International Journal of Engineering Sciences & Research Technology

(A Peer Reviewed Online Journal)
Impact Factor: 5.164

IJESRT

Website: www.ijesrt.com Mail: editor@ijesrt.com

Correspondence Address: 116, Sukhdev Nagar Ext-1, Airport Road, Indore – 452005, Madhya Pradesh, INDIA

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

AUTOMATIC STREET LIGHT CONTROL AND MONITORING SYSTEM

Samaditya Jatar [*1], Samarth Bhole [2]

[*1 ,2] Computer Science Engineering Department ,Shri Vaishnav Institute of Information Technology, Indore(M.P.),India

DOI: 10.29121/ijesrt.v11.i2.2022.6

ABSTRACT

Street lighting is one of the important parts of a city's infrastructure where the main function is to illuminate the city's streets during dark hours of the day. Previously, the number of streets in the town and city were less. So, the street lamps were relatively simple but with the development of urbanization, the number of streets increased rapidly with high traffic density. There are several factors needed to be considered in order to design a good street lighting system such as night-time safety for community members and road users, provide public lighting at low cost and the reduction of light pollution. Initially, the street lamps were controlled manually where a control switch is set in each of the street lamps. It is called first generation of the original street light. After that, another method that was been used was optical control method. This method uses high pressure sodium lamp in their system. It can be seen that this method is widely used in the country nowadays. This method operates by setting up an optical control circuit. It lights up automatically at dusk and turns off automatically after dawn in the morning. The system utilizes the latest technology for the sources of light as LED Lamps instead of generally used street lamps such as High Pressure Sodium Lamps, etc. The LED technology is preferred as it offers several advantages over other traditional technologies. The application is designed in such a way that we place light sensors in all street light circuits, which is responsible to switch on and off automatically. Once the lights are switched on current sensors placed at every street light circuits are responsible to report problem status to the centralized system with help of GSM module attached with the circuit The system described can effectively save energy by reducing the power consumption as per requirement. Since this is a sensor based system, so it is self-controlled and automated system. The system is also flexible for any modification or further expansion such as interfacing of new sensors, connecting surveillance camera for the security purpose, etc. This project of Smart Street Light System is a cost effective, practical, eco-friendly and the safest way to save energy.

KEYWORDS: Arduino Uno, Django Framework, SQLite, GSM

1. INTRODUCTION

Automatic Street Light Systems are developed to control and reduce the energy consumption of a town's public lighting system depending on the vehicle density. Automation, power consumption and cost effectiveness are the important considerations in the present field of electronics and electrical related technologies. Street lighting is one of the important part of a city's infrastructure where the main function is to illuminate the city's streets during dark hours of the day. Previously, the number of streets in the town and city were less. So, the street lamps were relatively simple but with the development of urbanization, the number of streets increased rapidly with high traffic density. There are several factors needed to be considered in order to design a good street lighting system such as night-time safety for community members and road users, provide public lighting at low cost and the reduction of light pollution.

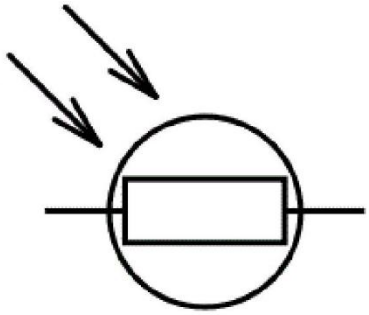
2. Tools and Technology

2.1 Hardware Tools

2.1.1 LDR Sensor

This A **Light Dependent Resistor** (LDR) or a photo resistor is a device whose resistivity is a function of the incident electromagnetic radiation. Hence, they are light sensitive devices. They are also called as photo conductors, photo conductive cells or simply photocells. They are made up of semiconductor materials having high resistance. There are many different symbols used to indicate a **LDR**, one of the most commonly used symbol is shown in the figure below. The arrow indicates light falling on it.

Figure:



2.1.2 Laser

A laser is a device that emits light through a process of optical amplification based on the stimulated emission of electromagnetic radiation. The term laser originated as an acronym for light amplification by stimulated emission of radiation. Lasers are distinguished from other light sources by their coherence. Laser beams can be focused to very tiny spots, achieving a very high irradiance, or they can have very low divergence in order to concentrate their power at a great distance enabling applications such as laser cutting and lithography.

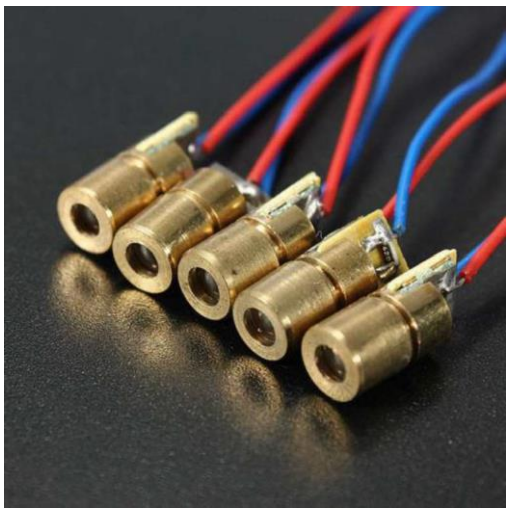


Figure No. 2.2 Laser Light

Working life: more than 2000 hours	Light power: <5mW
Spot mode: continuous output	Supply voltage: 5V DC
Laser wavelength: 650nm (red)	Operating current: <40mA
Copper head diameter: 6mm	Operating temperature: -36 ~ 65
Power lead length: 120mm	Shell material: brass

Table No. 2.1 Specification of Laser Light

2.1.3 Arduino Uno Board

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board.[2] To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

On board

There are many varieties of Arduino boards that can be used for different
purposes. Some boards look a bit different from the one below, but most Arduinos have the majority of these components in common:

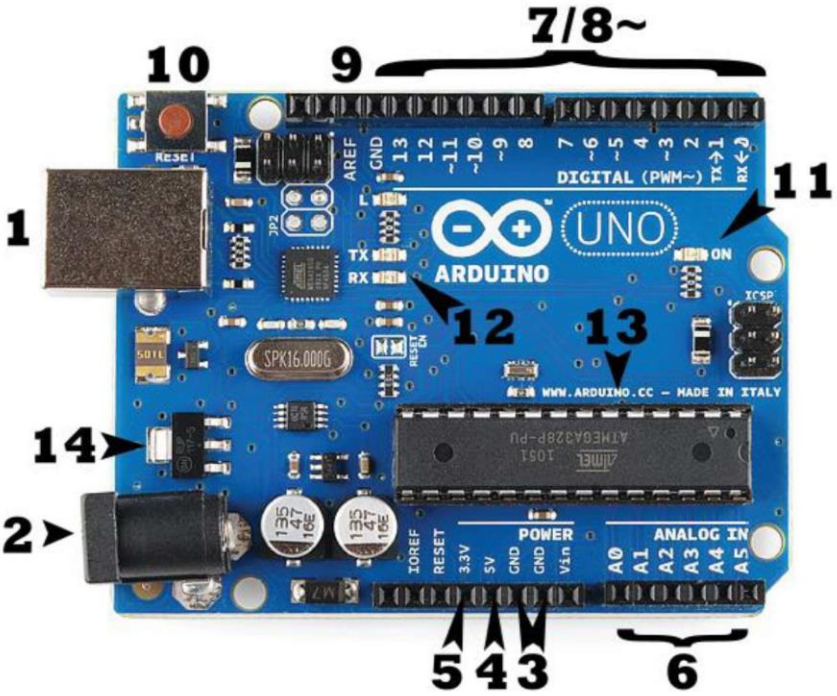


Figure No. 2.3 Arduino Board

Power (USB / Barrel Jack)

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply that is terminated in a barrel jack. In the picture above the USB connection is labeled (1) and the barrel jack is labeled (2).The USB connection is also how you will load code onto your Arduino board.

Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjunction with a breadboard and some wire. They usually have black plastic headers that allow you to plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

*GND (3):*Short for 'Ground' which can be used to ground your circuit. There are several GND pins on the Arduino, any of which can be used.

5V (4) & 3.3V (5): As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.

Analog (6): The area of pins under the 'Analog-in' labelled(A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.

Digital (7): Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).

PWM (8): You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins ,but can also be used for something called Pulse-Width Modulation (PWM).

AREF (9): Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Reset Button

Just like the original Nintendo, the Arduino has a reset button (10). Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very helpful if your code doesn't repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn't usually fix any problems.

Power LED Indicator

Just beneath and to the right of the word "UNO" on your circuit board, there's a tiny LED next to the word "ON" (11). This LED should light up whenever you plug your Arduino into a power source. If this light doesn't turn on there's a good chance something is wrong. Time to re-check your circuit!

TX RX LEDs

TX is short for transmit; RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the Arduino UNO where TX and RX appear once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs (12). These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data.

Main IC

The black thing with all the metal legs is an IC, or Integrated Circuit (13). Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type but is usually from the AT mega line of ICs from ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software.

Voltage Regulator

The voltage regulator (14) is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it is for .The voltage regulator does exactly what it says it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits anything greater than 20 volts.

2.1.4 ULTRASONIC SENSOR

An Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object. Since it is known that sound travels through air at about 344 m/s (1129 ft/s), you can take the time for the sound wave to return and multiply it by 344 meters (or 1129 feet) to find the total round-trip distance of the sound wave.

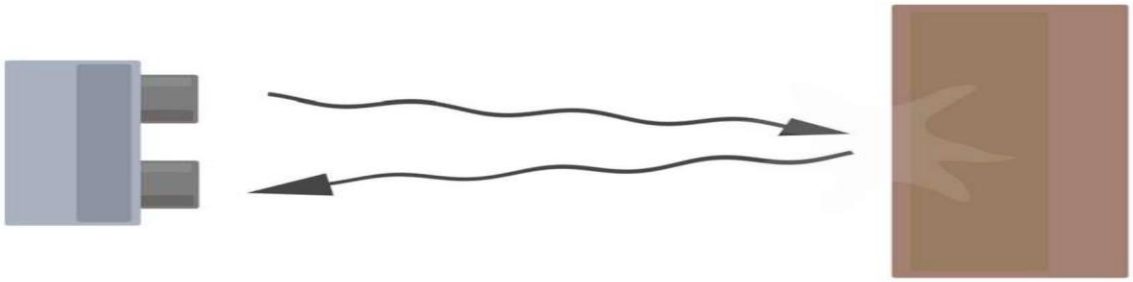


Figure No. 2.4 The basic ultrasonic sensor operation

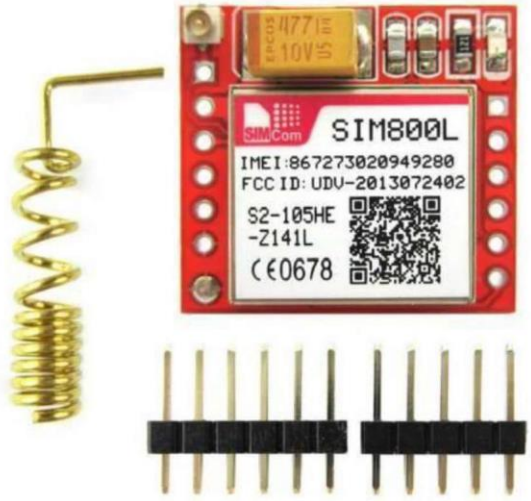
Round-trip means that the sound wave traveled 2 times the distance to the object before it was detected by the sensor; it includes the 'trip' from the sonar sensor to the object AND the'trip' from the object to the Ultrasonic sensor (after the sound wave bounced off the object).To find the distance to the object, simply divide the round-trip distance in half.

2.1.5 GSM Module

GSM/GPRS module is used to establish communication between a computer and a GSM-GPRS system. Global System for Mobile communication (GSM) is an architecture used for mobile communication in most of the countries. Global Packet Radio Service (GPRS) is an extension of GSM that enables higher data transmission rate. GSM/GPRS module consists of a GSM/GPRS modem assembled with power supply circuit and communication interfaces (like RS-232, USB, etc.) for computer. 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 like 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.

Figure No. 2.5 SIM800L module

Specifications for SIM800L:

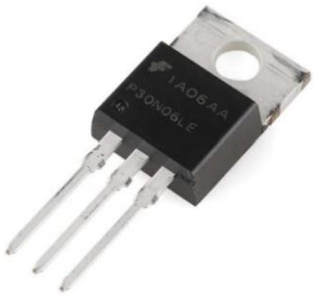
- Low power consumption: 1.5mA (sleep mode)
- Operation temperature: -40°C to +80 °C
- Input voltage :5V DC
- Specifications for GPRS Data:
- max. 85.6 kbps (downlink) max. 42.8 kbps (uplink)

Software features:

- Embedded TCP/UDP protocol
- FTP/HTTP
- Embedded AT

2.1.6 IRFZ MOSFET

MOSFET is a special type of field-effect transistor (FET) that works by electronically varying the width of a channel along which charge carriers (electron s or hole s) flow. The wider the channel, the better the device conducts. The charge carriers enter the channel at the source and exit via the drain. The width of the channel is controlled by the voltage on an electrode called the gate, which is located physically between the source and the



drain and is insulated from the channel by an extremely thin layer of metal oxide. There are two ways in which a MOSFET can function. The first is known as depletion mode. When there is no voltage on the gate, the channel exhibits its maximum conductance. As the voltage on the gate increases (either positively or negatively, depending on whether the channel is made of P-type or N- type semiconductor material), the channel conductivity decreases. The second way in which a MOSFET can operate is called enhancement mode. When there is no voltage on the gate, there is in effect no channel, and the device does not conduct. A channel is produced by the application of a voltage to the gate. The greater the gate voltage, the better the device conducts.

Figure No 2.6 IRFZ MOSFET

2*.1.7 ATMEGA328*

The ATmega328 is a single-chip microcontroller created by Atmel in the megaAVR family. ATmega328 is commonly used in many projects and autonomous systems where a simple, low-powered, low-cost micro-controller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino Uno and Arduino Nano models.

Parameter	Value
CPU Type	8-bit AVR
Performance	20 MIPS at 20 MHz
Flash Memory	32 Kb
SRAM	2 Kb
EEPROM	1 Kb
PIN COUNT	28-pin PDIP
Maximum Operating Frequency	20 Mhz
Maximum I/O pins	26
External Interrupts	2
USB Interface	NO
Temperature Range(C)	-40 to 85
Operating Voltage Range(V)	1.8 to 5.5

Table No. 2.2 Specification of ATMEGA328

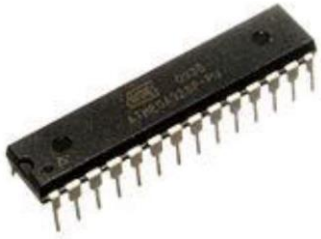


Figure No. 2.7 ATmega328P in a 28-pin dial inline package (DIP)

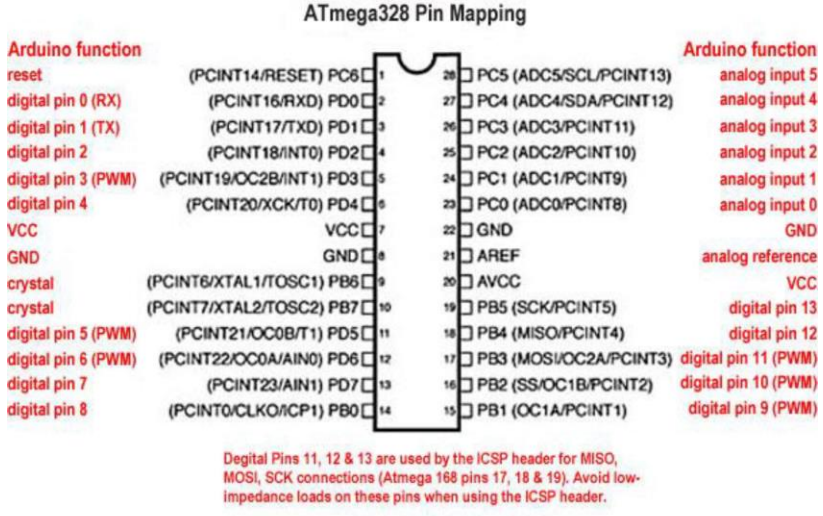


Figure No. 2.7 ATmega328P in a 28-pin dial inline package (DIP

2.2 Software Tools

2.*2.1 Arduino IDE*

The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It is designed to introduce programming to artists and other newcomers unfamiliar with softwaredevelopment. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and provides simple one-click mechanism to compileand load programs to an Arduino board. A program written with the IDE for Arduino is called a "sketch".The Arduino IDE supports the languages C and C++ using special rules to organize code. The Arduino IDE supplies a software library called Wiring from the Wiringproject, which provides many common input and output procedures. A typical Arduino C/C++ sketch consist of two functions that are compiled and linked with a programstub main () into an executable cyclic executive program:

- **loop():** a function called repeatedly until the board powers off.
- **setup():** a function that runs once at the start of a program and that can initialize settings.

After compiling and linking with the GNU tool chain, also included with the IDE distribution, the Arduino IDE employs the program coder to convert the executablecode into a text file in hexadecimal coding that is loaded into the Arduino board by a loaderprogram in the board's firmware.

Arduino programs may be written in any programming language with a compiler that produces binary machine code. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio, which can be used forprogramming Arduino.

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback whilesaving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right-hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

- **Verify**

  Checks your code for errors compiling it.

- **Upload**

  Compiles your code and uploads it to the configured board. See uploading below for details.

- **New**

  Creates a new sketch.

- **Open**

  Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

- **Save**

  Saves your sketch

- **Serial Monitor**

  Opens the serial monitor

2.2.2 EAGLE SOFTWARE

EAGLE is a scriptable electronic design automation application with schematic capture, printed circuit board layout, auto router and computer-aided manufacturing features. EAGLE contains a schematic editor, for designing circuit diagrams. Parts can be placed on many sheets and connected together through ports. The PCB layout editor allows back annotation to the schematic and auto-routing to automatically connect traces based on the connections defined in the schematic.

2.2.3 SQLITE

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file. The database file format is cross-platform - we can freely copy a database between 32-bit and 64-bit systems or between big-endian and little- endian architectures. These features make SQLite a popular choice as an Application File Format.

SQLite is very carefully tested prior to every release and has a reputation for being very reliable. SQLite is an in-process library that implements a self- contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. SQLite is the most widely deployed database in the world with more applications than we can count, including several high-profile projects.

2.2.4 Django Framework

Django is a free and open-source web framework, written in Python, whichfollows the model-view-template (MVT) architectural pattern. It is maintained by the Django Software Foundation (DSF), an independent organization established as a non-profit organization.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "plug ability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

- **Ridiculously fast.**

Django was designed to help developers take applications from concept to completion as quickly as possible.

- **Reassuringly secure.**

Django takes security seriously and helps developers avoid many common security mistakes.

- **Exceedingly scalable.**

Some of the busiest sites on the web leverage Django's ability to quickly and flexibly scale.

Django follows MVC pattern closely enough to be called an MVC framework. Django has been referred to as an MTV framework because the controller is handled by the framework itself and most of the excitement happens in models, templates and views.

2.3 Automatic Street Light Control System

2.3.1 Automatic Street Light Control system Using Microcontroller

this paper aims at designing and executing the advanced development in embedded systemsfor energy saving of street lights. Nowadays, human has become too busy, and is unable tofind time even to switch the lights wherever not necessary. This paper gives the best solutionfor electrical power wastage. Also the manual operation of the lighting system is completelyeliminated. In this paper the two sensors are used which are Light Dependent Resistor LDRsensor to indicate a day/night time and the photoelectric sensors to detect the movement onthe street. The microcontroller PIC16F877A is used as brain to control the street light system, where the programming language used for developing the software to the microcontroller is C-language.

2.3.2 Intelligent Street Lighting System Using Zigbee

street lighting systems in areas with a low frequency of passers are by online most of the night without purpose. The consequence is that a large amount of power is wasted meaninglessly. With the broad availability of flexible-lighting technology like light- emitting diode lamps and everywhere available wireless internet connection, fast reacting, reliably operating, and power-conserving street lighting systems become reality. Thepurpose of this work is to describe the Intelligent Street Lighting (ISL) system, a first approach to accomplish the demand for flexible public lighting systems.

Most of these systems developed have contributed some drawbacks. They have been considered to be outdated due to lack of communication capabilities, not allowing a system feedback. Hence, GSM communication technology is deployed so as to ensure a higher efficiency and overcome current drawbacks. GSM enabled street light is based on the wireless sensor network application that utilizes wireless communication protocol to enhance the technology of street lighting system by providing communication capabilities.GSM enabled street light are mostly battery powered or can work with current supplies hence

there is no need of laying separate underground cable connection. GSM is an attentioncommand based protocol which send and receive data serially.

3. DISCUSSION

3.1 Detailed Problem Statement

This system monitors the status of the street lamp and keeps track of power consumed by each and every lamp thus allowing us to estimate the overall consumption of lamp. With the application of the lamp illumination control on the system, the system is ableto turn ON the lights with low illumination when the surrounding condition needs the low light illumination of the lamps (e.g. rainy or cloudy day).

This system is the cost saving in terms of wiring as it can be easily installedon the already existing street light system. The GSM module will allow the streets lamps communicate to the control system via wireless connectivity. With the wiring method, the high cost of the construction and material makes the system uneconomical, moreover the reliability of the system will reduce.

3.2 Requirement Analysis

The project of a Wireless Sensor Network depends on the application wherethe network will be used, and it should be based on functional and non-functional requirements. Functional requirements for this type of project are:

3.2.1 Functional Requirement

- **Points supervision:** Node status, whether it is connected to the network or not,estimation of power consumed, and LEDs luminosity level.
- **Actuate the nodes through a remote tool:** besides the automatic actions, theoperator should be able to select a region through a remote tool to actuate.
- **Automation of information storage:** Status of lamp is sent periodically to thecentral server and power consumption of each lamp is calculate
- **Control:** switch on/off the luminosity level lamp, switch on/off a lamp post,a selected segment, a street, neighborhood, city.

To meet the functional requirements, a control and monitoring application isrequired to report system information to the user. This application will be able to monitor operating conditions, for example, notifying the user about the end of life of a lamp if it exceeds a particular limit of hours.

3.2.2 Non-Functional Requirement

- **Scalability:** when proposing an approach to sensor networks, it is important toconsider that the project should apply to small (up to 15 nodes), medium (15 to 103 nodes), or large networks (10^2 to 10^7 nodes);

- **Fault tolerance:** depending on the physical conditions, it may be necessary toadopt sensor nodes with special characteristics. In addition, due to difficult accessibility in some environments, it may be unmanageable to replace damaged nodes. Therefore, it is necessary to provide sensor nodes with adaptive capacities to deal with unstable environments. In the type of application described in this work, nodes must support high temperatures andheavy rain.
- **Delivery Guarantee:** regarding the unstable wireless communication nature, because of external interference and/or electronic noise, it is necessary to checkcontinuously for any error.
- **Lifetime:** For this project, most sensor nodes are powered by energy cables that are already used in current street lighting structures. Avoiding the use of batteries is an ecological issue, because devices maintenance involves a considerable amount of trash. Batteries would be used only in areas where there is no energy cable structure, such as rural areas
- **Latency:** This parameter reflects the time interval the network has to inform the observer about a specific phenomenon. Some applications can be sensibleto latency, therefore it is necessary to respond in a short time interval (equal orless than one second), for example, in control process supervision In relation to street lighting application, the acceptable latency should be on a time interval in the order of tens of seconds

3.3 Cost Estimation

RESULTS EXPECTED VS CURRENT SYSTEM

Energy utilization: Works on profile basis i.e. all street lights are ON from 6:30 pm to 6:30am, in other words street lights are functioning completely for 12 hrs a day.

- Assuming 1000 nodes to be working power consumed by them will be given as, Bulb used =400 W
- Number of nodes = 1000 nodes
- Number of working hrs per day = 12 hrs
- Power Consumed per day = 1000 * 12 * 0.40 = 4800
- Kwhr Power consumed in year = 4800*365 days
  =17,52,000 units
- Yearly Bill for 1000 nodes (Rs 7/kwhr) = 17,52,000 * 7 = 1,22,64,000 Rs peryear

Now replacement by LED light

- Assuming 1000 nodes to be working power consumed by them will be given as, LED Bulb used =200 W

- Number of nodes = 1000 nodes
- Number of working hrs per day = 12 hrs
- Power Consumed per day = 1000 * 12 * 0.20 = 2400 KwhrPower consumed in year = 2400*365 days
- =876,000 units
- Yearly Bill for 1000 nodes (Rs 7/kwhr) = 876000 * 7 = 61,32,000 Rs per year

Saving using led lamp over sodium = (1,22,64,000 -61,32,000)/1,22,64,000*100 =**50%**

Using led lamp with our system

- Assuming 1000 nodes to be working power consumed by them will be given as, LED used =200W
- Number of nodes = 1000 nodes
- Number of working hrs per day (high intensity)= 6 hour Number of working hrs per day (low intensity)=
- 6 hour Power Consumed per day = 1000 * 6* 0.200 =
- 1200 Kwhr
- Power Consumed per day = 1000 * 1* 0.010 = 60 Kwhr Power consumed in year = (1200+60)*365 days
- =459900 units
- Yearly Bill for 1000 nodes (Rs 7/kwhr) = 459900 * 7 = 32,19,300 Rs per year

Net saving using automatic street light = (61,32,000-32,19,300)/ 61,32,000 *100=47.58%

Net Saving using automatic street light over sodium lamp= (1,22,64,000-32,19,300)

/1,22,64,000=73.75 %

PROTOTYPE COST

For 1000 lamps

- Assuming 1000 nodes to be working power consumed by them will be given as, LED used =12W
- Number of nodes = 1000 nodes
- Number of working hrs per day (high intensity)= 6 hourNumber of working hrs per day (low
- intensity)= 6 hour Power Consumed per day =

1000 * 6* 0.012 = 12 KwhrPower Consumed per day = 1000 * 6* 0.001 = 6 Kwhr Power consumed in year = (12+6)*365 days =6570units

- Yearly Bill for 1000 nodes (Rs 7/kwhr) = 6570 * 7 = 45990 Rs per year

For 6 lamps

- Assuming 6 nodes to be working power consumed by them will be given as, LED used =12W
- Number of nodes = 6 nodes
- Number of working hrs per day (high intensity)= 6 hour Number of working hrs per day (low intensity)= 6 hour Power Consumed per day = 6 * 6* 0.012 = 0.432 KwhrPower Consumed per day = 6 * 6* 0.001 = 0.036 KwhrTotal power consumed in one day = 0.468
- Bill for 6 nodes (Rs 7/kwhr) = 0.468 * 7 = 3.276 Rs

Without using system

- Assuming 6 nodes to be working power consumed by them will be given as, LED used =12W
- Number of nodes = 6 nodes
- Number of working hrs per day (high intensity)= 6 hour Number of working hrs per day (low intensity)= 6 hour Power Consumed per day = 6 * 12* 0.012 = 0.864 KwhrTotal power consumed in one day = 0.864
- Bill for 6 nodes (Rs 7/kwhr) = 0.864 * 7 = 6.048 Rs

3.4 SAVING

(Cost without system-cost with system)/cost without system **= (6.048-3.276)/6.048 =45.8 %**

FINANCIAL ESTIMATION

Gsm module	: 600 Rs
LDR	: 30 Rs
Microcontroller	: 200 Rs
Laser light	: 36 Rs
UltraSonic	: 240 Rs
IRFZ	: 30 Rs

LED : 20 Rs

- Cost of implementing the system on a single pole = 30+36+30+20=116 Rs
- Cost of implementing on 6 poles =600+200+240+(116*6) = 1736 Rs
- Cost of implementing the system on 1000 poles =(1000/6)*1736 = 2,89,333 Rs

power ultrasonic sensor:1000

Rspower led :2000 Rs

power irfz :240 Rs

- Cost of implementing the system on a single pole = 30+36+240+20=2000 RsCost of implementing on 6 poles =600+200+1000+(2000*6) = 22800 Rs
- Cost of implementing the system on 1000 poles =(1000/6)*22800 = 38,00,000Rs

Saving in first year = (1,22,64,000 - (38,00,000+32,19,300))/ 1,22,64,000*100= 42.7%

3.5 DIAGRAMS

3.5.1 USE CASE DIAGRAM

In software and systems engineering, a **use case** is a list of actions or eventsteps, typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system, to achieve a goal. The actor can be a human or other external system.

Figure No. 3.1 Use Case Diagram

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

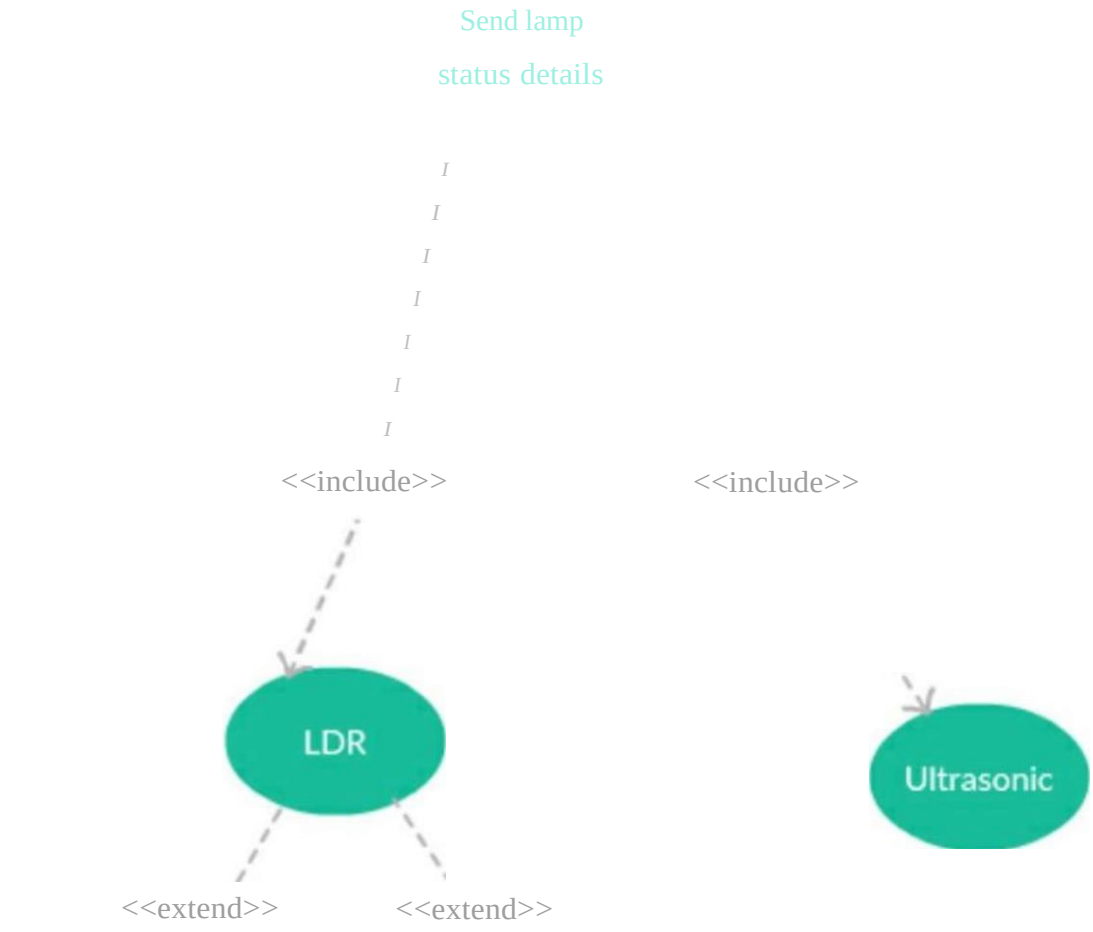


Figure No. 3.2 Detailed Use Case

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

Use Case Description for "Manage areas/street poles"

Goal	To manage the status of street lamps
Initiator	Admin
Triggering Event	Initiated by admin
Precondition	Authentication of admin is successful
Success Post Condition	Managing the poles and generate a power consumption reportFailure Not assessed by authentic user

Use case description for "Switch street light ON/OFF"

Goal	To switch the street lamps on/off
Initiator	Admin
Triggering Event	Initiated by admin
Precondition	Authentication of admin should be successful and the system Should be in remote mode
Success Post Condition	Switch all lamps on/off by the button provided for each lamp.Failure System not in remote mode

Use Case description for "Switch street light ON/OFF"

Goal	To switch the street lamps on/off
Initiator	Arduino System
Triggering Event	Vehicle is detected by the sensors
Precondition	Sensors should be set up correctly and the system should be in Arduino mode.
Success Post Condition	Switch all lamps on/off based on the sensor outputFailure System not in Arduino mode

Use case description for "View power consumption"

Goal	To generate report of power consumed by each lamp
Initiator	Admin
Triggering Event	Initiated by admin
Precondition Condition	Authentication of admin should be successful. Success Post Power consumption table of each lamp is displayedFailure Not assessed by authentic user

Use case description for "Adjust Intensity"

Goal	To adjust the intensity of each lamp
Initiator	Arduino System

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

Triggering Event	Initiated by detection of vehicle and Arduino system.
Precondition	1) Night time/Bad weather condition
	2) Vehicle is detected
Success Post Condition	Lights are dimly lit and lit on full intensity on the basis of the
	Above pre-conditions.
Failure	Sensors not working properly.

Use Case description for "Send Lamp status"

Goal	To send status of lamps to the server
Initiator	Arduino System
Triggering Event	Initiated by Arduino system.
Precondition	Change in the intensity of light lamps.
Success Post Condition	Status received.
Failure	GSM module is not working

3.5.1 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe dynamicaspects of the system. Activity diagram is basically a flow chart to represent the flow fromone activity to another activity. The activity can be described as an operation of the system.So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join etc.

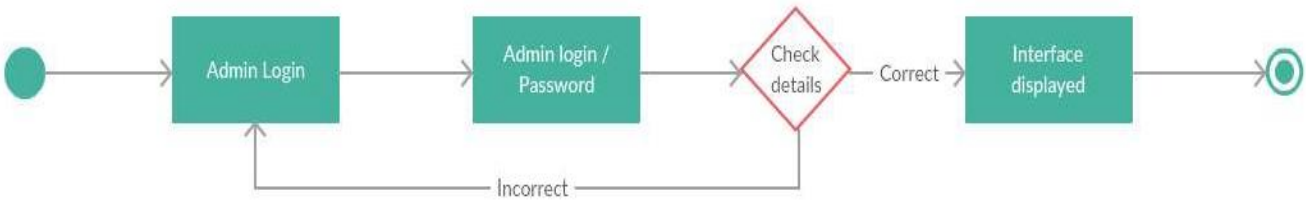


Figure No. 3.3 Admin Login

Figure No. 3.4 Lamp Details

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

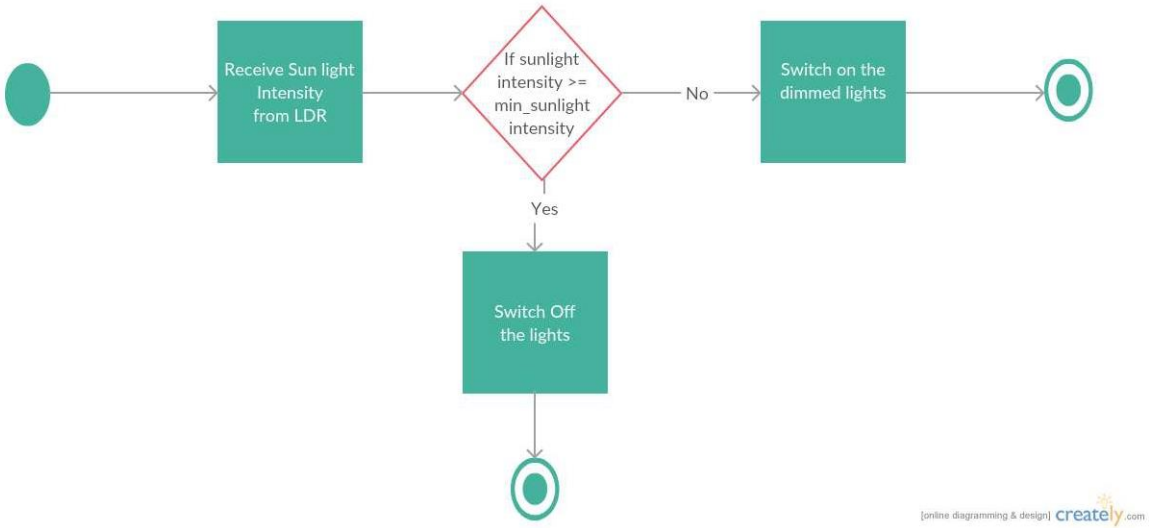


Figure 3.5 Start Street Lamps on the basis of Sunlight

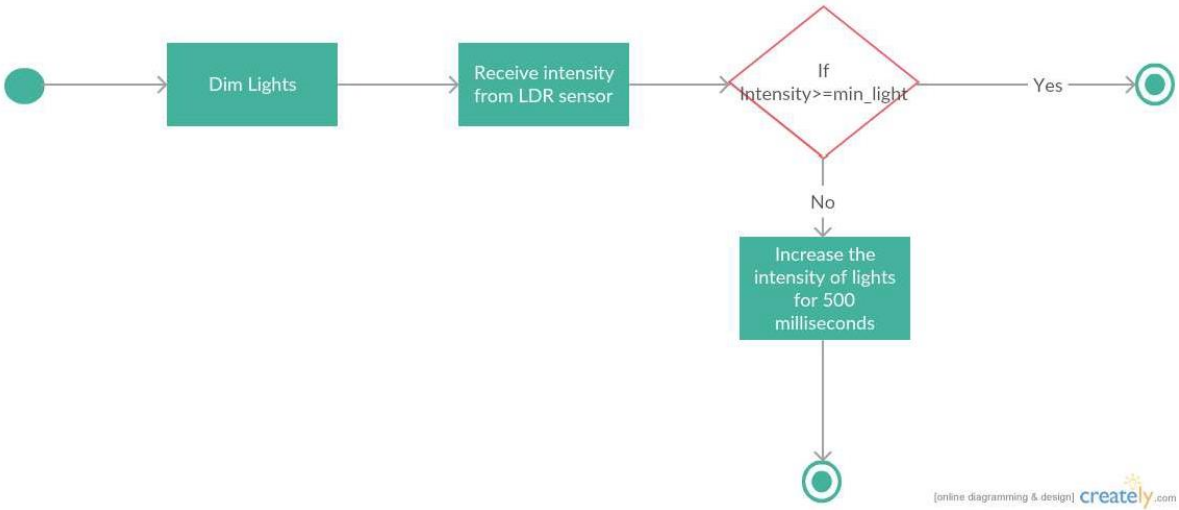


Figure 3.6 Switching On the light with high intensity

3.6 DESIGN

The Design Phase seeks to develop detailed specifications that emphasize the physical solution to the user's information technology needs. The system requirementsand logical description of the entities, relationships, and attributes of the data that were documented during the Requirements Analysis Phase are further refined and allocated intosystem and database design specifications that are organized in a way suitable for implementation within the constraints of a physical environment

3.6.1 ARCHITECTURAL DIAGRAM

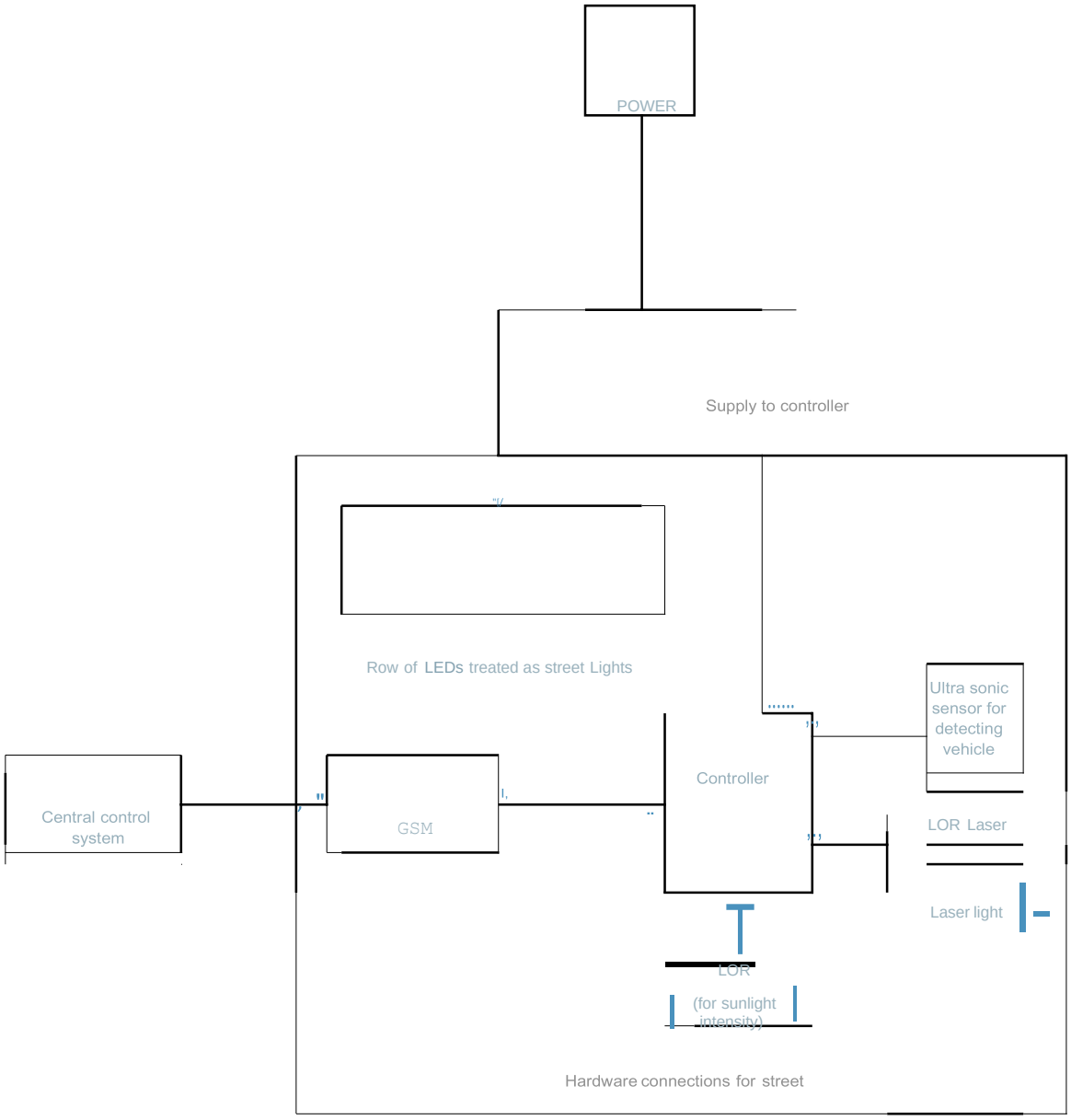
A system architecture is a conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

A system architecture comprise system components that will work together to implement the overall system. The system architecture diagrams help to understand, clarify, and communicate ideas about the system structure and the user requirements that the system must support. It's a basic framework which is used at the system planning phasehelping partners understand the architecture, discuss changes, and communicate intentions clearly. System architecture conveys the informational content of the elements comprising a system, the relationships among those elements, and the rules governing those relationships. The architectural components and set of relationships between these components that an architecture description may consist of hardware, software, documentation, facilities, manual procedures, or roles played by organizations or people.

EXPLANATION

The system consists of microcontroller atmega328 which is responsible for controlling the functioning of ultrasonic sensor and the LDR. Based on the intensity valuesthe lamp LEDs are turned on or off. Output of LDR and Ultrasonic are fetched to controllerand status of lamp will be send periodically to central system through GSM module. A common power supply is used to control controller as well as the LED lamps.



ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

Figure No. 3.6.1 System Architecture Diagram

3.6.2 FLOWCHART

A flowchart is a visual representation of the sequence of steps and decisionsneeded to perform a process. Each step in the sequence is noted within a diagram shape. Steps are linked by connecting lines and directional arrows. This allows anyone to view theflowchart and logically follow the process from beginning to end.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

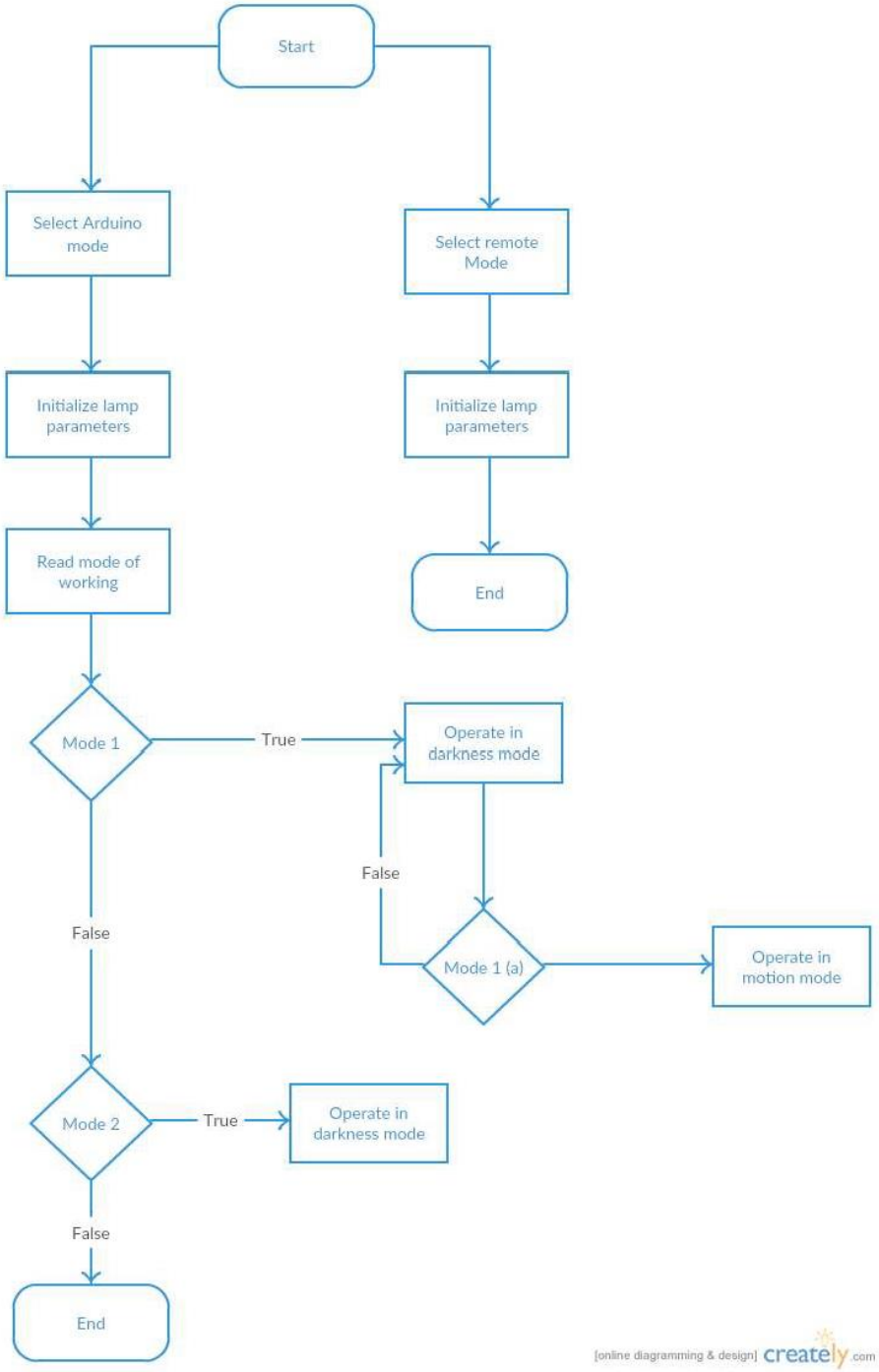


Figure No. 3.6.2 Flow Chart of System

3.6.3 DATAFLOW DIAGRAM

A data flow diagram (DFD) illustrates how data is processed by a system interms of inputs and outputs. As its name indicates its focus is on the flow of information, where data comes from, where it goes and how it gets stored.

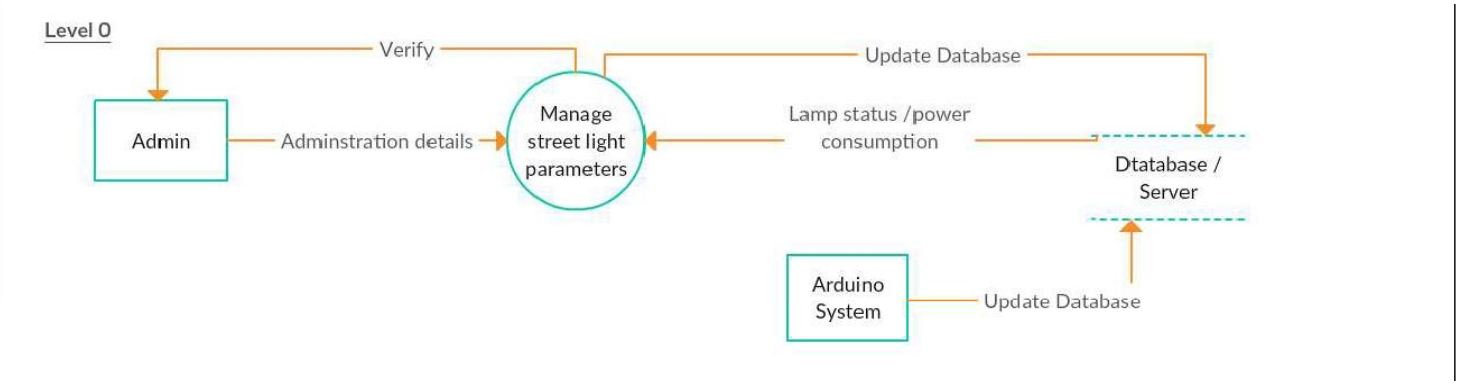


Figure No. 3.6.3 Level 0 DFD

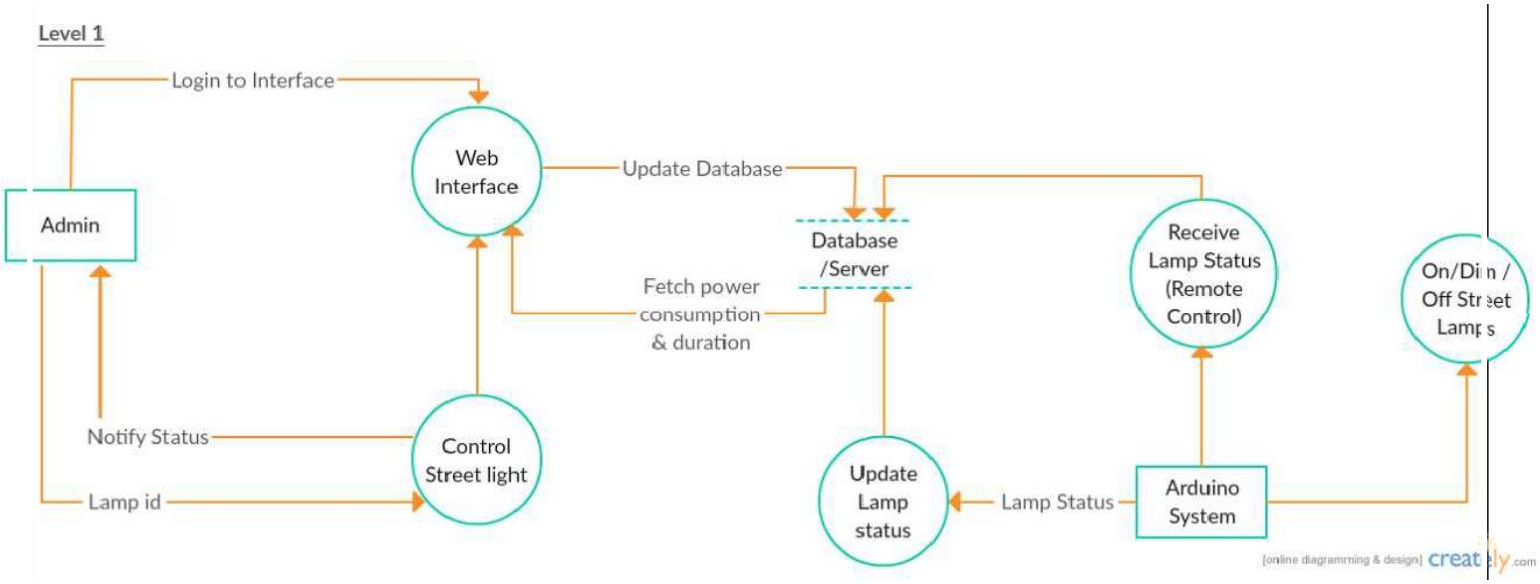


Figure No. 3.6.4 Level 1 DFD

ISSN: 2277-9655
Impact Factor: 5.164

3.7 IMPLEMENTATION

The purpose of the Implementation Phase is to deploy and enable operations of the new information system in the production environment. The major steps involved in this phaseare:

- Acquisition and Installation of Hardware and Software
- User Training
- Documentation

A prototype has been developed for the proposed system. This includes the two parts:

(i) Atmega328 Microcontroller Hardware System
(ii) Remote Server Based Software System

3.7.1 Atmega328 Microcontroller Hardware System

- Atmega328 Microcontroller

  The hardware system on the printed circuit based board with the Atmega328 Microcontroller at which all LDR sensors and GSM Module are boarded.

  Using it automatically during the less sunlight/ no sunlight the street lights will beon at the low intensity. And using LDR sensor whenever motion is detected the street lamps will glow at full intensity.

- LDR sensor and Laser Light Pair

  Motion detection using the Laser Light directed on the LDR sensor whenever the vehicle blocks the Laser Light then the corresponding LDR sensor will receive change in the intensity of light. This change in the intensity will be received by theAtmega328 microcontroller.

- Ultrasonic Sensor

  Ultrasonic sensor used to detect the moving object in combination with the LDRsensor.

- GSM Module (Receiver/Transmitter)

  GSM Module is used to send and receive the data i.e. Lamp status parameters,mode of working.

- Printed Circuit Board

  On PCB the different component like Atmega328 microcontroller, LDR Sensor,Ultrasonic, GSM Module, capacitors, resistors, IRFZ are connected.

3.7.2 REMOTE SERVER BASED SOFTWARE SYSTEM

- Server

  The Django Framework python based 24x7 running server on the static IP/website.On the server the microcontroller will send the lamp status parameters and power is calculated. These information are stored at server based SQLite database server.

- SQLite Database

  At the remote system where the server is deployed with it database schema is alsodefined. The information received from the Microcontroller and remote based status of the street lamps.

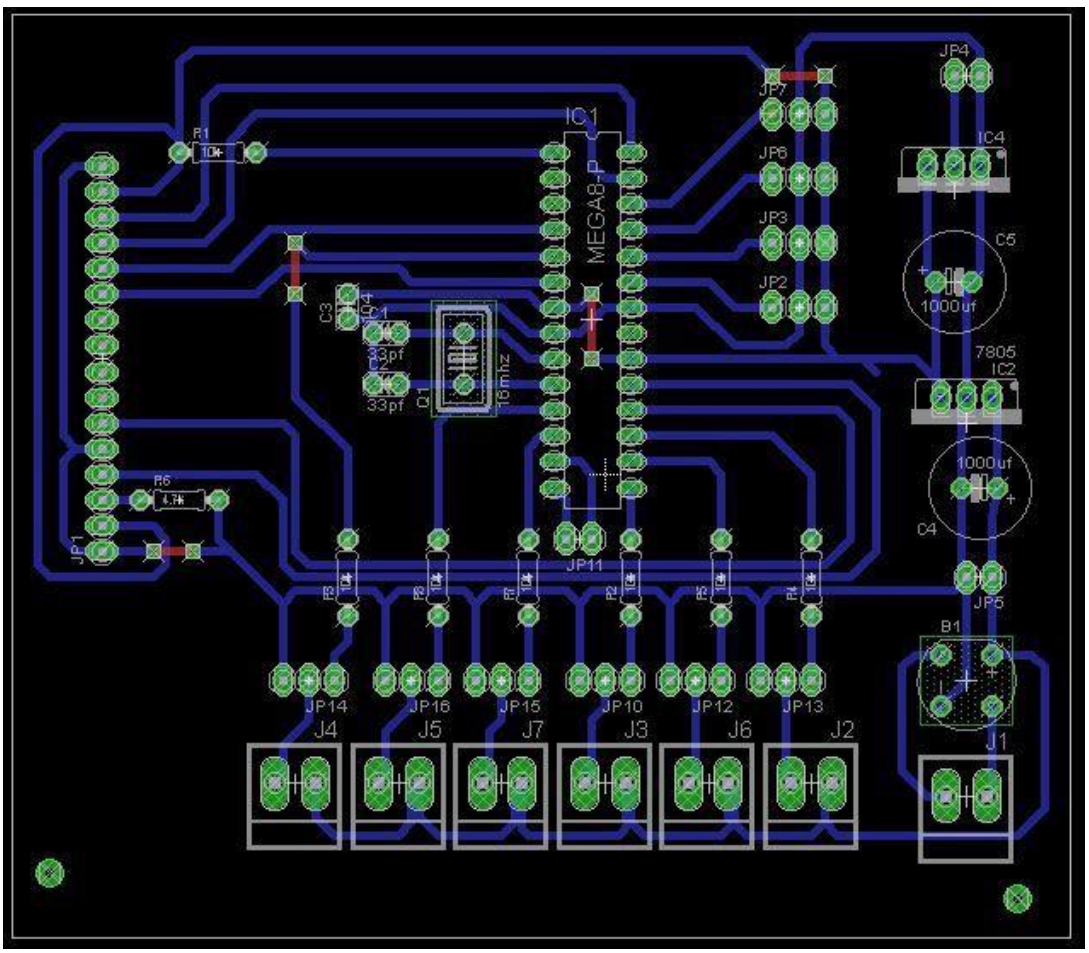
3.7.3 HARDWARE AND SOFTWARE TOOLS

- PC/Mac with Arduino programming IDE installed
- It is a software tool which facilitates the programming of Arduino Uno R3.
- PC/Mac with EAGLE PCB Design Software installed
- It is a software tool which facilitates the design of PCB circuit board layout.
- USB cable for programming Arduino boards
- It is used to connect Arduino board to the PC, which is responsible for poweringthe Arduino board.
- USB Driver for Arduino
- The driver serves the purpose of synchronizing the Arduino board with the PC.
- The source package associated with the programming workstation.

3.7.4PCB DESIGN LAYOUT ON THE EAGLE SOFTWARE

The hardware comprises of a circuit made on PCB that consists of a power supply module, relay module to control devices, an AVR family Atmega328 microcontroller with a GPRS modem to provide internet to the system and an Arduino Unoboard is used for programming.

Figure No. 5.1: Printed Circuit Board ConnectionsPCB MAKING



PROCESS:

1. First of all, the circuit layout was designed using eagle software on the computer and printed on a high gloss photo paper.

2. Then, the sheet was placed against the copper plate and pressed using an iron to get the circuit impression on the copper plate. This process is known as ironing.

3. After getting the circuit on plate, the plate was immersed in liquid solution of ferric chloride ($FeCl_3$) for about 5-10 minutes. This being more reactive replaces the copper from the plate and the circuit part is

ISSN: 2277-9655
Impact Factor: 5.164

what remains. This process is called etching.

ISSN: 2277-9655
Impact Factor: 5.164

4. Then drilling was done to make holes to set the components.

5. Then soldering was done to cover any gaps in the circuit and fix the components using asoldering iron.

6. Connect components (i.e. Atmega328 microcontroller, LDR Sensor, Ultrasonic, GSMModule, capacitors, resistors, IRFZ) on the copper sheet using the wires.

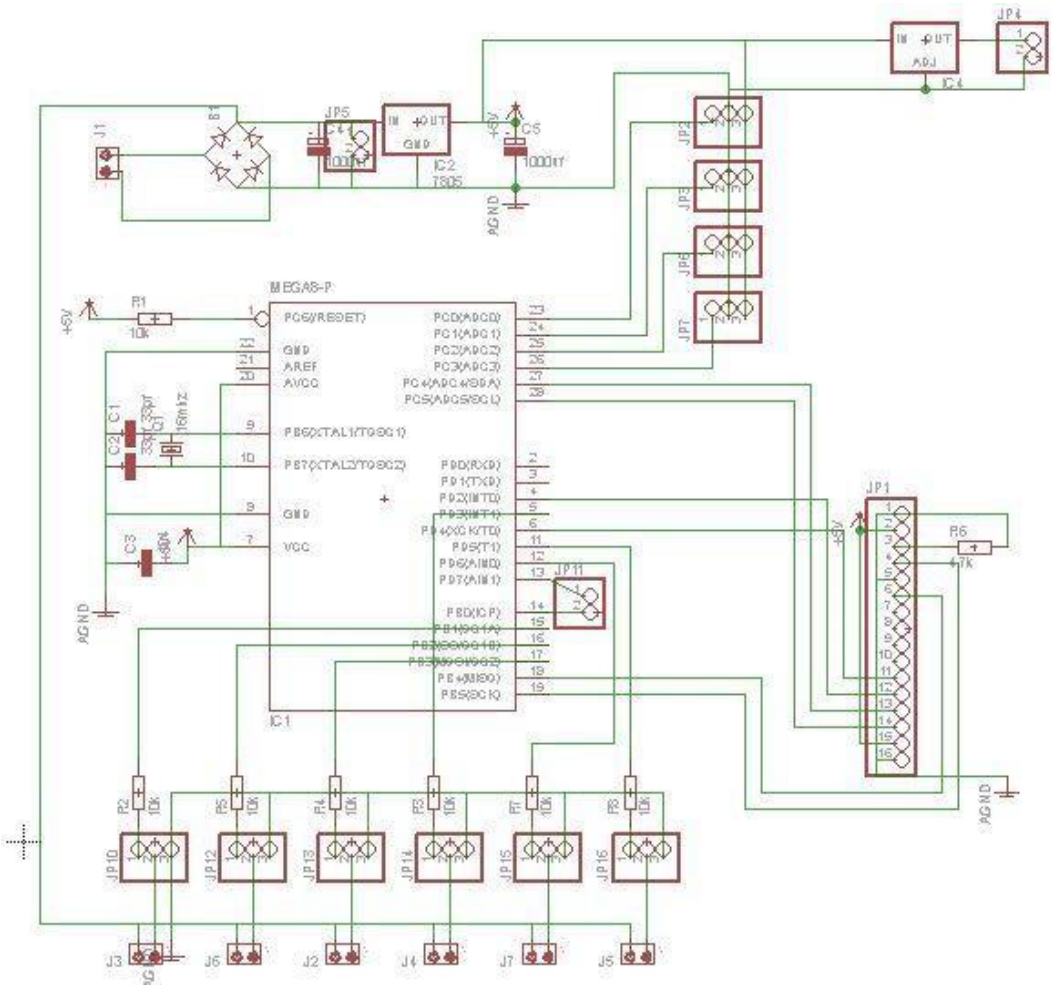


Figure 5.2: Board Connections

PIN DIAGRAM EXPLANATION

The circuit takes 230v A.C input and using a 12V step down transformer converts it into 12V DC. But this voltage contains noise and ripple which is removed usinga capacitor and pure dc voltage of 12 volts is given as input to 7805 voltage regulator. Thevoltage regulator gives the desired output of 5 volts. All the components that require 5voltsare connected to this for power. But to store the charge in case of extra voltage requirement

an extra capacitor is used as storage. This avoids the drawing of power from othercomponents. In addition, A voltage regulator is also connected in parallel to keep the currentsame. The microcontroller used is of AVR family Atmega 328 that has 28 pins with 14 digital pins and 6 analog pins. The Rxd and Txd of gprs modem is connected to pin number2 and 3 for serial data communication. At pin number 9 and 10 , a crystal oscillator of 16 Mhz is connected with addition of two capacitors of same capacity 33 pf that act as the bandpass filters. The lamps are connected via IRFZ triac which is used to smoothly vary the intensity of light. The VCC and ground connections of the SIM800L GPRS modem are given to the Arduino Uno board.

ISSN: 2277-9655
Impact Factor: 5.164

3.8 HARDWARE ARDUINO SYSTEM PROGRAM

3.8.1 Using Arduino IDE

This section consists of detailed instructions for programming theArduino board using the IDE.

Open the sketch from the source archive in the Arduino IDE and save a copy locally.

- Ensure the Arduino is NOT connect to power via the barrel connector.
- Connect the board to your programming workstation station with anappropriate USB cable.
- Set the board type to your selected Arduino board under the Tools >Boardmenu.
- Set the serial port to the port detected when you connected the Arduino boardunder the Tools > Port menu.

Description of functions used in Atmega328 microcontroller programming

This section contains a detailed description of all the functions that have been used in making based Automatic Street Light system.

- *SoftwareSerial.h*

The Arduino hardware has built-in support for serial communicationon pins 0 and 1 (which also goes to the computer via the USB connection). The native serial support happens via a piece of hardware (built into the chip) calleda UART. This hardware allows the Atmega chip to receive serial communication even while working on other tasks, as long as there room in the 64 byte serial buffer.

The SoftwareSerial library has been developed to allow serial communication on other digital pins of the Arduino, using software to replicate thefunctionality (hence the name "SoftwareSerial"). It is possible to have multiple software serial ports with speeds up to 115200 bps. A parameter enables inverted signaling for devices which require that protocol.

- *Void Setup()*

The setup() function is called when a sketch starts. Use it to initializevariables, pin modes, start using libraries, etc. The setup function will only run once,after each powerup or reset of the Arduino board.

- *Void Loop()*

After creating a setup() function, which initializes and sets the initialvalues, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

The basic structure of our program is fairly simple and it runs in twoparts. These two required parts, or functions, enclose blocks of statements.

```
void setup()
{
  statements;
}
void loop()
{
  statements;
}
```

Here setup() is the preparation, loop() is the execution. Both functions are required for the program to work.

The setup function follows the declaration of any variables at the very beginning of the program. It is the first function to run in the program. It is run only once, and is used to set pin Mode or initialize serial communication.The loop function follows next and includes the code to be executed continuously-reading inputs, triggering outputs, etc.

Functions used:

The mode of operation used in our code is Read Semi-Blocking Mode.In thismode, a call to any of the read() or readBytes() methods will remain blocked. The durationfor which they will be blocked is specified by the newReadTimeout parameter in milliseconds.

To specify this mode of operation, *setComPortTimeouts()* method with a timeout mode of *TIMEOUT_READ_SEMI_BLOCKING* is called.

- *delay()*

It pauses the program for the amount of time (in milliseconds)specified as parameter.

SYNTAX

delay(ms)

PARAMETERS

ms: the number of milliseconds to pause (*unsigned long*)*RETURN*

Nothing.

- *digitalWrite()*

It is required to set the digital pin either HIGH or LOW.

If the pin has been configured as OUTPUT with pinMode(), its voltage will be set to the corresponding value *i.e.*, 5V (or 3.3V on 3.3V boards) for HIGH and 0V (ground) for LOW.If the pin has been configured as INPUT, digitalWrite() will enable (HIGH) or disable (LOW) the internal pull up on the input pin.

If the pinMode() is not set to OUTPUT when connecting the LED to a pin, then while calling the digitalWrite(HIGH), the LED may appear dim. Without explicitly setting pinMode(), digitalWrite() will enable the internal pull-up resistor, which acts like alarge current-limiting resistor.

SYNTAX

digitalWrite(pin, value)

PARAMETERS

pin: the pin number value:
HIGH or LOW*RETURN*

None**.**

- *digitalRead()*

It reads the value from a specified digital pin, either HIGH or LOW.

SYNTAX

digitalRead(pin)

PARAMETERS

pin: the number of the digital pin you want to read (*int*).

RETURN

HIGH or LOW

- *analogWrite()*

This function writes an analog value to a pin. It is used to light the LED at varying brightness or drive a motor at various speeds. After a call to **analogWrite()**, the pinwill generate a steady square wave of the specified duty cycle until the next callto **analogWrite()** (or a call to **digitalRead()** or **digitalWrite()** on the same pin).

On most Arduino boards (those with the ATmega168 or ATmega328), this function works on pins 3, 5, 6, 9, 10, and 11. Before calling analogWrite(), a call to pinMode() is not required to set the pin to OUTPUT.

SYNTAX

analogWrite(pin, value)

PARAMETERS

pin: the pin to write to.

value: the duty cycle: between 0 (always off) and 255 (always on).

RETURNS

Nothing

- *begin()*

This function sets the data rate in bits per second (baud) for serial data transmission. For communication with the computer, one of these data rates is used: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, or 115200. An optional

second argument configures the data, parity, and stop bits. The default is 8 data bits, no parity and one stop bit.

ISSN: 2277-9655
Impact Factor: 5.164

SYNTAX

Serial.begin(speed)

Serial.begin(speed, config)

PARAMETERS

speed: in bits per second (baud) config: sets
data, parity, and stop bits.*RETURN*

Nothing

- *print()*

This function prints data to the serial port in human-readable ASCII text.

Numbers are printed using an ASCII character for each digit. Floats are similarly printed asASCII digits, defaulting to two decimal places. Bytes are sent as a single character. Characters and strings are sent as it is. For example:

- Serial.print(78) gives "78"
- Serial.print(1.23456) gives "1.23"
- Serial.print('N') gives "N"
- Serial.print("Hello world.") gives "Hello world."
- An optional second parameter specifies the base (format) to use; permitted valuesare BIN (binary, or base 2), OCT (octal, or base 8), DEC (decimal, or base 10), HEX (hexadecimal, or base 16). For floating point numbers, this parameter specifies the number of decimal places to use. For example:
- Serial.print(78, BIN) gives "1001110"
- Serial.print(78, OCT) gives "116"
- Serial.print(78, DEC) gives "78"
- Serial.print(78, HEX) gives "4E"
- Serial.println(1.23456, 0) gives "1"
- Serial.println(1.23456, 2) gives "1.23"
- Serial.println(1.23456, 4) gives "1.2346"
- Flash-memory based strings can be sent to Serial.print() by wrapping them withF(). For example:
- Serial.print(F("Hello World"))

To send a single byte, use Serial.write().

SYNTAX

Serial.print(val)

Serial.print(val, format)

PARAMETERS

val: the value to print - any data type.

format: specifies the number base (for integral data types) or number of decimalplaces (for floating point types).

RETURN VALUE

size_t (long): print() returns the number of bytes written, though reading that numberis optional.

- Function for the object(vehicle) int distance = pulseIn(inputPin, HIGH) - distance=duration * reading duration of return of pulse in microsecond

int object_distance= distance/29/2 -in cm (29 is a constant for ultra-sonic device).

Interface GSM Module to Arduino-Send and Receive SMS

A GSM Module is basically a GSM Modem (like SIM 900) connected to aPCB with different types of output taken from the board – say TTL Output (for Arduino, 8051 and other microcontrollers) and RS232 Output to interface directly with a PC (personal computer). The board will also have pins or provisions to attach mic and speaker, to take out +5V or other values of power and ground connections. These type of provisions vary with different modules.

Lots of varieties of GSM modem and GSM Modules are available in the market to choose from. For our project of connecting a gsm modem or module to arduino and hence send and receive sms using arduino – its always good to choose an **arduino compatible GSM Module** – that is a GSM module with TTL Output provisions.

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.

Connecting GSM Module to Arduino

There are two ways of connecting GSM module to arduino. In any case, thecommunication between Arduino and GSM module is serial. So we are supposed to use serial pins of Arduino (Rx and Tx). Connect the Tx pin of GSM module to Rx pin of Arduinoand Rx pin of GSM module to Tx pin of Arduino. **GSM Tx –> Arduino Rx** and **GSM Rx**

–> Arduino Tx. Now connect the ground pin of arduino to ground pin of gsm module. Wemade 3 connections and the wiring is done. Now we can load different programs to communicate with gsm module and make it work.

The circuit diagram given below is the circuit diagram to connect gsm module to arduino – and hence use the circuit to send sms and receive sms using arduinoand gsm modem.

The program has two objectives as described below:-

1) Send SMS using Arduino and GSM Module – to a specified static IP address(website) inside the program

2) Receive SMS using Arduino and GSM Module – to the SIM card loaded in theGSM Module.

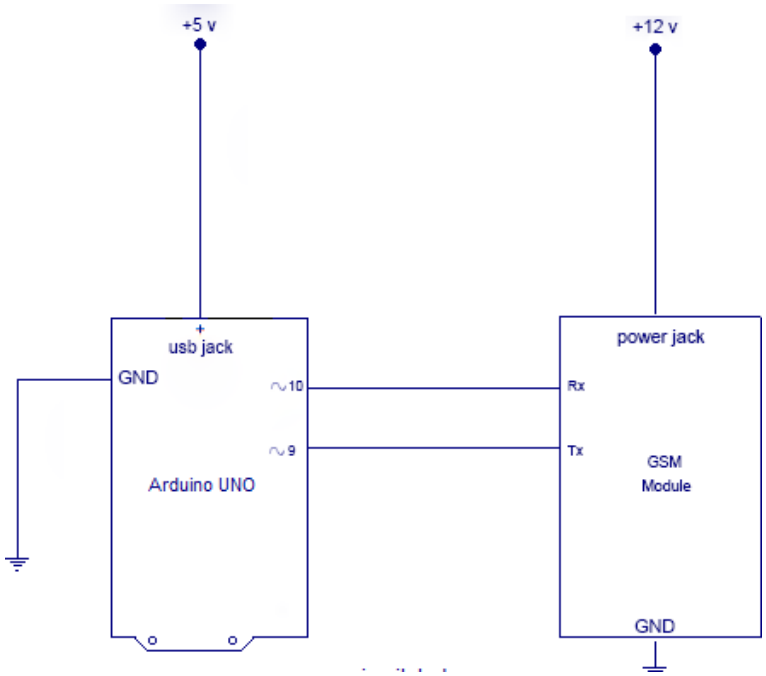


Figure No. 5.3 Connection between GSM and Arduino

Functions used in the GSM Module

- **SoftwareSerial SIM900(7, 8)** - soft serial 7,8 used for assigning tx and rx respectively.
- **Serial.available()** – checks for any data coming through serial port of arduino. Thefunction returns the number of bytes available to read from serial buffer. If there is no data available, it returns a -1 (value less than zero).
- **Serial.read()** – Reads all the data available on serial buffer (or incoming serial dataif put otherwise). Returns the first byte of incoming serial data.
- **SIM900.begin(9600)** - setting baud rate for sim900L
- **Serial.begin(9600)** - opens serial port, sets data rate to 9600 bps

GSM modules work on AT commands

- **ATE0** – IT is used to turn off the Echo of GSM shield.
- **AT** – Just to check that your GSM module is working fine.
- **AT + CMGF = 1** – This command will convert the message style to text. In other words we are telling our shield that we are expecting a text message.
- **AT+CNMI=1,2,0,0,0** – This command will alert our GSM shield and now whenever it will receive message, it will automatically send an alert on the serial port.

void SubmitHttpRequest()

- **SIM900.println("AT+CSQ")** – AT+CSQ AT command returns the signalstrength of the device.
- **SIM900.println("AT+CGATT?")** – Check the status of Packet service attach. '0'implies device is not attached and '1' implies device is attached.
- **SIM900.println("AT+SAPBR=3,1,\"CONTYPE\",\"GPRS\"")** – Set the connection type to GPRS
    - *SIM900.println("AT+SAPBR=3,1,\"APN\",\"AIRTELGPRS.COM\"")– setting apn*
- **SIM900.println("AT+SAPBR=1,1")** – Enable the GPRS
- **SIM900.println("AT+HTTPINIT")** – AT+HTTPINIT AT command initializesthe HTTP service

Void send_data_url()

- **SIM900.println("AT+HTTPPARA=\"URL\",\"http://iotdemoo.herokuapp.co m/next/\"")** – set SMS mode to text , AT+HTTPPARA AT command sets up HTTP parameters for the HTTP call
- **SIM900.println("AT+HTTPACTION=0")** – AT+HTTPACTION AT Commandis used perform HTTP actions such HTTP GET or HTTP post 0: READ 1:POST 2:HEAD
- **SIM900.println("AT+HTTPREAD")** – AT+HTTPREAD AT command is usedto read the HTTP server response.
- **SIM900.print("AT+HTTPPARA=\"URL\",\"http://iotdemoo.herokuapp.com/ street_light_status/")** – sending lamp status to remote

5.2 Web Server Software Implementation

Django is a free and open-source web framework, written in Python, whichfollows the model-view-template (MVT) architectural pattern. It is maintained by the Django Software Foundation (DSF), an independent organization established as a non-profit.

Model View Controller or MVC as it is popularly called, is a software designpattern for developing web applications. A Model View Controller pattern is made up of thefollowing three parts:

Model - The lowest level of the pattern which is responsible for maintaining data.

View - This is responsible for displaying all or a portion of the data to the user.

Controller - Software Code that controls the interactions between the Model and View.

MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response.

DJANGO MVC - MVT Pattern

The Model-View-Template (MVT) is slightly different from MVC. In fact the main difference between the two patterns is that Django itself takes care of the Controller part (Software Code that controls the interactions between the Model and View),leaving us with the template. The template is a HTML file mixed with Django Template Language (DTL).

The following diagram illustrates how each of the components of the MVTpattern interacts with each other to serve a user request −

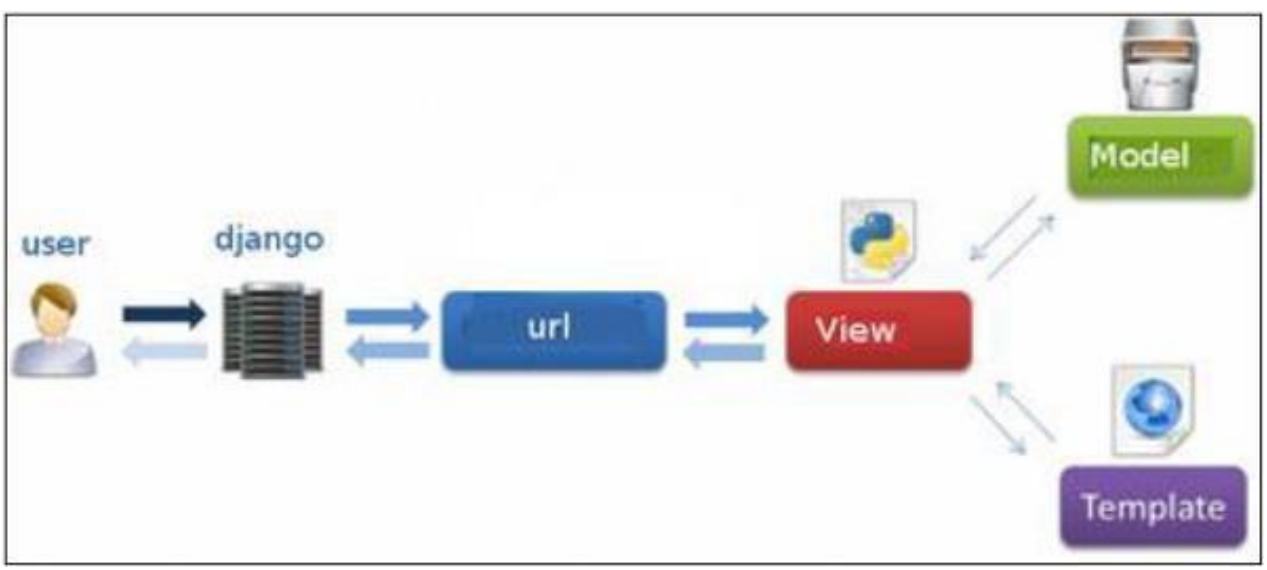


Figure No. 5.5 User connecting to server using MVT Architecture

ISSN: 2277-9655
Impact Factor: 5.164

The developer provides the Model, the view and the template then just maps it to a URL and Django does the magic to serve it to the user.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even forsettings files and data models. Django also provides an optional administrative create, read,update and delete interface that is generated dynamically through introspection and configured via admin models.Django development environment consists of installing and setting up Python, Django, and a Database System SQLite.

WebServer: Django comes with a lightweight web server for developing and testing applications. This server is pre-configured to work with Django, and more importantly, it restarts whenever you modify the code.

Templates: URL Pattern Used

r= iotdemoo.herokuapp.com

```
# Street Light Automation
    url(r'^$', views.index),

    # url(r'^image_change/$', views.change_image), url(r'^change_control_type/$',

    views.change_control_type),url(r'^light_on/$', views.light_on),

    url(r'^light_off/$', views.light_off),

    url(r'^light_power/$', views.light_power),

    url(r'^image_reload/$', views.image_reload),

    url(r'^next/$', views.next),

    url(r'^user_string/(?P<s1>.+)/(?P<s2>.+)/$', views.user_string),

    url(r'^street_light_status/(?P<status>.+)/$', views.street_light_status)
```

Models: Database Schema defined for lamps structure:

class Lamp(models.Model: // the database attributes of each lamps day = models.CharField(max_length=20)

week_day = models.IntegerField(default=0)

last_status = models.IntegerField(default=0)

last_update_time = models.DateTimeField()power = models.FloatField(default=0)

Views : Function Used

Class defined of lamps(Models)

class StreetLight(models.Model): // Class of complete StreetLight System comprises of 6 streetlamps.

l1 = models.IntegerField(default=0)

l2 = models.IntegerField(default=0)

l3 = models.IntegerField(default=0)

l4 = models.IntegerField(default=0)

l5 = models.IntegerField(default=0)

l6 = models.IntegerField(default=0)

Functions defined corresponding to the URL patterns are:

def index(request):- Render the current database status of lamps to the user.

def image_reload(request):- Reload the home page of the website and send http responsein string form

def light_on(request):- Arduino sends the request to turn on or dim the lamps at the serverso that it update the database and calculate the power consumption.

ISSN: 2277-9655
Impact Factor: 5.164

def light_off(request):- Arduino send request to turn off the lamps at the server so that itupdate the database and calculate the power consumption.

def light_power(request): Calculate the power consumption of lamps.

def street_light_status(request, status): Show the current status of lamps.

def login(request): Login Authentication function.

def logout(request): Logout function.

3.9 TIMELINE PLANNING

GANTT CHART

A Gantt chart is a horizontal bar chart developed as a production control toolwhich is frequently used in project management, a Gantt chart provides a graphicalillustration of a schedule that helps to plan, coordinate, and track specific tasks in a project.

A Gantt chart is constructed with a horizontal axis representing the total timespan of the project, broken down into increments Horizontal bars of varying lengths represent the sequences, timing, and time span for each task.

ISSN: 2277-9655
Impact Factor: 5.164

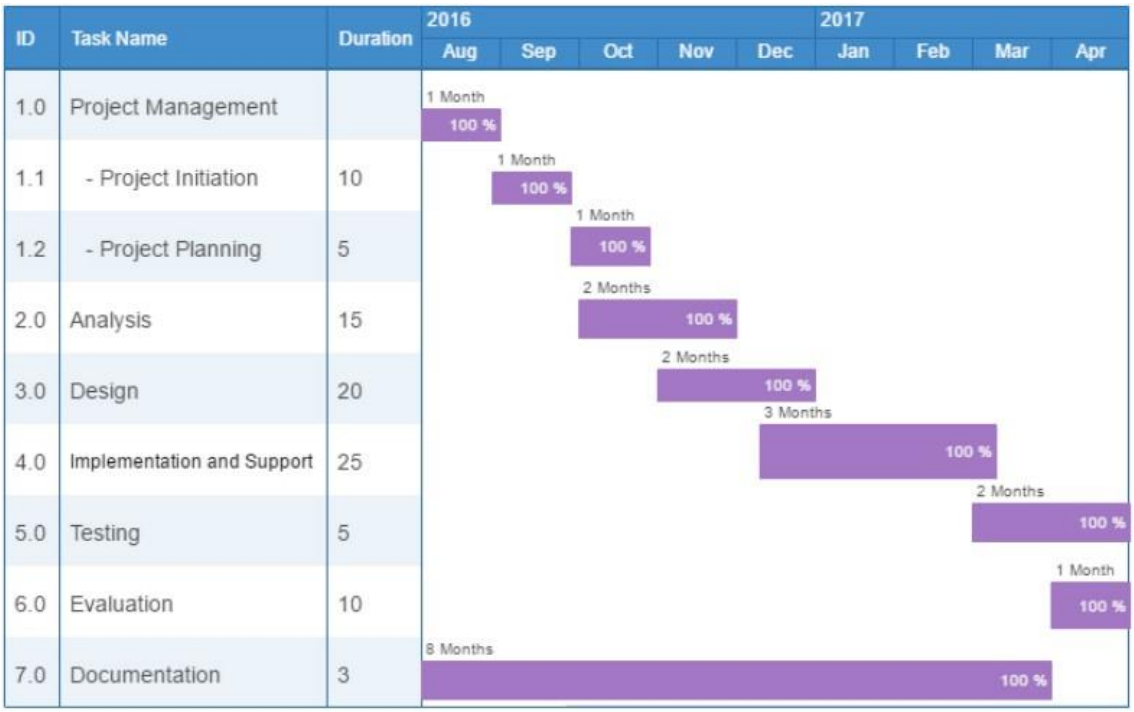


Figure No. 3.9 Gantt chart

RESULTS

TESTING

Testing is a process of executing an application with the intent of finding software bugs. It is the process of validating and verifying that an application meets the technical requirements that guided its design and development and it works as expected. Testing is a process rather than a single activity.

Static Testing – It can test and find defects without executing code. It involves reviewing of the documents and static analysis.

Dynamic Testing – In this, software code is executed to demonstrate the result of running test cases. The results are evaluated under the test and matched with the expected results. This helps us to decide whether we have finished testing and passed the test cases.

TEST CASES AND SCENARIOS

The testing of project was carried out in different scenarios. The project was tested for various object during certain climatic condition. Each sensor was tested individually as a unit and then combined with other component to perform the integration testing. The remote controlling was also tested separately without integrating with hardware.

ISSN: 2277-9655
Impact Factor: 5.164

Notations for Sensors

- Ultrasonic sensor - controls Lamp L1 and L2
- LDR 1- Detect day or night
- LDR-laser pair1 – control Lamp L2 and L3
- LDR-laser pair2 – control Lamp L3 and L4
- LDR-laser pair3 – control Lamp L4 and L5
- LDR-laser pair4 – control Lamp L5 and L6


LAMP OFF


LAMP FULLY POWERED LAMP

PARTIALLY POWERED

ISSN: 2277-9655
Impact Factor: 5.164

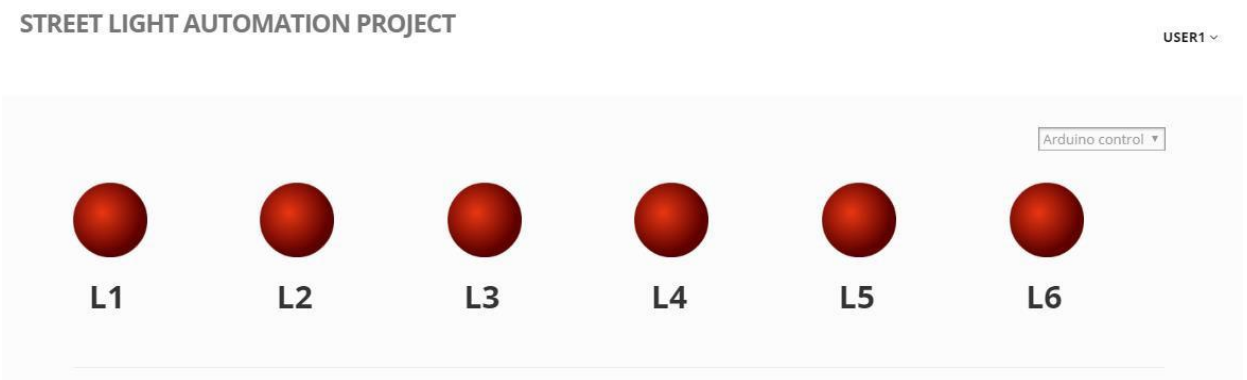
Test Cases:-

Case 1: During day time when there is sunlight

The LDR 1 attached to microcontroller returns some voltage values. This voltage value is compared with the reference value and because of this all the lights are switched off and every laser LDR-LASER pair and ultrasonic sensor are disabled.

Reference voltage = 4 volts

Voltage returned >4 volts



Case 2: During day time when there is no sunlight

The LDR 1 attached to microcontroller return some voltage values. This voltage value is compared with the reference value. Since the weather was not good, the microcontroller enables all the laser-LDR pair, ultrasonic sensor and also switch on all the lights with less intensity.

Reference voltage =4 volts

Voltage returned <4 volts

In this case the system will work as night mode

Case 3: During night time when there is vehicle (ultrasonic)

In this case, the ultrasonic sensor detects the vehicle based on reception of sound wave and generate a digital output which is sent to the digital pin of microcontroller.

ISSN: 2277-9655
Impact Factor: 5.164

Depending on the value received the vehicle is detected if the distance returned is less than2m.

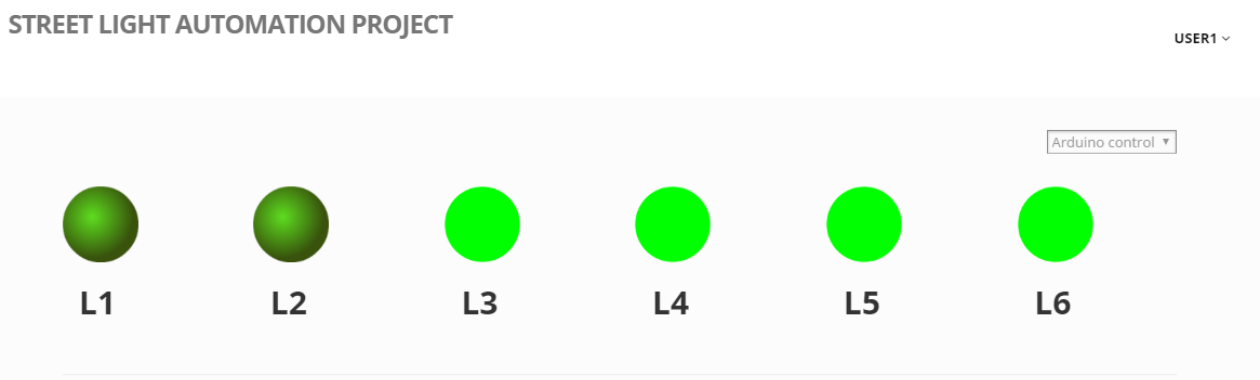
On detection of vehicle the corresponding LED1 and LED2 are powered with full voltagefor a particular period of time.

Usually when there is no object the distance return is greater than 2m and LED are dimly litat this time.

Reference distance: 2m Distance

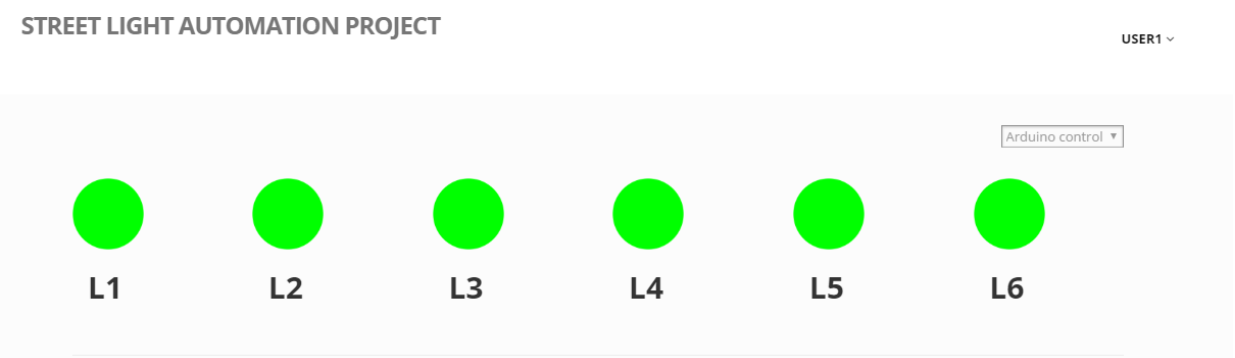
read: less than 2mFull Voltage

supplied: 12v



Case 4: During night time when there is no vehicle (ultrasonic)

In this case, the ultrasonic sensor return value greater than the referencevalue and due to this no change in voltage occurs at LEDs implying no vehicle.

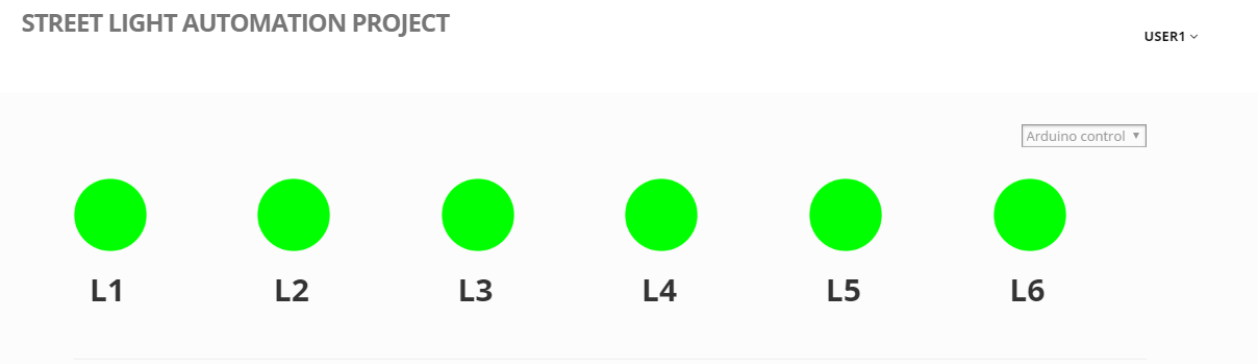


Case 5: During night time when there is no vehicle (laser-LDR)

In this case, the laser light strike the LDR directly, because of this the voltage drop across LDR is less and voltage returned by LDR exceeds reference voltage implying no vehicle. In this case the light are lit dimly.

Reference voltage: 4 volts

Returned voltage: greater than 4 volts



Case 6: During night time when there is vehicle (laser-LDR pair)

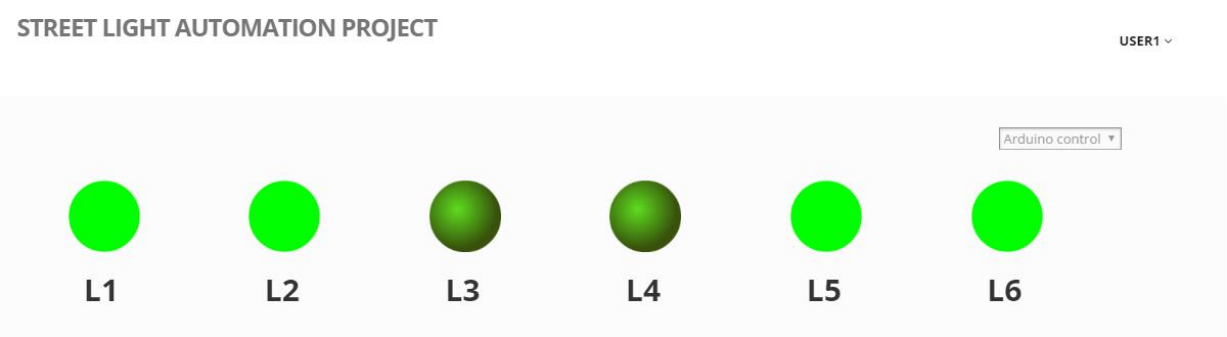
In this case, the laser light strike the vehicle directly because of this the voltage drop across LDR is high and voltage returned by LDR is less than the reference voltage implying vehicle presence. Once the vehicle is detected the corresponding LED arepowered ON. Here LDR–LASER pair 2 detect a vehicle because of this LED3 and LED4 are fully powered.

Reference voltage: 4 volts

Returned voltage: less than 4 voltsFull

Voltage supplied: 12v

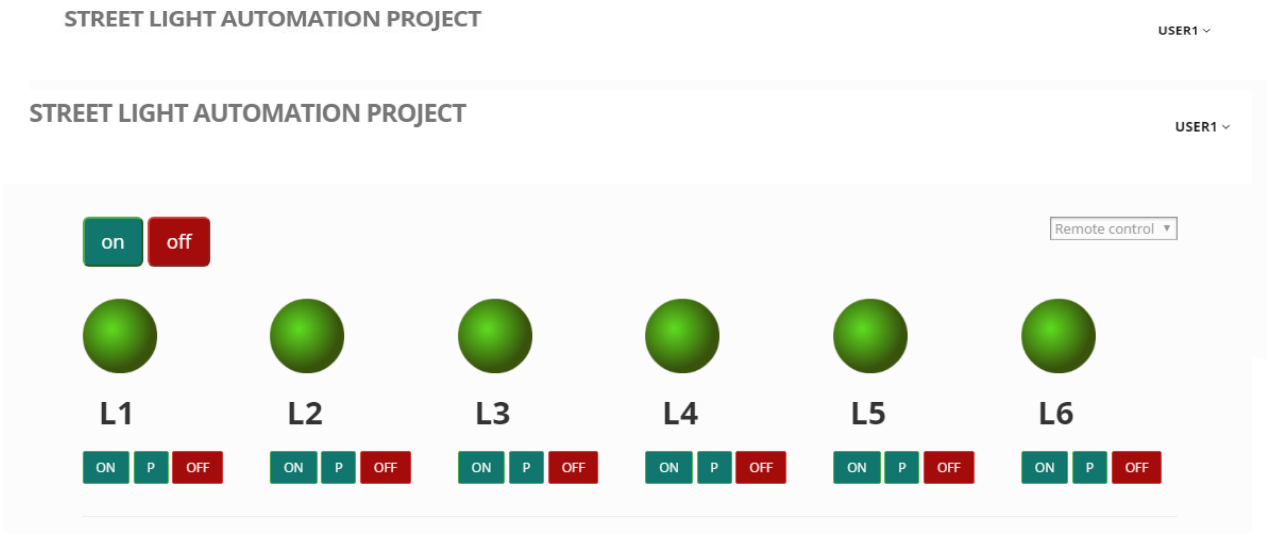
ISSN: 2277-9655
Impact Factor: 5.164



Case 7: When Remote control is on

In this case all the controlling of hardware is disabled and each and every lamp is controlled remotely. Central server can control intensity also.

Here all lamp are switch off:-

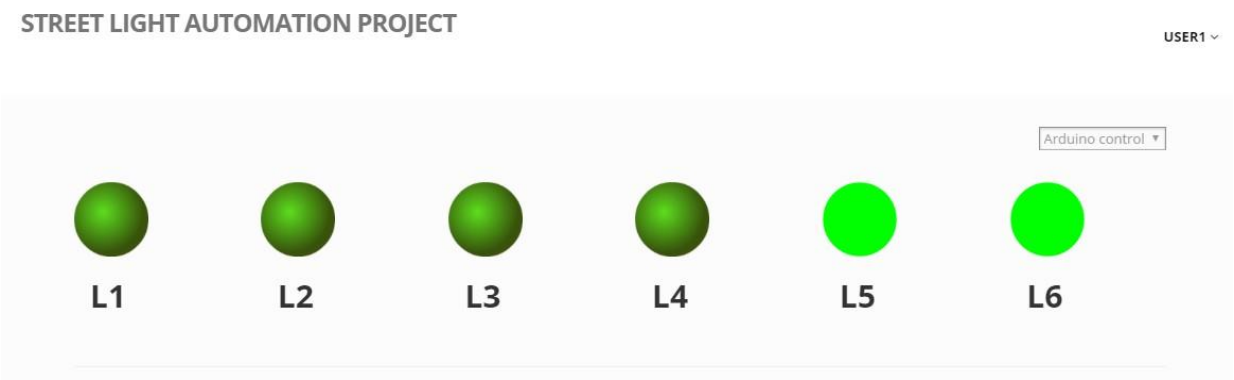


Here all lamp are switch on:-

Case 8: When there are multiple vehicles

In this case both ultrasonic and LDR returned values are greater than the reference values,due to which the corresponding LEDS are fully powered.

Here ultrasonic and LDR-LASER pair 2 has detected the vehicle, due to which LED1, LED2, LED3, LED4 are fully powered.



4.CONCLUSION

4.1WORK CARRIED OUT

This phase describes a new intelligent street lighting system which integrates new technologies available on the market to offer higher efficiency and considerable savings. The control system is the intelligent management of the lamp posts by sending data to a central station by GSM wireless communication. The system maintenance can be easily and efficiently planned from the central station, allowing additional savings.

Estimated net saving of system for our prototype containing 6 led lamps is 45.8 % ,
Actual net saving of system
Case 1 : When density is high – The time interval between 2 consecutive vehicle is 3 second.The deviation comes out to be 34.37.
Case 2 : When density is low – The time interval between 2 consecutive vehicle is 8 second. The deviation comes out to be 53.67.

The system described can effectively save energy by reducing the power consumption as per requirement. Since this is a sensor based system, so it is self-controlled and automated system. The system is also flexible for any modification or further expansion such as interfacing of new sensors, connecting surveillance camera for the security purpose, etc. This project of Smart Street Light System is a cost effective, practical, eco-friendly and the safest way to save energy. It efficiently saves the energy by replacing the conventional bulbs by LEDs and by automatic switching/dimming of LEDs as and when required. Main drawbacks of this system are the initial cost and maintenance. However large scale implementation of this proposed system will definitely reduce the overall cost of the project up to great extent.

4.2 FUTURE WORK

The project has scope in various other applications like :-

- Solar panel and motion sensor integration
- Practical application and economic feasibility to be checked for.
- Effects of varying weather conditions on the functioning of lighting system.
- Developing an application that keeps a log and monitors the functioning of the street lights.
- Checking the quality of lighting (illuminance).
- Examining the integration with existing power circuitry.
- Develop a separate network for communication with wireless sensors.

6. ACKNOWLEDGEMENTS

We would like to express our sincere thanks and gratitude to everybody who motivated us to complete this research.

REFERENCES

[1] GOI (Government of India) and Central Electricity Authority, All India Electricity Statistics (2014-2015), General Review. 2015.
[2] Automatic Street Light Control System Using Microcontroller MUSTAFA SAAD, ABDALHALIM FARIJ, AHAMED SALAH and ABDALROOF ABDALJALIL .2014.

[3] Energy Efficient Outdoor And Office Lighting System Based On Zigbee Network Nishanth ,Nirmalakumari ,Ramesh S.2012