya,  Kuluvan,  Hariharan | An intelligent Street Light system based on Piezoelectric sensor networks. | New system is proposed in this sensor is placed ahead of the lamp where the day and night is detected using LDR sensors and then piezoelectric sensor is placed on either side of the road in case of any movement in the street is detected it send command to microcontroller to change the intensity of the lamp, it saves a lot of energy. |
| 9. | P Kavita A. Bajaj , Tushar S. Mote | Intelligent Street Lightening System” | This project describes an intelligent street lighting system using LED supplied by solar energy and with a control system for efficient management. This features switching ON the lights only when necessary increasing the energy saving and Lamps lifetime. The wireless nature of the control system using ZigBee offers very less maintenance and flexible, extendable and fully adaptable user needs in rural and urban areas. |
| 10. | Amul  Shravan kumar Jalan | A Survey on Automatic Street Lightning System on Indian Streets Using Arduino | The system used here is a closed loop on-off system. Controlling lighting system by means of LDR and Arduino together on Indian streets is relatively a new concept. Still today research has been done only on street light system based on Passive Infrared receiver and few are LDR based but they are controlled by means of timers and analog circuits. Some were controlled by wireless GSM/GUI networks which are too costly and not affordable.. |
| 11. | Gouthami. C , Santosh. C ,  A. Pavan Kumar, Karthik. A, Ramya.K.R | Design and Implementation of Automatic Street Light Control System using Light Dependent Resistor | Authors proposed automatic control using LDR helps to save a large amount of electric power which is wasted in conventional street lighting system. The automatic switching operation observed using the developed control circuit is found to be very efficient and the maintenance cost is very less. The circuit controls the turning ON or OFF the street light. The street lights have been successfully controlled by microcontroller. With commands from the controller the lights will be ON when it's dark. Furthermore the drawback of the street light system by just using timer controller has been overcome, where the system depends on both timer and LDR sensor. |
| 12. | Abdul Latif Saleem, Raja Sagar R,  Sachin Datta N S,Sachin H S,  Usha M S | Street Light Monitoring and Control System | Author proposed a technique in which 2- way communication can be possible. Technical solution for implementation of wireless intelligent smart street lighting system is made easier. It provides a low cost infrastructure for managing street lighting system. Single point controlling of street lights is made possible. Energy consumption can be controlled making it eco-friendly in usage. |
| 13. | Sindhu.A.M, Jerin George, Sumit Roy, Chandra J | Smart Streetlight Using IR Sensors | By using Smart Street light, one can save surplus amount of energy which is done by replacing sodium vapor lamps by LED and adding an additional feature for security purposes. It prevents unnecessary wastage of electricity, caused due to manual switching of streetlights when it’s not required. It provides an efficient and smart automatic streetlight control system with the help of IR sensors. It can reduce the energy consumption and maintains the cost. The system is versatile, extendable and totally adjustable to user needs |
| 14. | Mr. A.  Syed Noor-ul-Hassan Bukhari,  Syed Zulfiqar Haider Bukhari,  Ahmad Faheem Alam,  Javed Iqba | Automatic street Light ControlSystem using LDR and IC555 sensor | This project is aimed at designing and implementation ofan automatic system in the streets. This project can beimplemented on large scale. It requires the initial cost only fordesigning and installation and not for utilization. Hence, suchsystems are very useful for the government to reduce theutilization of electric power. Therefore, such systems are onceimplemented on a large scale can bring significant reductionof the power consumption caused by street lights. Thisinitiative will help the government to save this energy andmeet the domestic and industrial needs. |

Table 2.1 Literature Survey

**CHAPTER 3**

**SYSTEM ANALYSIS**

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

In this Piezoelectric sensors are used to detect the vehicle movements and accordingly switching on the lights ahead of it, it is placed in between of the roads in which if a when a vehicle passes it detects and sends it to micro controller.

**3.2 DRAWBACKS OF EXISTING SYSTEM**

In this case in case of any failure occurred it cannot be detected and since piezoelectric sensors are placed inside the road when the roadway is repaved, or if a pothole appears the sensor would need to be replaced.

**3.3 PROPOSED SYSTEM**

In this model we have adapted many advanced methods. Here we have used PIR sensor instead of piezoelectric sensor so that even if there are road maintance it doesn’t bring damage to the sensors since we have mounted the sensors on the lamp post instead of keeping it in the road. Even if there is low light in day time due to cloudy weather or rainy atmosphere the LDR detects the intensity of the surrounding and make the street lights to glow also we have used RTC so that the street lights are completely automatic and so human efforts are reduced. We have also used GSM module so that in case of any failures of lamp it automatically generates a message and sends to service authority.

**4.4 HARDWARE REQUIREMENT**

1)LDR sensor(Resistance changes in change in light intensity)

2)RTC Real Time Clock module to feed time to board

3)Arduino UNO is used as a processor

4)PIR sensor(Detects the movement of vehicle)

5)GSM is used to send message to service authority

**3.5 SOFTWARE REQUIREMENT**

1)Arduino IDE for programming the board.

**CHAPTER 4**

**SYSTEM ARCHITECTURE**

**CHAPTER 4**

**SYSTEM ARCHITECTURE**

**4.1 SYSTEM ARCHITECTURE DIAGRAM**

Power Supply

PIR Sensor

LED array

Arduino

U

GSM

Mobile network

LDR Sensor

RTC Module

**4.2 ALGORITHM**

The following is the algorithm illustrating the automation of street lights,

if((hour>=18) && (hour<22))

Lights glow at full intensity

if((hour>=22) && (hour<06))

Lights glow at full intensity when vehicle is detected

if((hour>=06) && (hour<18))

Lights get ON only if surrounding is dark

**4.3 PSEUDOCODE**

**Step 1:** The time running in the RTC is checked.

**Step 2:** If the time is between 18 and 22, it is peak hour and so the light glows at full intensity.

**Step 3**: If the time is between 22 and 06, the lights are made to glow at full intensity if any vehicle is detected, else the lights are made to glow at 20% intensity.

**Step 4:** If the time is between 06 and 18, then the lights made to glow only if the atmospheric brightness is dark because of dark cloudy weather.

**Step 5:** And if the lights does not get ON between 18 and 22 , then an alert message is send to Service Authority using an GSM module.

**CHAPTER 5**

**CONCLUSION**

**CHAPTER 5**

**CONCLUSION**

The implementation of this method is quite easy and many advanced methods were brought . Since PIR sensors have been used instead of piezo electric sensor even if there is maintenance of road ,it does not bring damage to the sensors. This method is cheaper too. Since the street lights work automatically it reduces human effort. In case of any failure of the lights a message which include the number of the particular street lamp will be send to the service authority.

**FUTURE ENHANCEMENT**

Adaptive, interoperable lighting solutions are needed to bring savings to the next level, facilitated by connecting LED bulbs with a central management system (CMS) over the internet. These networked streetlighting systems allow operators to monitor and regulate light levels in unprecedented ways, resulting in increased energy savings and lower operational costs. The 50% energy savings that are realised by switching to LEDs increase to 80% when connectivity and a central management system (CMS) are added. With a proven business case, a number of vendors are entering the market and many of the world’s largest cities including London, New York, Hong Kong and Sydney are already implementing these connected street lighting systems. While the energy and cost saving benefits are driving adoption, cities are increasingly seeing infrastructure. With an even and widespread distribution across urban areas, readily available power and integrated connectivity, smart street lighting is being used to form the technology foundation of a city.

**APPENDICES**

**APPENDIX A**

**CODING**

**A1 CODING**

/\* Induino R3 User Guide - Program 11.0 - Interfacing with the Simple Labs DS1307 RTC Module \*/

#include <Wire.h> // I2C Library

#include<SoftwareSerial.h>

#define myrtc 0x68 // I2C Address of DS1307

char \*dow[] = {

" ", "MON", "TUE", "WED", "THU", "FRI", "SAT", "SUN"

}; // An Array to store the DAY text to match with the DAY parameter of the RTC

char \*mode[] = {

"HR", "AM", "PM"

}; // An Array to store the time mode

int dd, mm, yy, day, hh, mins, ss, mde; // Variables to store the retrieved time value

unsigned long interval1 = 15000;

unsigned long previousMillis1 = 0;

unsigned long currentMillis1;

unsigned long interval2 = 15000;

unsigned long previousMillis2 = 0;

unsigned long currentMillis2;

unsigned long interval3 = 15000;

unsigned long previousMillis3 = 0;

unsigned long currentMillis3;

unsigned long interval4 = 15000;

unsigned long previousMillis4 = 0;

unsigned long currentMillis4;

SoftwareSerial gsm(12,13);

void setup()

{

Serial.begin(9600); // Initialise Serial Communication

Wire.begin(); // Initialise Wire Communication - Join the I2C Bus

delay(500);

set\_time(3, 4, 18, 19, 20, 11, 50, 0); // Call the set\_time function to set the intial time.

pinMode(A0, INPUT); //ldr

pinMode(A1, INPUT);

pinMode(A2, INPUT);

pinMode(A3, INPUT);

pinMode(2, INPUT); //pir

pinMode(3, INPUT);

pinMode(4, INPUT);

pinMode(5, INPUT);

gsm.begin(9600);

pinMode(8, OUTPUT); //led

pinMode(9, OUTPUT);

pinMode(10, OUTPUT);

pinMode(11, OUTPUT);

pinMode(12, OUTPUT);

pinMode(13,OUTPUT);

pinMode(6,OUTPUT);

}

void loop()

{

Serial.println("");

get\_time();

Serial.print(dd);

Serial.print("/");

Serial.print(mm);

Serial.print("/");

Serial.print(yy);

Serial.print(" ");

Serial.print(dow[day]);

Serial.print(" ");

Serial.print(hh);

Serial.print(":");

Serial.print(mins);

Serial.print(":");

Serial.print(ss);

Serial.print(" ");

Serial.print(mode[mde]);

Serial.println();

delay(1000);

if ((hh >= 22) || (hh < 6)) //lamp post 1

{

int w = digitalRead(2);

Serial.println(w);

if (w == 1)

{

Serial.println("vehicle detected");

analogWrite(6, 255);

previousMillis1 = millis();

}

currentMillis1 = millis();

if ((unsigned long)(currentMillis1 - previousMillis1) >= interval1)

{

analogWrite(6,20);

previousMillis1 = millis();

}}

if ((hh >= 6) && (hh < 18))

{

float i;

i=analogRead(A0);

Serial.println(i);

if(i<=50)

{

analogWrite(6,150);

}

else

{

analogWrite(6,0);

}

}

if ((hh >= 18) && (hh < 22))

{

analogWrite(6, 255);

}

if ((hh >= 22) || (hh < 6)) //lamp post 2

{

int x = digitalRead(3);

Serial.println(x);

if (x == 1)

{

Serial.print("vehile detected");

analogWrite(9, 255);

previousMillis2 = millis();

}

currentMillis2 = millis();

if ((unsigned long)(currentMillis2 - previousMillis2) >= interval2)

{

analogWrite(9, 20);

previousMillis2 = millis();

}

}

if ((hh >= 6) && (hh < 18))

{

float j;

j=analogRead(A1);

Serial.println(j);

if(j<=50)

{

analogWrite(9,150);

}

else

{

analogWrite(9,0);

}

}

if ((hh >= 18) && (hh < 22))

{

pinMode(9,OUTPUT);

digitalWrite(9, 1);

delay(1000);

pinMode(7,INPUT);

int x1 = digitalRead(7);

Serial.println(x1);

if (x1 == 0)

{

gsm.println("ATD+919751472255;");//add target mobile number

delay(15000);

gsm.println("ATH");

delay(1000);

gsm.println("AT+CMGF=1");

delay(1000);

gsm.println("AT+CMGS=\"+919751472255\"");

//The text of the message to be sent.

delay(1000);

gsm.print("HELLO WORLD");

delay(1000);

gsm.write(0x1A);

delay(1000);

gsm.println("");

delay(15000);

}

}

if ((hh >= 22) || (hh < 6)) //lamp post 3

{

int y = digitalRead(4);

Serial.println(y);

if (y == 1)

{

Serial.println("vehicle detected");

analogWrite(10, 255);

previousMillis3 = millis();

}

currentMillis3 = millis();

if ((unsigned long)(currentMillis3 - previousMillis3) >= interval3)

{

analogWrite(10, 20);

previousMillis3 = millis();

}

}

if ((hh >= 6) && (hh < 18))

{

float k;

k=analogRead(A2);

Serial.println(k);

if(k<=50)

{

analogWrite(10,150);

}

else

{

analogWrite(10,0);

}

}

if ((hh >= 18) && (hh < 22))

{

pinMode(11,OUTPUT);

digitalWrite(10, 1);

delay(1000);

pinMode(12,INPUT);

int y1 = digitalRead(12);

Serial.println(y1);

if (y1 == 1)

{

gsm.println("ATD+918667814527;");//add target mobile number

delay(15000);

gsm.println("ATH");

delay(1000);

gsm.println("AT+CMGF=1");

delay(1000);

gsm.println("AT+CMGS=\"+918667814527\"");

//The text of the message to be sent.

delay(1000);

gsm.print("HELLO WORLD");

delay(1000);

gsm.write(0x1A);

delay(1000);

gsm.println("");

delay(15000);

}

}

if ((hh >= 22) || (hh < 6)) //lamp post 4

{

int z = digitalRead(5);

Serial.println(z);

if (z == 1)

{

Serial.println("vehicle detected");

analogWrite(11, 255);

previousMillis4 = millis();

}

currentMillis4 = millis();

if ((unsigned long)(currentMillis4 - previousMillis4) >= interval4)

{

analogWrite(11, 20);

previousMillis4 = millis();

}

}

if ((hh >= 6) && (hh < 18))

{

float l;

l=analogRead(A3);

Serial.println(l);

if(l<=50)

{

analogWrite(11,150);

}

else

{

analogWrite(11,0);

}

}

}

// The set\_time function takes parameters in the order of date, month, year, day of week, hours, minutes, seconds & mode

// the mode can have 3 possible values (0=>24HR, 1=> AM, 2 => PM)

void set\_time(int sdd, int smm, int syy, int sday, int shr, int smin, int ssec, int smode)

{

Wire.beginTransmission(myrtc); // Initialise transmission to the myrtc I2C address

Wire.write(0x00); // Write the value of the register to start with, 0 in this case represented in BCD format

Wire.write(dec\_to\_bcd(ssec)); // convert the seconds value from decimal to bcd and write it to the seconds register

// after the write operation the register pointer will be at the next register, so we do not have to set the value of the register again

Wire.write(dec\_to\_bcd(smin)); // convert the minutes value from decimal to bcd and write it to the minutes register

if (smode == 0) // Check if the mode is 24hrs mode

{

Wire.write(dec\_to\_bcd(shr)); // if 24 hours mode is on then convert the hours value from decimal to bcd and write it to the hours register

}

else // if the mode is 12 hr mode

{

// If 12 hour mode is selected then the 12 Hour mode bit (the 6th bit) has to be set to 1

// convert the hour value to bcd first and then adding 64(2^6) to the converted hrs value will set the 6th bit HIGH

shr = dec\_to\_bcd(shr) + 64;

if (smode == 1) // check if it is AM

Wire.write(shr); // if it is AM we can directly write the value of the above modified hours values to the hours register

if (smode == 2) // check if it is PM

Wire.write(shr + 32); // If it is PM, then adding 32 (2^5) sets the 5th bit (the PM indication bit) HIGH, the calculated value is written to the hours register

}

Wire.write(dec\_to\_bcd(sday)); // convert the day value from decimal to bcd and write it to the day register

Wire.write(dec\_to\_bcd(sdd)); // convert the date value from decimal to bcd and write it to the date register

Wire.write(dec\_to\_bcd(smm)); // convert the month value from decimal to bcd and write it to the month register

Wire.write(dec\_to\_bcd(syy));// convert the year value from decimal to bcd and write it to the year register

Wire.endTransmission(); // end the transmission with the I2C device

}

// the get\_time() function will retrieve the current time from the RTC and store it in the Global Variables declared

void get\_time()

{

Wire.beginTransmission(myrtc); // Initialise transmission to the myrtc I2C address

Wire.write(0x00); // Write the value of the register to start with, 0 in this case represented in BCD format

Wire.endTransmission(); // end the transmission with the I2C device

Wire.requestFrom(myrtc, 7); // Now ask the I2C device for 7 Bytes of Data // This corresponds to the values of the 7 registers starting with the 0th register

ss = bcd\_to\_dec(Wire.read()); // The first read will retrieve the value from the register address 0x00 or the seconds register, this is in the BCD format, convert this back to decimal

mins = bcd\_to\_dec(Wire.read());// The second read will retrieve the value from the register address 0x01 or the minutes register, this is in the BCD format, convert this back to decimal

hh = Wire.read();// The third read will retrieve the value from the hours register, this value needs to be processed for the 24/12 hr mode

// Check of if the BCD hours value retrieved is greater than 35 (this indicates that the hours is in 12 hour mode

// 35 is the maximum BCD value possible in the 24hr mode

if (hh > 35)

{

hh = hh - 64; // in the 12 Hours Mode the 12 hour mode bit (6th bit) is set to high, so we need to subtract 2^6 from our hours value

if (hh > 32) // Now check if the hour value is greater than 32 (2^5 = 32) (this indicates that PM bit (5th bit) is high)

{

mde = 2; // Set the mde variable to indicate PM

hh = hh - 32; // subtract 32 from the hours value

}

else // if the hour value is less than 32 it means that its in the AM mode

{

mde = 1; // Set the mde variable to indicate AM

}

}

else // if the 12 hour mode bit was not set, then the hour is in the 24 hour mode

{

mde = 0; // Set the mde variable to indicate 24 Hours

}

hh = bcd\_to\_dec(hh); // Convert the final hour value from BCD to decimal and store it back into the same variable

day = bcd\_to\_dec(Wire.read());// The fourth read will retrieve the value from the register address 0x03 or the day register, this is in the BCD format, convert this back to decimal

dd = bcd\_to\_dec(Wire.read());// The fifthread will retrieve the value from the register address 0x04 or the date register, this is in the BCD format, convert this back to decimal

mm = bcd\_to\_dec(Wire.read());// The sixth read will retrieve the value from the register address 0x05 or the month register, this is in the BCD format, convert this back to decimal

yy = bcd\_to\_dec(Wire.read());// The seventh read will retrieve the value from the register address 0x06 or the year register, this is in the BCD format, convert this back to decimal

}

// The dec\_to\_bcd() function converts a given decimal number to BCD format

int dec\_to\_bcd(int dec)

{

return dec / 10 \* 16 + (dec % 10); // convert and return the number from decimal to bcd format

}

// The dec\_to\_bcd() function converts a given BCD number to decimal format

int bcd\_to\_dec(int bcd)

{

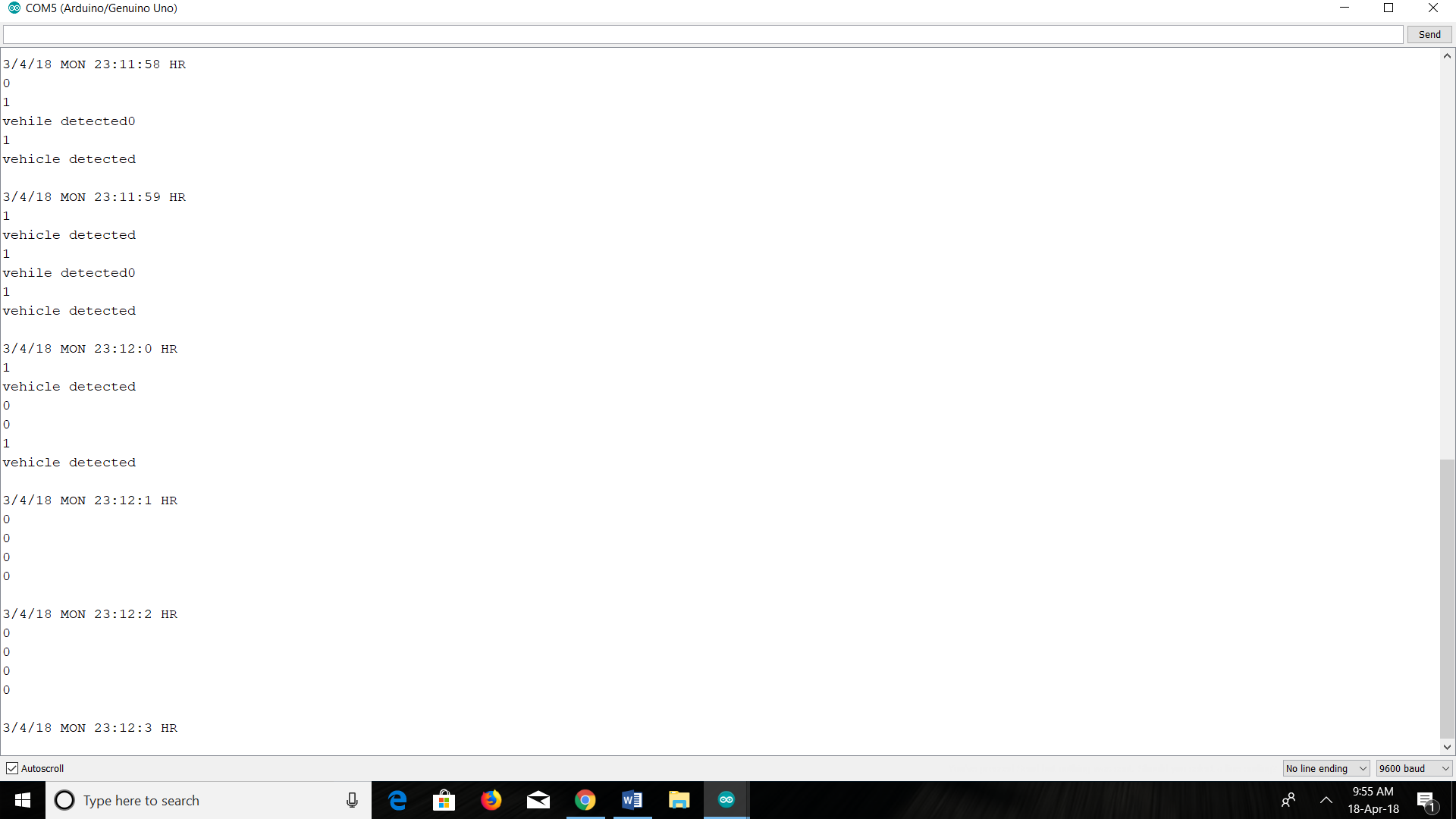
return bcd / 16 \* 10 + (bcd % 16); // convert and return the number from bcd to decimal format

}

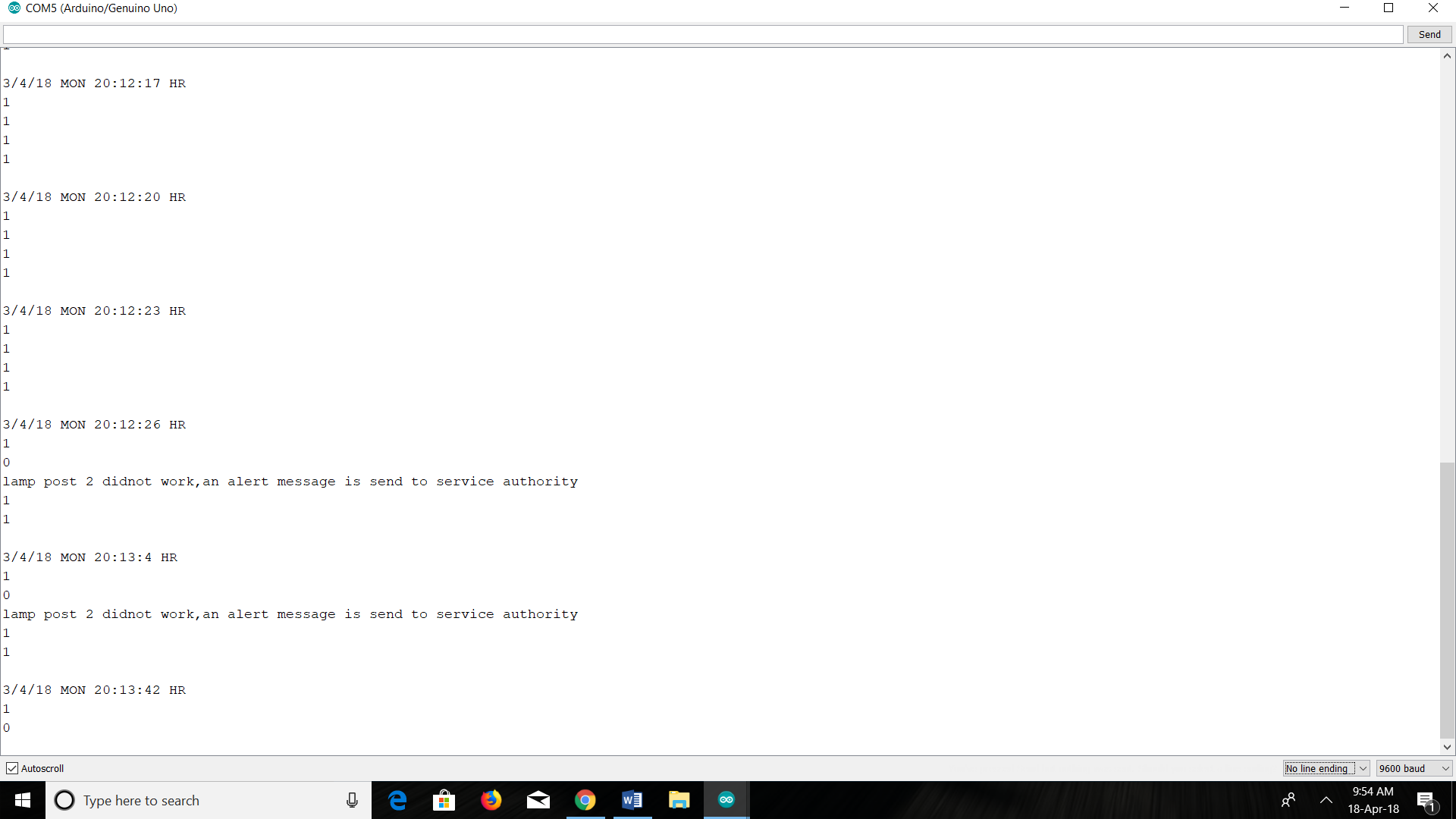
**APPENDIX B**

**SCREENSHOTS**

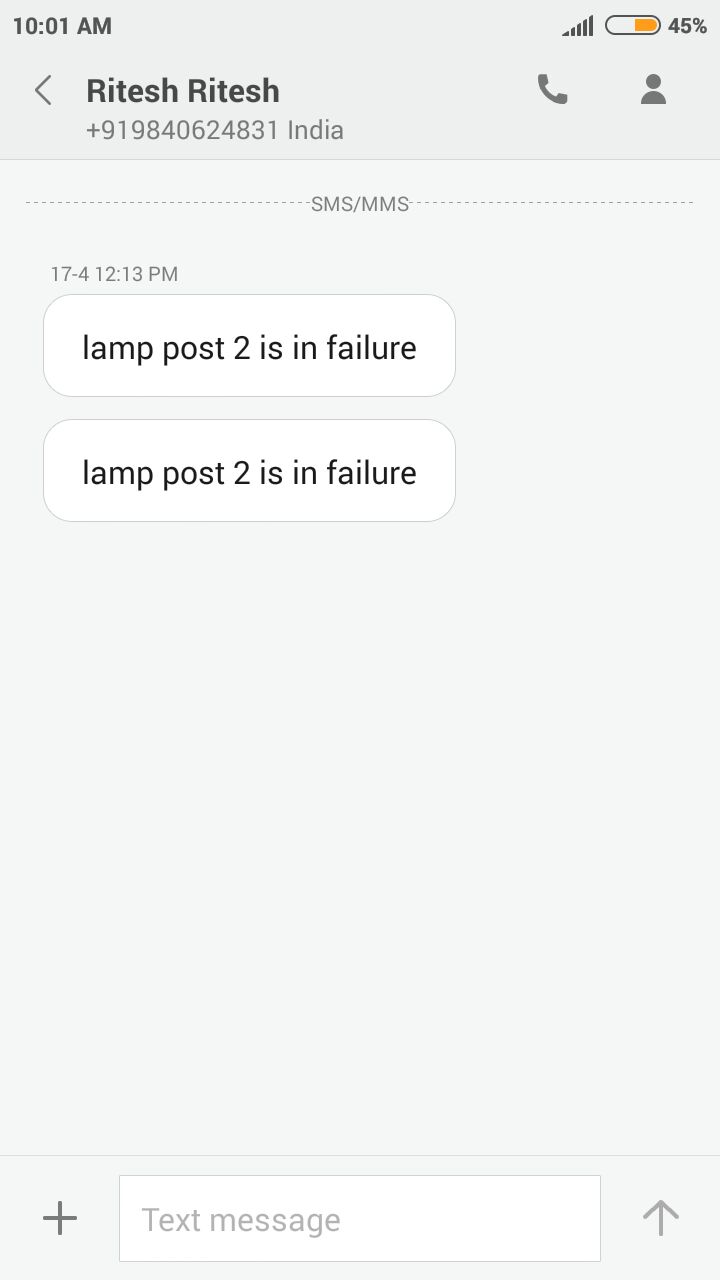
**A2 SCREENSHOT**



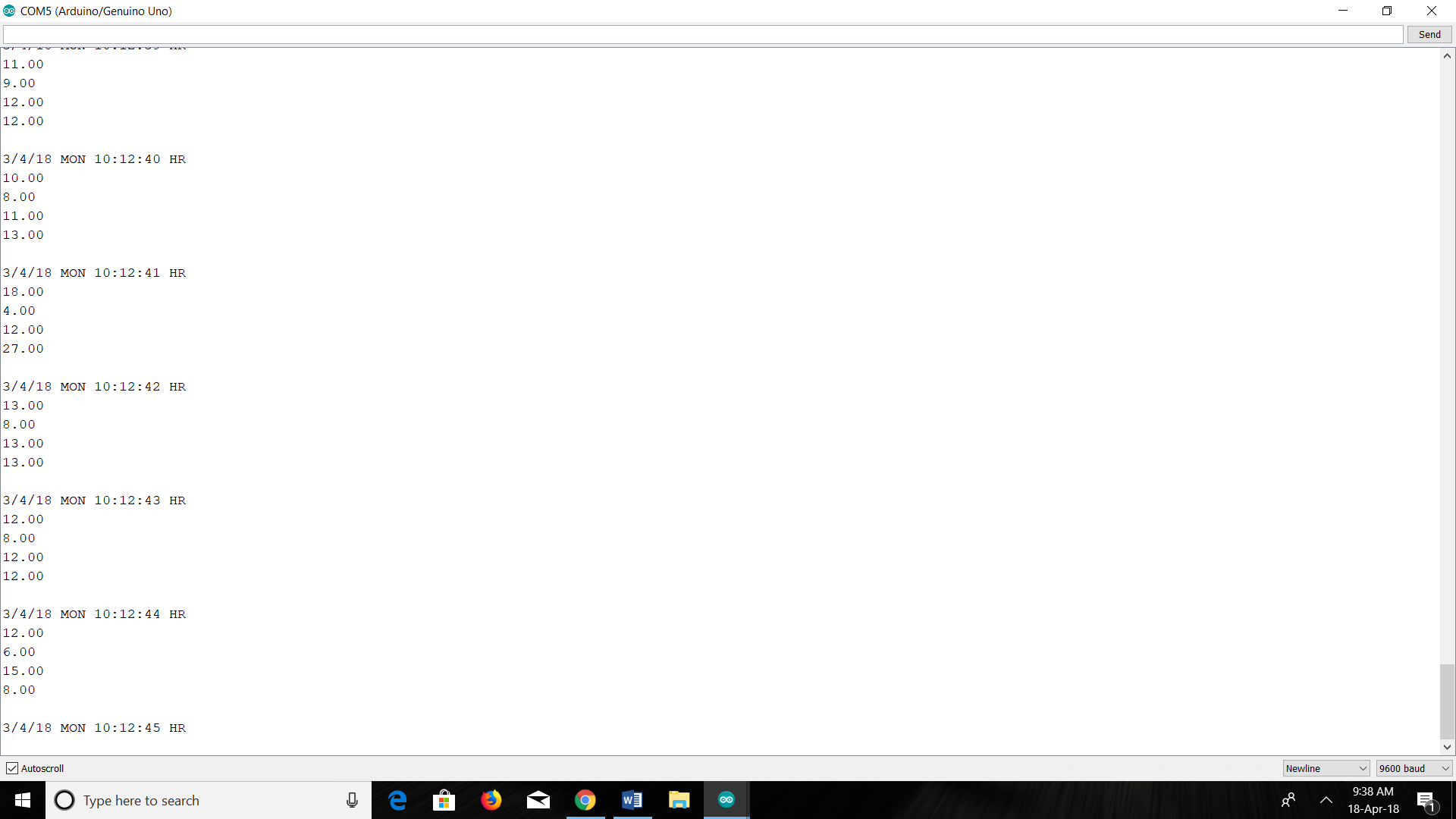
**A3 Lamp status between 22 and 06**



**A4 Lamp status between 18 and 22**



**A5 Screenshot of message received by Authority**



**A6 Lamp status between 06 and 18**

**REFERENCES**

[1] [Noriaki Yoshiura](https://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Noriaki%20Yoshiura.QT.&newsearch=true)[Yusaku Fujii](https://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Yusaku%20Fujii.QT.&newsearch=true)[Naoya Ohta](https://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Naoya%20Ohta.QT.&newsearch=true), University Technology MARA, Shah Alam, Selangor, Malaysia; “Automatic Street Lighting System For Energy Efficiency Based on Low cost microcontroller”, ISSN:1437-804x online, January 2005,DOI:10.5013/IJSSST, pp.43,44

[2] Chetna Badgaiyan , Palak Sehgal, Dept. of Control Engg, college of Electronics & Technology, Baniwalid-Libya, “Automatic Street Light Control System Using Microcontroller”, ISBN: 978-960-474-399-1,PP. 92-95

[3]. Samir A. Elsagheer Mohamed , Street light glow on detecting vehicle movement using sensor

[4]. Prof. K.Y.Rajput , Gargeyee Khatav , Monica Pujari, Priyanka Yadav, Survey on Street Lighting System Based On Vehicle Movements

[5]. Srikanth M, Sudhakar K N,ZigBee Based Remote Control Automatic Street Light System

[6]. Abdul Latif Saleem,Raja Sagar R,Sachin Datta NS,Sachin H,Usha M S, Design and implementation of traffic flow based street light control system with effective utilization of solar energy, International journal of Science Engineering and Advance Technology, IJSEAT, Vol 3, Issue 9, September -2015

[7]. Jayesh Patel,Sumit Tayade,Manjusha Pattadkal,Neelambari Nawale,Harsha Bhute, Analysis of Solar energy based street light with auto tracking system, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol 2, Issue 7, July 2013

[8]. R Abinaya,Kuluvan,Hariharan,“Street Light Monitoring – a Practical Solution magazine” November/December 2002

[9]. P Kavita A. Bajaj , Tushar S. Mote, Design and Implementation of CPLD based Solar Power Saving System for Street Lights and Automatic Traffic Controller, International Journal of Scientific and Research Publications.

[10]. Amul Shravan kumar Jala, Electric Energy Savings and Light Guides, Energy& Environment, 3 rd IASME/WSEAS International Conference on, Cambridge, UK, February 2008, pp.470-474.

[11]. Gouthami. C , Santosh. C ,A. Pavan Kumar, Karthik. A, Ramya.K.R,” Intelligent Lighting System Using Wireless Sensor Networks”, International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC) Vol.1, No.4, December 2010, pp 17-27

[12]. Abdul Latif Saleem, Raja Sagar R, Sachin Datta N S,Sachin H S,Usha M S,” Smart Personal Sensor Network Control for Energy Saving in DC Grid Powered LED Lighting System”, IEEE Trans. Smart Grid

[13] Sindhu.A.M, Jerin George, Sumit Roy, Chandra J “Automatic Street Light Control System Using Microcontroller” Department of Control Engineering College of Electronic Technology/ Baniwalid Baniwalid- Libya LIBYA, Mathematical Methods and Optimization Techniques in engineering

[14] Mr. A.  Syed Noor-ul-Hassan Bukhari,Syed Zulfiqar Haider Bukhari,Ahmad Faheem Alam,Javed Iqba, Design and Implementation of CPLD based Solar Power Saving System for Street Lights and Automatic Traffic Controller, International Journal of Scientific and Research Publications, Vol. 2, Issue11, November 2012.

[15] J. Mohelnikova, Electric Energy Savings and Light Guides, Energy& Environment, 3rd IASME/WSEAS International Conference on, Cambridge, UK, February 2008, pp.470-474.

[16] M. A. Wazed, N. Nafis, M. T. Islam and A. S. M. Sayem, Design and Fabrication of Automatic Street Light Control System, Engineering e-Transaction, Vol. 5, No. 1, June 2010, pp 27-34.

[17] R. Priyasree, R. Kauser, E. Vinitha and N. Gangatharan, Automatic Street Light Intensity Control and Road Safety Module Using Embedded System, International Conference on Computing and Control Engineering, April 2012.

[18] K. S. Sudhakar, A. A. Anil, K. C. Ashok and S. S. Bhaskar, Automatic Street Light Control System, International Journal of Emerging Technology and Advanced Engineering, Vol. 3, May 2013, PP. 188-189.

[19] K.Y. Rajput, G. Khatav, M. Pujari, P. Yadav, Intelligent Street Lighting System Using Gsm, International Journal of Engineering Science Invention, Vol2, Issue 3, March 2013, PP. 60- 69.

[20] M. Popa, C. Cepişcă, Energy Consumption Saving Solutions Based on Intelligent Street Lighting Control System. U.P.B. Sci. Bull., Vol. 73, April 2011, PP. 297-308.

[21] R. Mohamaddoust, A. T. Haghighat, M. J. M. Sharif and N. Capanni, A Novel Design of an Automatic Lighting Control System for a Wireless Sensor Network with Increased Sensor Lifetime and Reduced Sensor Numbers, Sensors, Vol. 11, PP. 8933-8952.

[22] L. Jasio, T. Wilmshurst, D. Ibrahim, J. Morton, M. Bates, J. Smith D. Smith and C. Hellebuyck, PIC Microcontrollers: know it all, Publishing Elsevier Science, 2008.

[23] W. Bolton. Instrumentation and Control Systems, Elsevier Science & Technology Books, August 2004.

[24] PIC16F87XA Data Sheet. 2003 Microchip Technology. Mathematical Methods and Optimization Techniques in Engineering ISBN: 978-960-474-339-1 96

1. [↑](#footnote-ref-1)