CHERUIYOT ISAAC LANG'AT

A PROJECT REPORT ON

AUTOMATED SWITCHING SYSTEM FOR SMART LIGHTS

Submited to 2019 China-US Young Maker Competition

DECLARATION

I declare that this project entitled "Automated Switching for Smart Lights" is my work and that all sources that I have used have been indicated and acknowledged by means of complete references. No part of this project may be reproduced without prior permission of I, the author.

……………………………

Cheruiyot Isaac Langat

+254711709566

Isaaclangat001@gmail.com

GLOSSARY

List of Tables

LIST OF EQUATIONS

List of Figures

ABSTRACT

This report presents the design and implementation of an ***automated and remote switching for smart lights***, its objective being to use ambient light to control the status of a lighting system and varying the lighting colour to suit a particular occasion-***adaptive adjustment***.

The report also details the design of remote light switching that incorporates lighting control via a smartphone.

The lighting system in this context could be situated in an office, home or factory, but the adaptive adjustment specification would most appropriately suit home lighting system. The lights will be able to automatically turn ON or OFF depending on ***ambient lighting level***. The individual as well selects the desired colour of the light and brightness level which is then relayed to a microcontroller that sets the preferred brightness level by determining the appropriate duty cycle of the ***PWM signal***.

KEYWORDS: *Adaptive adjustment*, *LED, Automated control*, *Remote control*, *Smart lights*

LIST OF ABBREVATIONS AND SYMBOLS

LED- Light-emitting diode

PWM-Pulse-Width Modulation

RGB- Red, Green, Blue (referring to three-part LEDs)

PC-Personal Computer

IC- Integrated Circuit

PFC- Power Factor Correction

Table of Contents

1. Introduction

BACKGROUND

Light is a fundamental aspect of human life. Light systems constitute the bulk of energy consumption in a home or an average residential area. As such, there has been an ever-growing human need to make it more suitable, its control more intuitive, seamless and simplified. Moreover, energy cost saving in the lighting system has been a factor to consider in many lighting control designs.

According to (Margaret P. Calkins, 2013), every colour affects our mood in a different way and carries different energy. When a person is happy, for instance, he or she usually likes warm colours such as a red and orange. They excite and stimulate, focus one's visual sensitivity, stir one into action and create a sense of vigour. When one is sad he or she prefers cold colours such as blue as it slows down the rate of respiration and heartbeat. It gives off a soothing effect and predisposes one to contemplation. Colours can also create a warm feeling in the room and they can make a room seem smaller or larger. Red and other warm colours, for instance, make a room appear smaller and raises the room's temperature. Cool colours on the other hand, make a room appear bigger and lower the room's apparent temperature.

PROBLEM STATEMENT

Light control majorly consists of either or both of the following:

a) Brightness adjustment
b) Adaptive adjustment
c) Basic ON/OFF switching

Brightness control is concerned with varying the brightness of the lamps to achieve the desired light level.

Adaptive adjustment on the other hand is concerned with selecting the appropriate and most suitable lighting colour to suit a particular mood, setting or environment. As earlier discussed, adaptive lighting adjustment is an important area to human living for functional and architectural lighting purposes as well as for physiological purposes in the sense that the mood in human beings has been shown to be affected by the colour of the light in their surroundings.

Also, failure to turn off lights in either commercial, industrial or residential premises has been a leading cause of in efficient power use and wastage far as lighting is concerned.

In response to these needs, we proposed a multicolour focusable LED light selection mechanism that the user can vary as per their mood or intended activity e.g. reading, watching movies, sleeping, etc. to enhance concentration and comfort during the undertaking of such activities. This project also seeks to address the need for the smart light switching where the lights would sense that they are being required and turn ON automatically or vice versa.

Moreover, moving around the premises to the location off the light switches are can be very cumbersome and even encourage wastage as the user “may not want to get out of their comfort zone” to for instance turn OFF the light. This issue is being addressed in that the user is able to turn OFF the lights using a smartphone. This smartphone also has precedence over the light sensing mechanism in that if the ambient light level dictates that the lights be ON while the user prefers otherwise, he can overrule the sensor command via a smartphone.

Even with the lights ON, the user may want to adjust its brightness. We have explored the Pulse-width modulation switching capability of the Arduino to effect this brightness. The invention included setting the duty-cycle of the rectangular discrete wave produced by the Arduino output, which will directly control the apparent level of brightness of the lights.

Review of the papers and patents revealed that much work had been done in colour control and brightness level control of lighting systems but when it comes to merging the two into one pack, not much had been done yet. Or, also, the lighting control systems were found to be a little too costly.

The block diagram we proposed is quite simple enough to be implemented as a low cost system. We borrowed majorly from the patents discussed above an approach presented therein was adopted and implemented with modifications as follows:

a. The white LED be turned on or off automatically as well as via the web i.e. remotely.
b. Communication between the computer and the light system be effected via Local Area Network and Wi-Fi and with development be hosted on the Web and operate strictly via Wi-Fi which would be faster and offer real-time response with minimal delay. The method as proposed used Bluetooth which was quite limited in terms of permissible operation radius.
c. Inclusion of brightness control given the subjectivity of lamp brightness in ten discrete steps.

OBJECTIVES

1.1. Overall Objectives

a) Energy conservation

b) Integration of smart and remote lighting control

c) Adaptive lighting adjustment

1.2. Specific Objectives

Lighting accounts for a substantial portion of energy consumption in homes and offices worldwide. More importantly, a lot of artificial lighting in use is not necessary. In this project, we attempt to implement a system that will not only reduce energy consumed from lighting by using accurate naturally available ambient light sensing, but also provide a more pleasing lighting experience that will achieve one’s desired lighting intensity and colour as well as enable remote switching and adjustment of the same for convenience.

The objective of this project is to develop a mechanism that alters the colour output of an LED based lighting system in a room to positively affect the mood of the occupants, an implementation we referred to as *adaptive adjustment.*

a) Automated lighting control

b) Remote lighting control
c) Merging of remote and automated lighting control

d) Brightness control of lighting system

JUSTIFICATION

We propose an adaptive lighting system that responds to the environment in determining the appropriate light status at a given time of day using luminance detection and current light levels. A light sensor unit will be set at strategic points of the area to be illuminated e.g. next to the window.

These sensors will transmit to a microcontroller which will control ceiling-mounted LEDs. The feedback from the sensors will keep the desired colour and intensity at the area of interest, so that if there is sufficient sunlight, the LEDs need not consume as much power. Occupancy sensors will be mounted near the light and will only turn on the light when a person is detected in the area. We will demonstrate the device with two lights to show that it is adaptable to more fixtures.

1.4.2. Benefits:

1. Remote lighting control form anywhere as long as there is internet connection
2. Reduced lighting system energy consumption
3. Automate lighting control
4. Enhanced, advanced and more suitable lighting experience (e.g. can replicate sunlight, or have warmer colours at night)

1.4.3. Features:

1. User sets desired colour by selecting Red, Yellow or Green colour using a multiposition switch. He would the control this selected light via the web-page HTML form.
2. System accurately senses colour intensity for the location, and the main controller sets or resets the lighting status depending on the pre-set set-point.

3. Moreover, the user will be able to set the desired brightness of the White light once he turns it ON, remotely or automatically. An accessible off switch for each major component

2. Literature Review

2.1. Introduction

All lighting control systems can be grouped into two broad areas:

I. Logical control
II. Brightness Control
III. Colour control

Logical control is the toggling ON and OFF of a lighting system for example using a standard switch. This type of control is the most common because they allow more certainty and simplify control. Moreover, the use is stuck between two light statuses.

Brightness control is concerned with varying the brightness of the lamps to achieve the desired light level. It can be viewed as continuous control because the brightness can be adjusted to suitable levels. It is also more intuitive albeit complex in design.

Colour control is concerned with varying the colour of the lamps to achieve the desired light colour. As discussed in the problem statement, colour control of lighting systems is an important area to human living not just for functional and architectural lighting purposes but also for physiological purposes; the mood in human beings has been shown to be affected by the colour of the light in their surroundings.

Energy Management Systems, Home Automation and Smart Lighting are modern technology concepts aimed at lighting control that are fast getting popular. Their overall objective is energy efficiency and automated control of electrical systems.

Lighting is often the largest consumer of electricity in a building. In a naturally ventilated building, lighting may account for 40% or more of the total electricity cost. The overall lighting design together with the choice of lamps and fittings pays the major part in determining the energy cost of the lighting system; further reductions can be achieved by use of an effective lighting control system.

Building regulations require the use of efficient light sources and specifically requires the use of controls. Moreover, lighting conditions produced by the control system must at all times provide satisfactory illumination for the occupant of the building.

2.2. Discussion of Previous Technologies

2.2.1. Localised Manual Control

Local switches and dimmers can either be permanently hardwired as wall-mounted switches or ceiling-mounted pull switches, or may be operated by remote control.
Remote control devices based on infrared or ultrasonic signals. The receiver may be ceiling mounted near to the luminaries being controlled, so that walls or partitions may be moved without the need for re-wiring.
In a large space, the luminaries should be switchable in groups. Luminaries controlled by a single switch should cover an area with roughly a constant daylight factor.
Manual switches should be placed as near as possible to the area being controlled. According to building regulations, the required distance from the light switch to the furthest fitting that it controls should be no more than 6m, or twice the height of the fitting above floor level, whichever is greater.

2.2.2. Automatic Lighting Control

This technology involves use of mechanisms that make lighting control (switching or dimming) convenient by making it happen automatically without application of unnecessary effort. Moreover, energy efficiency is realized.

New light automation and lighting control solutions are now available to help reduce energy usage and cost by eliminating uncontrolled lighting. These solutions provide centralized control of all lighting within a room allowing easy implementation of scheduling, occupancy control, daylight harvesting etc. Many systems also support Demand Response and will automatically dim or turn off lights to take advantage of DR incentives and cost savings.
Newer control systems are using wireless mesh open standards (such as ZigBee), which provide benefits including easier installation (no need of control wires) and interoperability with other standards-based building control systems (e.g. security).

Different control approaches have/can be used to achieve automated lighting control: a.

Time Control

One of the best means of controlling light systems automatically is by time. Time switches are especially effective for buildings with fixed occupancy times.

By using a time switch (sometimes called a time clock), lights that are ON can go OFF automatically after a fixed period of operation. This is achieved when simple clock mechanisms operate a switch by operating a contact closure at set times.

A time switch, by automatically switching the lights at fixed intervals, ensures convenience and saves energy. Besides saving the cost of electricity, the lamp life is preserved.

However, there should be some form of override to the time switch should there be activity beyond the set times. This should incorporate an extension timer to prevent lights being left all night.
Time switches are effective for external lighting, but require adjustment with season.

b. Motion Sensing/Occupancy Sensing

Motion sensors detect the presence of people and respond by automatically turning lights on. The lights are switched OFF again when no occupancy has been detected for a set interval; the interval is chosen to avoid frequent switching, which can shorten lamp life. A minimum ON period of 10 minutes should be set. An alternative strategy is to use a manual switch to switch the lights ON; the occupancy sensor switches the light OFF when no occupancy has been detected for a set time.

Motion sensors mounted in the ceiling can be connected to relays, and several sensors can connect to the same relay. Operations of this control mechanism include reaction by passive infrared (PIR) sensors to changes in heat, such as the pattern created by a moving person.

For larger ranges of operation, a microwave detector may be used, which operates by the Doppler shift of radiation reflected from a moving body. Operating ranges of 50m are possible.

Ultrasonic sensors may also be used in occupancy-sensing and are better suited to see around obstructions besides working effectively in open areas. Ultrasonic sensors transmit sound above the range of human hearing and monitor the time it takes for the sound wave to return. A break in the pattern caused by any motion in the area triggers the control.

Acoustic sensors are available which respond to noise. Occupancy sensors are normally combined with a photocell, which prevents operation of the lighting control when there's sufficient ambient light.

Benefits include a reduction in lighting energy consumption.

c. Daylight Control

This mechanism mostly involves use of daylight-linked automated response systems to facilitate a further reduction in energy consumption.

Daylighting controls use photoelectric sensors that turn lights OFF or dim them when daylight is sufficient. For ON/OFF switching function, they may be used to turn OFF parking lot lights and streetlights during the day. For dimming functions, they reduce the energy used by electric lights in spaces where windows or skylights provide most of the light actually needed in the space and increase light levels at night and on dark days.

Photoswitches are simple devices that offer basic dusk-to-dawn lighting controls. In photoswitches, a photocell throws a switch when ambient light levels are sufficiently low. Photoswitches are most common in streetlights and parking lot lights, but they can also be used to switch indoor lights, especially in daylighted spaces like malls and lobbies.

2.2.3. Remote Control

Remote control gives you the convenience of controlling lighting from wherever you happen to be at the time, like your couch, car or even in your bed. There are several different "methods" of controlling lights remotely.

Remote control of lighting systems can be viewed in a number of ways; control using relay systems, control over the Internet/Ethernet, wireless communication standards e.g. Bluetooth, Wi-Fi etc.

a. Relay Systems

A low-voltage control system can be used to remotely control lighting through relays. Relays are devices that control lighting power by mechanically opening or closing according to signals sent from low-voltage rocker switches, time clocks, or computer-based energy management systems.

In a relay control system, each group of lights that are switched together must be connected to the same relay. Many relays are located together in a panel, usually next to the circuit breaker panel. Relay systems are best for large facilities with big rooms that do not require dimming, such as schools, laboratories, factories, and convention centers.

b. Control over the Internet/Ethernet

The most basic way to look at control of lighting systems over a wireless network e.g. The Internet, is to imagine that a light bulb has a unique IP address that allows it to be controlled by any internet-enabled device to allow for ON/OFF switching, dimming, and light colour control among other control operations.

The advantages of such a network would be significant energy saving, ease of communication from a remote location to facilitate control and alerts of any problems.

c. Powerline Communication

Another wired communication technology, called Powerline Communication, requires no dedicated wire. Command and control functions are sent and received on 110 or 220 V AC wires. Although many variations of Powerline Communications exist, frequency shift keying (FSK) is a popular choice because of the low data rate for command and control functions. In this method, high frequency is used for communication on a 60 Hz power line. The quality of Powerline Communication inevitably varies, and the error correction, such as Forward Error Correction, reduces the error rate and improves the bandwidth of the medium. The data rate can achieve 30 kbps or even higher.

2.2.4. Dimming (Light Level Control)

For continuously occupied spaces which have a substantial proportion of daylighting, light control by switching OFF artificial lighting when daylight is sufficient or switching ON when daylight contribution falls, may cause noticeable and annoying changes in light level.

To avoid this, more effective control can be effected by the use of dimmers. Suitable lamps can be smoothly dimmed down to 10% of their maximum light output before becoming unstable. A ceiling-mounted photocell looking downwards responds to the combined daylight and artificial illumination and the control system is set to provide a constant level of illumination. This system not only takes account of varying daylight, but also the fall-off in light output during the maintenance cycle of the lamp and fitting. The use of an automatic control system will show a saving in energy at the beginning of the maintenance cycle, when lamps are new and the luminaries clean.

Reasons for dimming of light bulbs are many; restaurants dim lights for ambiance, conference rooms have lights dimmers for presentation. So at least for LED designs going into general service lighting the ability to work on a dimmer circuit is highly desirable.

A common light level control technology is based on the Triac dimmer,

Triac Dimmer

The basic waveform generated (output waveform) by a Triac Dimmer is as shown below (Figure 2-2);

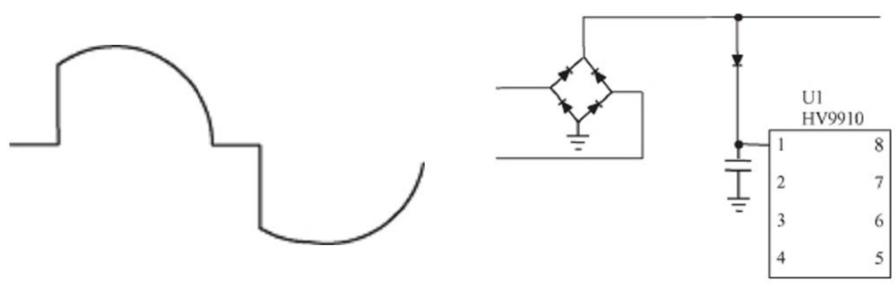


Figure 2.1: Triac Dimmer waveform and schematic

The voltage is zero until some phase angle is reached, and then abruptly resumes the normal sinusoidal voltage for the rest of the cycle. The missing part can either at the beginning of each cycle or at the end, called respectively leading or trailing edge. You see both kinds in commercially available dimmers.

There are two problems with this waveform.

The **first problem** is that when the line voltage abruptly turns on, there is a large surge of current into the input capacitor, which tries to charge up very quickly. This can cause an electrolytic capacitor to blow up within minutes. The solution here is to be PFC

The **second problem** is that LEDs are so efficient that the dimmer may not work! The reason is that dimmers are meant for incandescent light bulbs, the smallest of which is typically a 40W device. Many LED light bulbs, however, are sub-10W. Below about 30W, the triac doesn't work right, you end up with abnormal AC waveforms, and the whole system doesn't dim properly. This problem is not solved by PFC, and indeed is made worse by it, because the current drawn by the supply is lower at the lower line voltages.

3. Design and Implementation

3.1. HIGH-LEVEL DESIGN

The system was conceptualized as follows:

Figure 3.1: High-Level Design

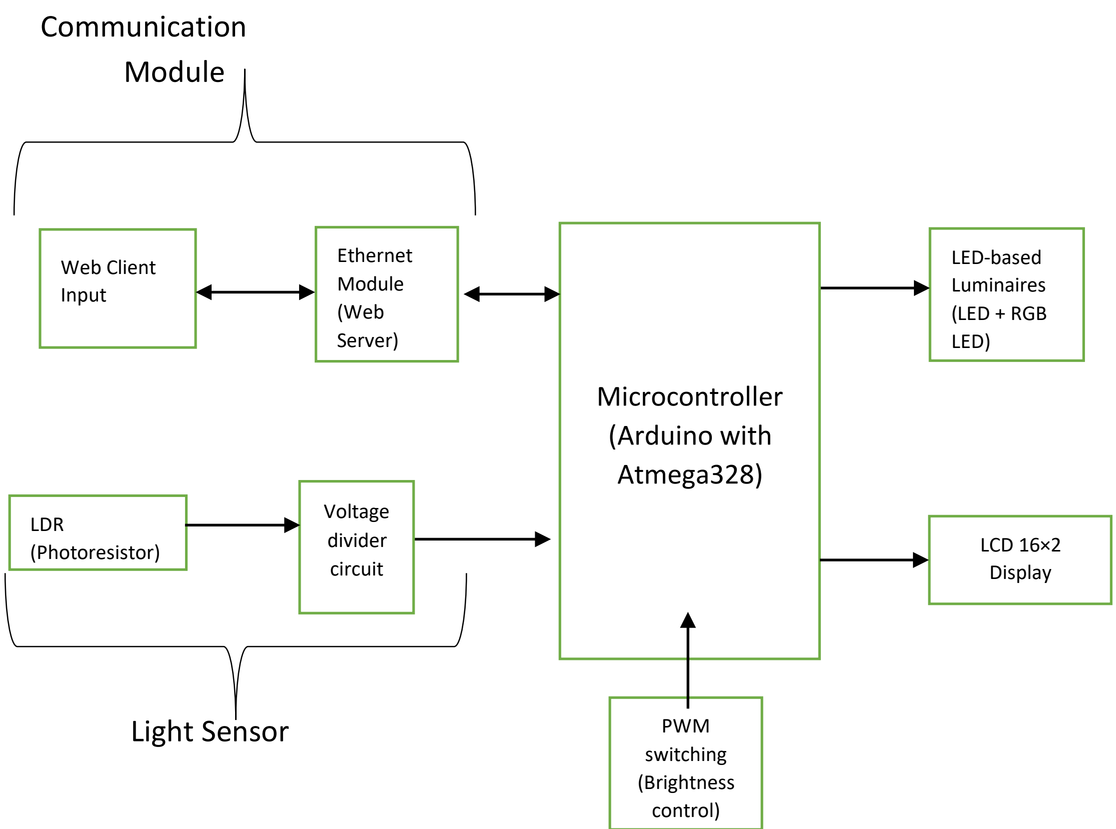


Figure 3.1.: High-Level Design

3.1.1. Description of High-Level Design

The automatic control will be achieved via a photo-sensor incorporated into the prototype. The photo-sensor would sense the level of ambient light and automatically turn the Lighting system ON or OFF depending on the level of ambient luminance.

Remote control is achieved via a web-based application through which the operator can turn the lights ON or OFF. The Ethernet shield will be used for this purpose alongside the Arduino microcontroller.

3.1.2. Performance Requirement

The lighting system will have an automated and remote ON/OFF switching system, option for adaptive adjustment and an intuitive lighting control mechanism.

On-board monitoring as well as a remote feedback mechanism should also be possible.

3.2. SUB-SYSTEM DESIGN

3.2.1. LED System

Basically there exists three types of LEDs:

I. Miniature LEDs
II. High-power LEDs
III. Application-specific LEDs

In choosing the appropriate LED for use, factors that need to be considered are cost, ease of use, suitability to application and the availability of the LEDs.

Miniature LEDs are easy to implement as they come in two or four-pin configurations that are easy to solder or insert into the breadboard slots. They are also more cost-effective since they do not require a heat-sink to dissipate the heat generated during its working. Additionally, advances in LED technology have enabled the development of miniature LEDs with enough luminous intensity and efficiency to be sufficiently used in illumination applications.

Moreover, in this prototype it was required that the light output be diffusional given the objective was adaptive adjustment, and functional lighting colour and levels.

A 4-pin, common-cathode RGB as well as single LEDs for the lighting array was determined suitable for this application.

Resistors to be used for biasing and current limiting were chosen as 10k Ohms and 220 Ohms respectively.

3.2.2. Remote control System

Automated control was achieved via a smartphone application i.e. an optimized browser with which the user will be able to access the webpage and control the lighting system. As the name suggests, the user will be able to control the lighting system from anywhere in the world provided there was a working data service.

Another control means was use of an Android application but this was discouraged due to a number of factors listed in the Challenges section of this documentation.

3.2.3. Automated Control System

This block generally consists of a photoresisitor supplying an analog input to the Arduino which is based on the level of luminous intensity of ambient light. As per the sensitivity level of the LDR in the Arduino sketch, the LDR will be able to control this light automatically so long as the level of light intensity on the LDR surface falls below or above the set-point.

Soon as dark falls the system will automatically turn ON the lights and vice versa.

3.2.4. Adaptive adjustment system

We incorporated a mechanical switch to select a particular colour as preferred by a user. This was due to limited number of PWM pins that would enable digital control of the lighting system. Nonetheless, the objective of the project was still achieved with this modification.

4. Project Methodology

4.1. Function Block Diagram

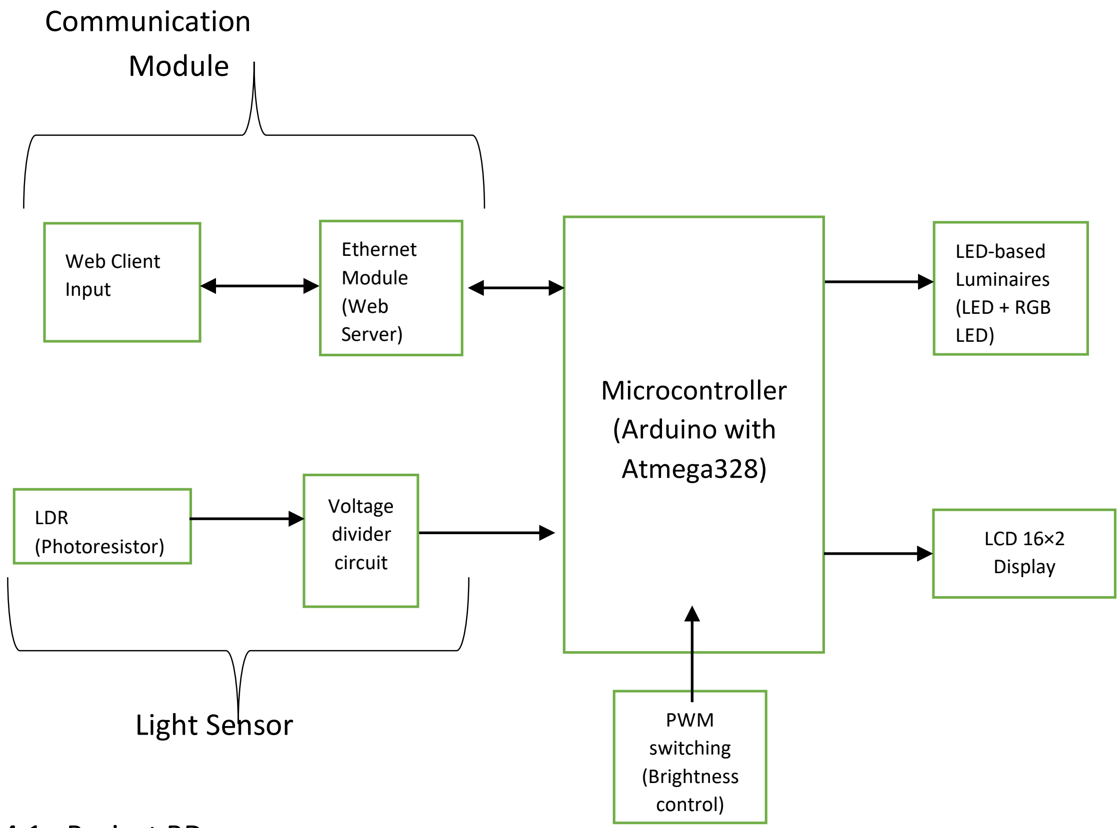
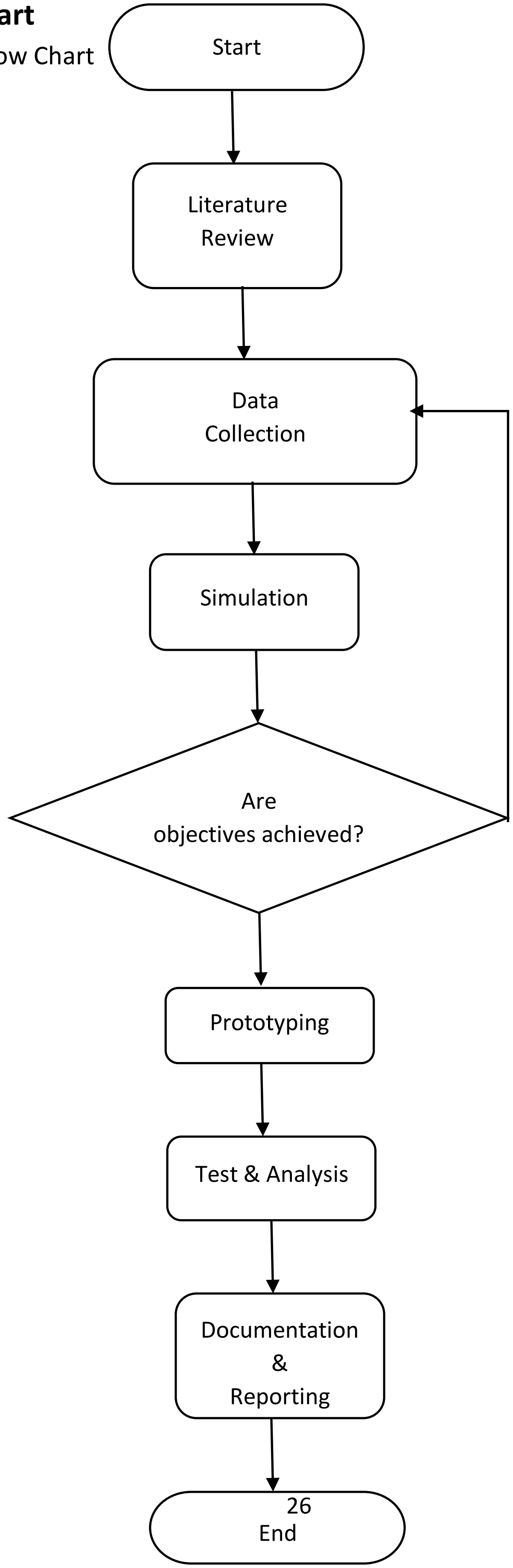


Figure 4.1.: Project BD

4.2. Flow-Chart

Figure 4.2: Project Flow Chart



4.3. Specific Modules Design

4.3.1. Liquid Crystal Display(LCD) Module

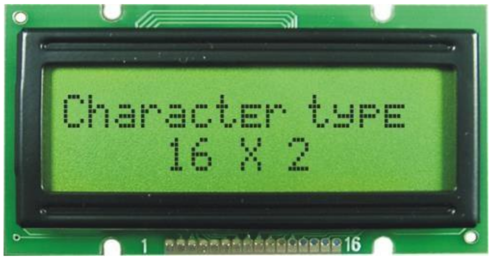


Fig. 4.3: LCD

By adding a liquid crystal display (LCD) to our Arduino, we can more easily display complex information (sensor values, timing information, settings, progress bars, etc.) directly on our Arduino project without having to interface with the serial monitor through the computer using the LiquidCrystal library of the Arduino.

We will use the LCD display to obtain light intensity and light status data report it to us and control our lighting system to compensate for amount of ambient light.

We soldered the header on the LCD so that we could easily install it in our breadboard.

Table 4.1. Pin Configuration of LCD

PIN #	PIN NAME	PIN PURPOSE
1	VSS	Ground connection
2	VDD	+5V connection
3	V0	Contrast adjustment (to potentiometer)
4	RS	Register selection (Character vs. Command)
5	RW	Read/write
6	EN	Enable
7	D0	Data line 0 (unused)
8	D1	Data line 1 (unused)
9	D2	Data line 2 (unused)
10	D3	Data line 3 (unused)
11	D4	Data line 4
12	D5	Data line 5
13	D6	Data line 6
14	D7	Data line 7
15	A	Backlight anode
16	K	Backlight cathode

Table 4.2: LCD-Arduino pin interfacing

LCD PIN	ARDUINO PIN NUMBER
RS	Pin 2
EN	Pin 3
D4	Pin 4
D5	Pin 5
D6	Pin 6
D7	Pin 7

Here's a breakdown of the pin connections:

- The contrast adjustment pin changes how dark the display is. It connects to the center pin of a potentiometer.
- The register selection pin sets the LCD to command or character mode, so it knows how to interpret the next set of data that is transmitted via the data lines. Based on the state of this pin, data sent to the LCD is either interpreted as a command (for example, move the cursor) or characters (for example, the letter a).
- The RW pin is always tied to ground in this implementation, meaning that we are only writing to the display and never reading from it.
- The EN pin is used to tell the LCD when data is ready.
- Data pins 4–7 are used for actually transmitting data, and data pins 0–3 are left unconnected.

4.3.1.1. Adding Text to the Display

1. Include the LiquidCrystal library
2. Initialize the LCD object
3. Specify display characteristics i.e. rows and columns
4. Call print() and setCursor() functions to print text to a given location on the display.

For on-board monitoring, the LCD display will be tuned (using an *if….else* statement in the Arduino sketch) to indicate light status at each time i.e which LED is ON, etc.

We will configure the LCD as described above to modify the sketch above to be able to print light status in real time. Given that the string display on the LCD will be limited to 16X2

Moreover, for remote status feedback, the labels in the Webserver will be tuned to change colour for a particular status of each LED.

4.3.2. Ethernet Shield/Module

This is used as a communication module between the webserver and the Arduino through to our light system.

The Arduino Ethernet is a microcontroller based on the Atmega34. It has 14 digital input/output pins, 6 analog inputs, a 16MHz crystal oscillator a RJ45 connection, a power jack an ICSP header and a RESET button.

Figure.4.4. Ethernet Shield

10, 11, 12 and 13 are reserved for interfacing with the Ethernet module and should not be used otherwise. Thus only 9 PWM pins available for use.

We chose Ethernet module over other boards since it instead of a rather slower USB-to-serial driver chip, it uses a Wiznet Ethernet shield. This is the same interface found on the Ethernet shield.

An on-board micro SD card reader, which can be used to store files for serving over the network, is accessible through the SD Library. Pin 10 is reserved for the Wiznet interface, SS for the SD card is on Pin 4.

4.2.2.1. Power

The board can also be powered via an external power supply, an optional Power over Ethernet (PoE) module, or by using a FTDI cable/USB Serial connector. External power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a

2.1mm centre-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. The power pins are as follows:

VIN.-The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V-The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

3V3-A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50mA.

GND-Ground pins.

The Power Over Ethernet (PoE) module is designed to extract power from a conventional twisted pair Category 5 Ethernet cable. Moreover, we preferred it over other methods due to the following merits:

- IEEE802.3af compliant
- Low output ripple and noise
- Input voltage range 36V to 57V
- Overload and short-circuit protection
- 9V output
- High-efficiency DC-DC conversion
- 1500V isolation(input to output)

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.

4.2.2.2. Input and Output

Each of the 14 digital pins on the Ethernet board can be used as an input or output, using pinMode(), digitalWrite(), and 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 20-50k Ohms

The Ethernet board has 6 analog inputs, labelled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function

Arduino-Ethernet pin mapping has been included in the Appendix.

4.2.2.3. Communication

The Arduino Ethernet has a number of facilities for communicating with a computer, another

Arduino, or other microcontrollers.

A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports TWI and SPI communication. The Arduino software includes a Wire library to simplify use of the TWI bus. For SPI communication, use the SPI library.

The board also can connect to a wired network via Ethernet. When connecting to a network, you will need to provide an IP address and a MAC address. The Ethernet Library is fully supported.

The on-board microSD card reader is accessible through the SD Library. When working with this library, SS is on Pin 4.

4.2.2.4. Programming

It is possible to program the Arduino Ethernet board in two ways: through the 6 pin serial programming header, or with an external ISP programmer.

The 6-pin serial programming header is compatible with FTDI USB cables and the Sparkfun and Adafruit FTDI-style basic USB-to-serial breakout boards including the Arduino USBSerial connector. It features support for automatic reset, allowing sketches to be uploaded without pressing the reset button on the board. When plugged into a FTDI-style USB adapter, the Arduino Ethernet is powered off the adapter.

4.2.2.5. Features:

i. IEEE802.3af compliant Small SIL package size - 56mm (h) x 14mm (h)
ii. Low output ripple and noise Input voltage range 36V to 57V
iii. Overload and short-circuit protection iv. High efficiency DC/DC converter (75% typ)
v. 1500V isolation (input to output).

This module could be used with products Arduino Ethernet shield and Arduino Ethernet

4.2.3. Arduino Uno Microcontroller

Figure. 4.5.: Arduino Uno Components

The microcontroller of choice in our project is an Arduino Atmega328 attached by Ethernet shield. The Arduino Uno will be used to monitor and control our light system. It's been programmed to read data from LDR and Webserver and use this data to determine the light status and in case of adaptive adjustment, vary its brightness as per the user preference as well.

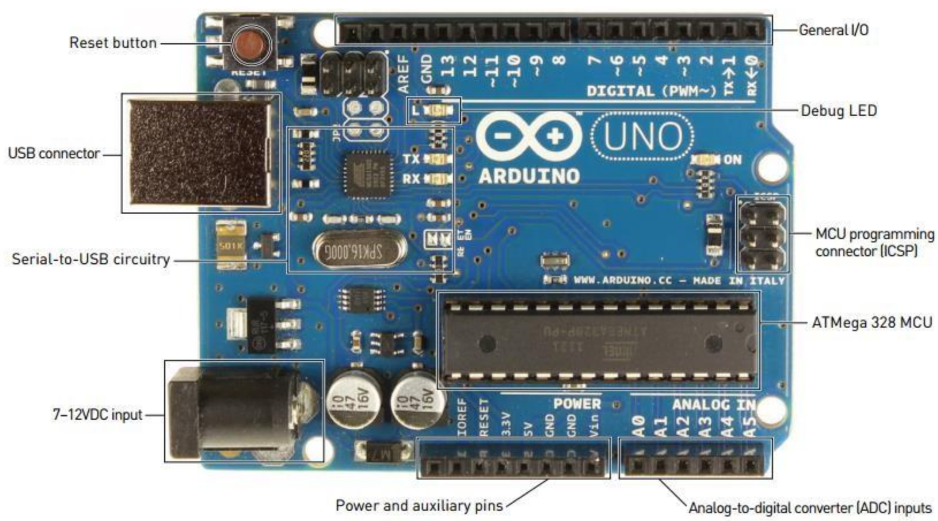


Figure 1-1: Arduino Uno components

Credit: Arduino, www.arduino.cc

Also, through the LCD display, the Arduino Uno should be able to indicate the real-time status of the lighting system to the user. The programming also allows the Arduino to interface command and monitoring of the light status via the Webserver.

Table 4.3. Arduino Atmega328 specifications

Microcontroller	**ATmega328**
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA

Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

The following are an intended aspects of control of the Arduino Uno microcontroller:

1. Brightness control using Pulse-Width Modulation
2. Remote light switching via web
3. Light status display on the LCD
4. Automatic switching of smart lights via LDR/photodetector

4.2.4. Light Dependent Resistors

A photoresisitor/photo-cell is a light-controlled variable resistor. Its resistance changes based on luminance.

It has two Cadmium Sulphide photoconductive cells with spectral responses similar to that of the human eye. The light falling on the zigzag lines on the sensor (usually made of Cadmium Sulphide), causes the resistance of the device to fall. This is known as a negative-coefficient LDR.

If LDR resistance increases with light intensity, then it's called a positive-coefficient LDR. Other types of optical sensors include: photo-emitters, photovoltaics and photodiodes, optical fibre and sensors with negligible sensor dynamics.

We opted for the photo-resistors because of its relative affordability, compactness, easy handling and ruggedness.

The voltage of the LDR exposed to light is measured and inputs to the SELECT Arduino pin in our project. The whole system acts as a potential divider and the voltage across the LDR is proportional to the current through it.

Light dependent resistors have a particular property in which they tend to 'remember' the lighting conditions in which they have been stored. This memory effect can be minimised by storing the LDRs in light prior to use. Light storage reduces equilibrium time to reach steady resistance values.

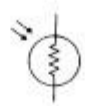
Fig. 4.6.: LDR Specifications

Two cadmium sulphide (cdS) photoconductive cells with spectral responses similar to that of the human eye. The cell resistance falls with increasing light intensity. Applications include smoke detection, automatic lighting control, batch counting and burglar alarm systems.

Guide to source illuminations

Light source	Illumination (Lux)
Moonlight	0.1
60W bulb at 1m	50
1W MES bulb at 0.1m	100
Fluorescent lighting	500
Bright sunlight	30,000

Circuit symbol



Light memory characteristics

Light dependent resistors have a particular property in that they remember the lighting conditions in which they have been stored. This memory effect can be minimised by storing the LDRs in light prior to use. Light storage reduces equilibrium time to reach steady resistance values.

NORP12 (RS stock no. 651-507)

Absolute maximum ratings

Voltage, ac or dc peak	320V
Current	75mA
Power dissipation at 30°C	250mW
Operating temperature range	-60°C to +75°C

Electrical characteristics

T_A = 25°C. 2854°K tungsten light source

Parameter	Conditions	Min.	Typ.	Max.	Units
Cell resistance	1000 lux	-	400	-	Ω
	10 lux	-	9	-	kΩ
Dark resistance	-	1.0	-	-	MΩ
Dark capacitance	-	-	3.5	-	pF
Rise time 1	1000 lux	-	2.8	-	ms
	10 lux	-	18	-	ms
Fall time 2	1000 lux	-	48	-	ms
	10 lux	-	120	-	ms

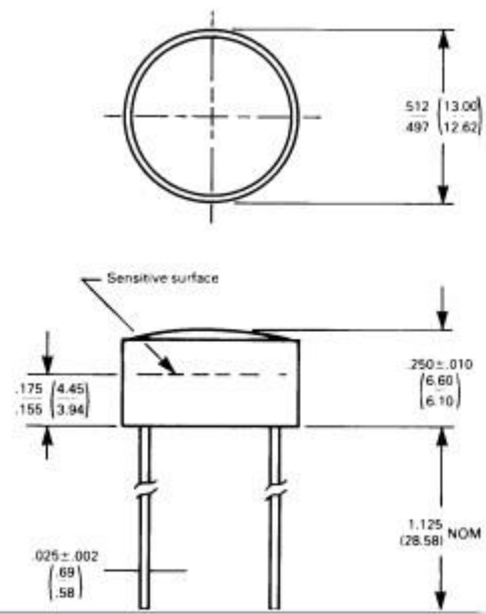
1. Dark to 110% R_L
2. To 10 × R_L

R_L = photocell resistance under given illumination.

Features

- Wide spectral response
- Low cost
- Wide ambient temperature range.

Dimensions



How to use a Light-dependent Resistor (LDR)

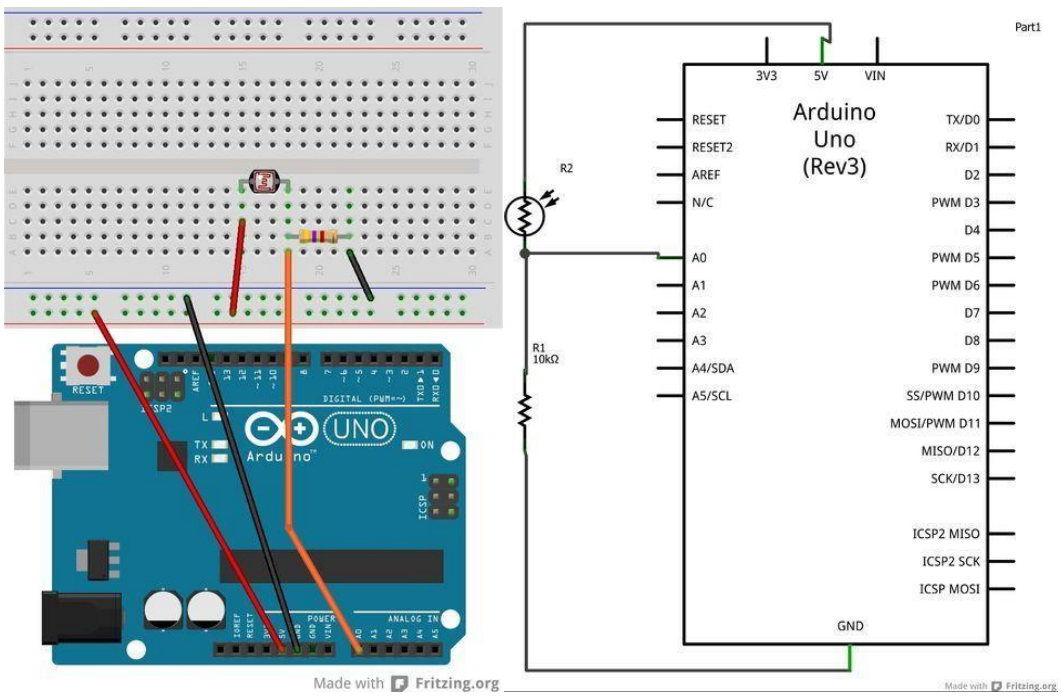
How to use LDR to sense light levels with the Arduino and print the measurements to the serial port.

Arduino measures voltages on pins A0-A5, but the LDR has a value of resistance that varies with level of luminance. We thus need to convert this value of resistance to a voltage that the Arduino can measure.

This can be achieved by using the LDR and a resistor in a potential divider circuit.

Working:

Figure. 4.7.: LDR Arduino Simulation



The top of the potential divider is 5V, the bottom is at 0V and the middle(connected to A0) is some value between 5V and 0V that varies as the LDR resistance varies. Since, LDR resistance varies with luminance occuring on the LDR surface, the valotage thus varies with level of luminance, albeit indirectly.

For precision and technicality purposes, the voltage at pin A0 will be given by:

$$Vao = 5 * \frac{R1}{R1 + R2}$$

Va0 – Voltage at A0 pin

R2- top resistor value

R1- bottom resistor value

In the code [See Appendix], the light sensitivity of the LDR has been set to 500 and if luminance on the LDR falls below this level, the LDR will turn on and vice versa. The level of ambient light to turn the LEDs ON or OFF can be so appropriately adjusted.

5. Results and Analysis

The average latency between actuating an output response to the actual response was found to be dependent on the means of accessing the webpage. For smartphones, it was found to be between 0-3s while for a PC it was almost instantaneous.

This could be attributed to website optimization problems i.e. had not been well optimized for faster, seamless control. Particularly, a mobile phone was somehow slow in terms of =sending a signal over Wi-Fi. Nonetheless, this delay was almost negligible.

At the beginning, we used a LAN cable to communicate between the Ethernet shield and the wireless router. In crossing over the signal from Wi-Fi to LAN channel, we discovered that this had some significant implications on real-time control and feedback. We replaced the LAN cable with a high-speed cross-over cable with which we were able to realise the speedy signal conveyance and response.

5.1. Communication Results

According to the Arduino web server sketch (Appendix), an Ethernet is first configured to establish connection of Arduino in the network. Essentially, it involves placing the client (PC or Phone) and the server (Arduino with Ethernet Shield) on the same subnet.

Two ways can be used to confirm this connectivity: through the serial monitor and through the LCD display.

The IP address is then entered in a browser to access controls web page as shown in Figure 4-2;

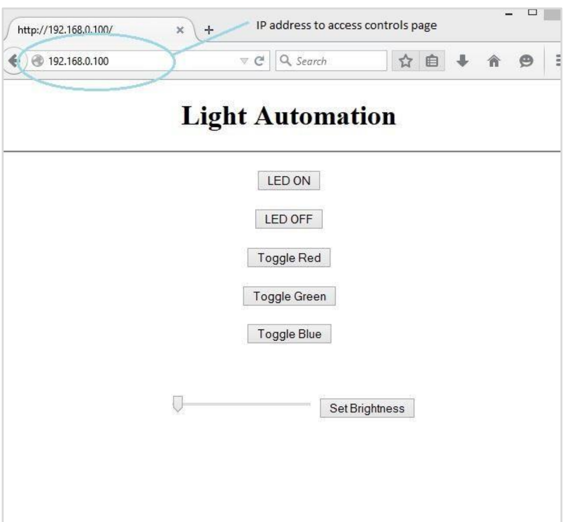


Figure 5.1: IP Address to Access Control Page

The LCD display also gives a response to indicate that a connection has been established. It does this by displaying **"LED Automation"** once the system is powered. This is shown on the figure below;

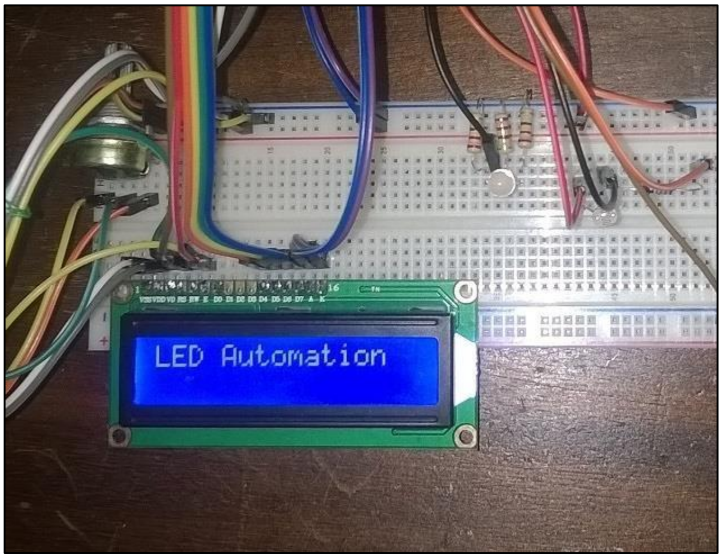


Figure 5.2: LED Automation

5.2. ON/OFF Switching

From the control web page (Figure 4-2), the first two buttons allow for ON/OFF LED switching. Upon clicking either of the buttons, the appropriate effect is seen with LED either going ON or OFF.

Besides this visual feedback, both the serial monitor and LCD give an indication of this effect as shown in figures (Figure 4-4 and Figure 4-5) below;

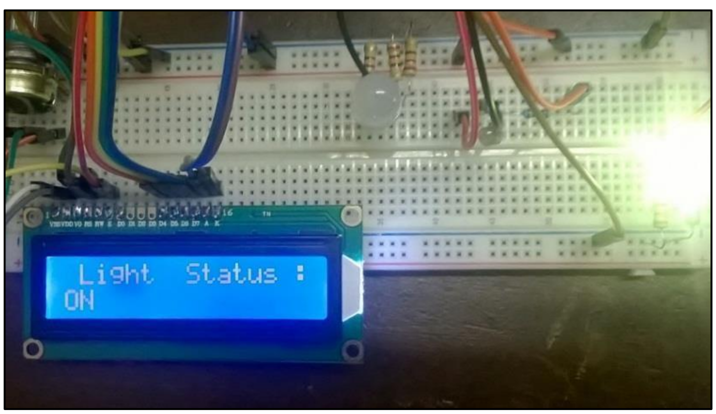


Figure 5.3: Light Status ON

5.3. Brightness Control (Dimming)

Adjusting the brightness is achieved by sliding the slider input on the web page to set brightness. The range slider input is moved in increments of 51 for brightness byte range of between 0 and 255 to select a brightness value. This is so because we use analogWrite() command on the Arduino IDE to set brightness of an LED. This command is an 8-bit value and only accepts values between 0 and 255.
This means that there are five input steps to select brightness as depicted in the table below;

Table 5.1.: Brightness Level Set points

Brightness Bytes (0 to 255)	**Brightness Percentage**
0 - 51	≤ 20 %
52 - 102	≤ 40 %
103 - 153	≤ 60 %
154 - 204	≤ 80 %
205 - 255	≤ 100 %

Upon setting the brightness from range slider on the web page, effect is observed on the LED whose light intensity changes accordingly.

The LCD also shows the change in brightness by using a progress bar to show level of brightness. By using a progress bar, this implies that the operator needs not to visually see the LED to acknowledge the brightness intensity change. Therefore, suiting this control techniques to remote control applications.

The images below (Figure 4-6) shows two sets of brightness levels with their respective LED and LCD outputs.

Brightness at 20%	Brightness at 80%

Figure 5.4: Brightness Level Output Comparison

5.4. LDR (Light Sensor)

Automatic light control is based on a LDR (photocell) that is part of a voltage divider circuit. This voltage divider circuit gives a voltage output (shown below) that corresponds to the minimum and maximum brightness.

$$Vout = Vin(\frac{R2}{R1 + R2})$$

Where either R1 or R2 is a LDR (light-dependent variable resistance and the other a fixed resistor (that can be changed to change sensor sensitivity).

In this project a 200kΩ (complete darkness) photoresistor is used whereas a 10k Ω fixed resistor is chosen to form the voltage divider analog input. However, an analysis can be made to determine the suitability of these values with regard to sensor sensitivity required plus the suitable voltage divider configuration.

a. Sensor Grounded via a resistor

Figure 5.5: Sensor Grounded via Resistor

When R is varied using 10Ω, 1k, 10k, 100k and 1M and corresponding Vo is measured, the results in the table below were obtained.

Table 5.2: Resistor Selection (Sensor Grounded via Resistor)

	Vo (Light)	**Vo (Dark)**	**ΔV**
10Ω	0.833	0.000496	0.832504
1kΩ	3.333	0.00495	3.32805
10kΩ	4.762	0.04911	4.71289
100kΩ	4.975	0.451	4.524
1MΩ	4.997	2.48	2.517

b. Sensor Grounded Directly

When R is varied using 10Ω, 1k, 10k, 100k and 1M and corresponding Vo is measured, the results in the table below (Table 4-3) were obtained.

Table 5.3: Resistor Selection (Sensor Grounded Directly)

	Vo (Light)	**Vo (Dark)**	**ΔV**
10Ω	4.902	4.999	-0.097
1kΩ	1.667	4.995	-3.328
10kΩ	0.238	4.949	-4.711
100kΩ	0.0248	0.4538	-4.5132
1MΩ	0.00247	2.48	-2.47753

From tables above, it can be seen that using a 10kΩ resistor gives the largest change in Vo. The sensor is therefore most sensitive to changes in light level when used with a 10kΩ resistor than any other resistor.

Setting LDR sensitivity

As mentioned earlier, the light sensor is wired as a voltage divider giving an analog voltage input. This analog input has to be converted into a digital value by means of an ADC. The ADC therefore returns values of between 0 and 1023 bits.

Light sensitivity is first determined by choosing optimum values of the voltage divider elements. For this project, a resistance of 200kΩ (complete darkness) is chosen for R1 (photocell) and 10kΩ for R2.

Using the following Arduino sketch (Appendix), minimum and maximum brightness values are determined in terms of values between 0 and 1023 bits.

The maximum brightness is obtained by exposing the LDR to a very intense light source (For this project, I used an LED flashlight).

The minimum brightness is obtained by blocking the LDR by use of the thumb to avoid light access.

These values are displayed on the serial monitor are displayed on the serial monitor as shown on the figure below (Figure 4-9);

Minimum Brightness

Maximum Brightness

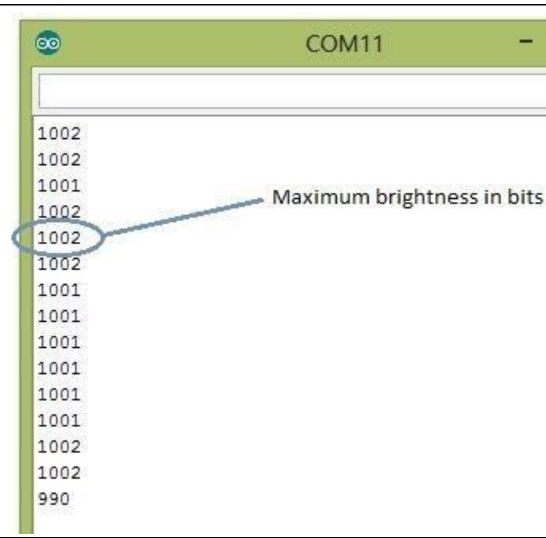


Figure 5.5: Serial Output of Brightness Level

As shown above the light intensity values ranges from 644 to 1002. Ideally, this range should be from 0 to 1023 bits. However, this full range can never be realized because the variable resistor will never have a resistance of zero.

The range obtained (644 – 1002) is used to set the light sensor sensitivity to be in the Arduino sketch (Appendix) under "**MIN_LIGHT**" and "**MAX_LIGHT**". These values can also be used to determine other sensitivity points as shown in the table below (Table 4-4); Table 5.4: Light Sensor Sensitivity Selection

	Least Sensitivity	**Optimal Sensitivity**	**Most Sensitivity**
MIN_LIGHT	600	600	700
MAX_LIGHT	900	800	850

5.5. Light Colour

A common-anode RGB LED is used in this project to effect colour change. The colour change is effected using three sets of resistors for Red, Green and Blue colours. Varying the proportions of R, G and B preferably using separate variable resistors produces any colour imaginable.

If this effect (varying resistor values) is combined with Arduino PWM outputs to control the brightness of each colour, a more functional colour change is affected.

5.6. 4.6 Project Cost Analysis

Table 5.5: Project Cost Analysis

	Item	**Unit**	**Unit Cost (Ksh.)**	**Total (Ksh.)**
1	Arduino Microcontroller	1	2000	2000
2	Ethernet Shield	1	1200	1200
3	16×2 LCD	1	800	800
4	Prototyping Breadboard	1	400	400
5	Jumper Wires Pack	1	350	350
6	Resistors	6	5	30
7	LDR	1	20	20
8	RGB LED	2	30	60
9	Potentiometer	1	50	50
TOTAL				4910

6. Miscellaneous

6.1. RECOMMENDATIONS

Having observed the performance, achievements and shortcomings of our implementation, we made the following observations:

- Use of an Arduino board with more PWM pins would widen the scope of a prototype and number of items being controlled.
- Automatic smart lights improves lighting system efficiency since it is able to sense when light is required or especially when the user is forgetful or is away.
- Remote lighting improves lighting system management since it enables the user to operate the lights from where they are as long as they have an internet-enabled phone.

6.2. CONCLUSION

The literature review on current lighting control technologies, attitude of users towards automation and the process of fabrication revealed that the field of automatic lighting control had little been explored and much still needed to be done particularly in the Kenyan scenario.

In most households, lighting accounts for a major art of power usage and a technology improvement on lighting system, remote control and automation would majorly affect people's livelihood in many ways.

The results of the project proved that the RGB LED system really did improve the mood of the occupants in that particular room. It also proved to help in playing with the human psychology so that the rooms appeared smaller or larger in different colour lightings and also in making a room feel cooler or warmer. This system is applicable to all genders and ages.

6.3. CHALLENGES ENCOUNTERED

1. We ran out of PWM output pins on the Arduino board. As such, we were not able to implement the RGB adaptive adjustment aspect. We were forced to connect three LEDs

in parallel and incorporate a mechanical switch to select as particular lighting colour and proceed to control the same remotely via a smartphone web app.

2. As mentioned earlier, there was a real-time control problem at the initial stages of implementation because of the need to switch from LAN to wireless communication during operation. This was however taken care of by use of a cross-over cable.

3. We had intended to also use a common-anode RGB LED instead of three separate, different-colour LEDs for the adaptive adjustment block of our project. However, on arrival of our order, the supplier had packed common-cathode LEDs instead. It was not possible to control the individual colours. Moreover, assuming the LEDs were commoncathode, we reverse-biased the first RGB LED, causing it to blow.
4. It was impossible to model and simulate some blocks of the project e.g. the Ethernet part since its library was not available on any simulation software.

7. Appendix

7.1. Appendix I: Implementation Drawings and Source Code

Coding

The following Arduino sketch was programmed into the Arduino to perform the above specified objectives. It’s a snippet from the Proteus Simulator software and the Arduino IDE.

7.2. Pulse-Width Modulation

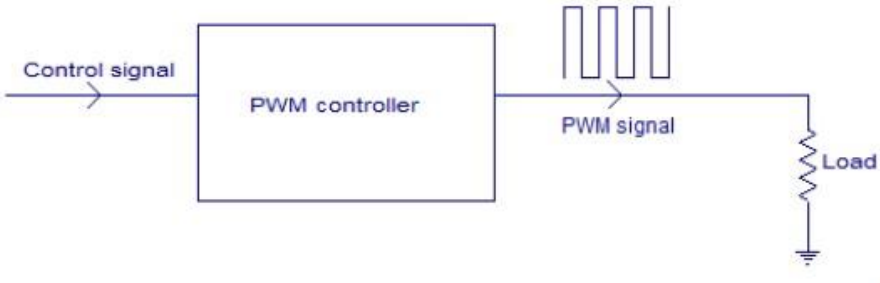
Pulse-width modulation is a commonly-used technique of controlling power across loads. This method is very easy to implement and has high efficiency. PWM signal is essentially a highfrequency square wave (typically greater than 1 kHz). The duty-cycle of this square wave is varied in order to vary power supplied to the load. Duty cycle is usually stated as a percentage and can be calculated using the equation:

$$\%Duty\ Cycle = \left(\frac{Ton}{Ton = Toff}\right) * 100$$

Where:

 T_{ON} is duration during which the square wave is high
T_{OFF} is the time for which the square wave is low.

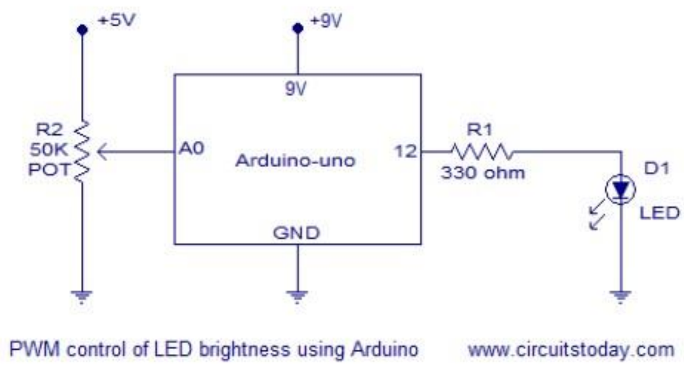
The block diagram for the PWM controller is shown below:



Control signal is what is given to the PWM controller as input-could be analog or digital depending on design of PWM controller. The control sgnal contains information on how much powewr has to be applied to the load. The PWM uses this input to control the duty cycle of the PWM signal as per requirements.

LED Brightness control using Arduino

Figure. 7.2.: LED Brightness Control using Arduino



The slider of the 50k Ohm potentiometer is connected to analog input A0 of the Arduino. The LED is connected to digital pin 12 in series with a current-limiting resistor.

Arduino reads the voltage at pin A0. Necessary calculations are calculated using this value and the duty cycle is adjusted according to it.

A step-wise working of the code is outlined in the Code section of the Appendix.

7.3. APPENDIX II: SIMULATIONS

During the prototyping of our project we made several simulation for model testing purposes of the various blocks of our project using the Proteus and Fritzing simulation software. The snippets shown below were the results:

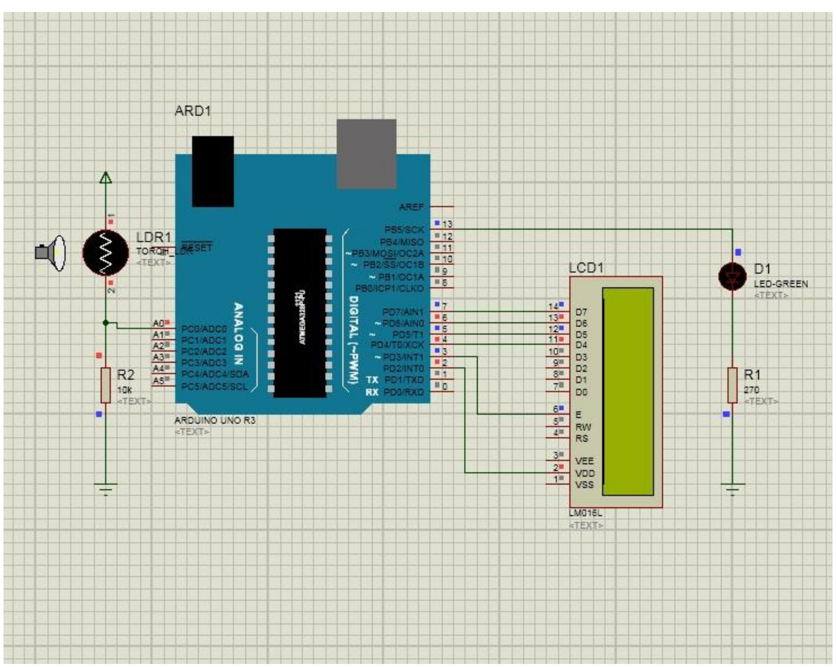


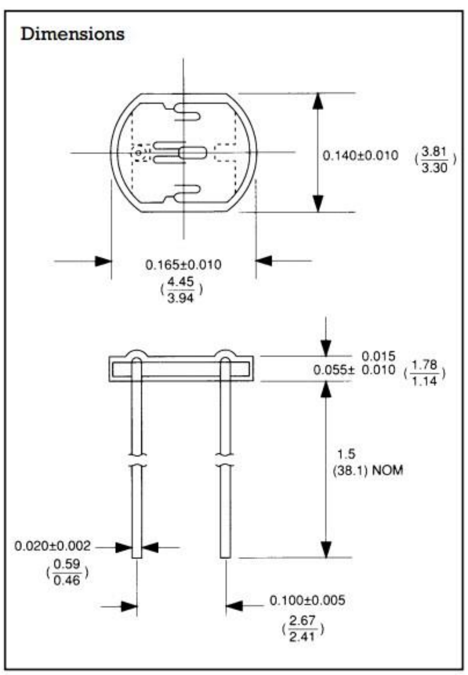
Figure 7.3: LCD and LDR simulation using Proteus software

The Ethernet module could not be simulated because its library was not available in any of the simulation software available to us.

7.4. APPENDIX III: DATASHEETS

7.4.1. *Light-Dependent Resistor Characteristics*

Figure. 7.4.: LDR Characteristics



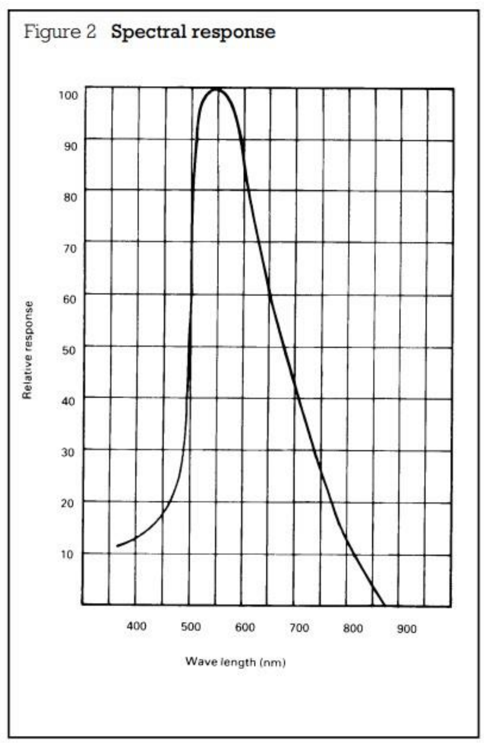
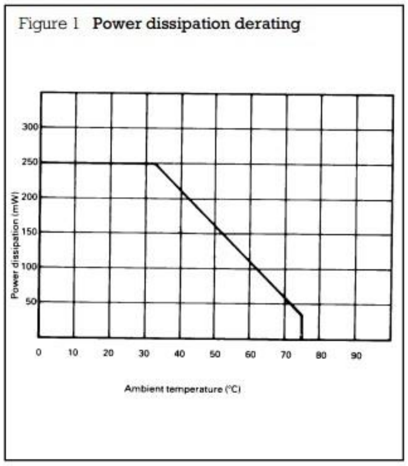
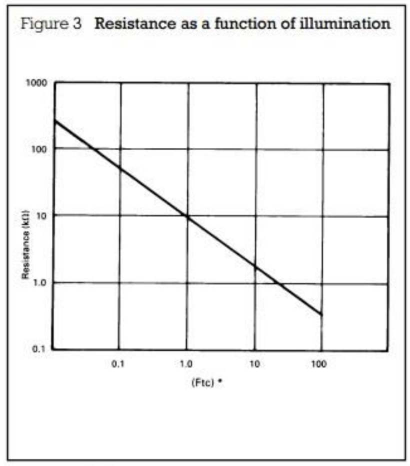


Figure. 7.5.: LDR Curves





*1Ftc=10.764 lumens

7.4.2. Arduino Uno

Fig.7.6. Arduino Uno

Table 7.1.: Summary of Arduino Uno

Microcontroller	**ATmega AT368**
Operating Voltage	**5V**
Input Voltage(Recommended)	**7-12V**
Input voltage(limits)	**6-20V**
Digital I/O pins	**22(8 provide PWM)**
Analog input pins	**8**
DC current per I/O pin	**40mA**

Flash memory	**256KB of which 8KB is used by bootloader**
SRAM	**8KB**
EPROM	**4KB**
Clock Speed	**16MHz**

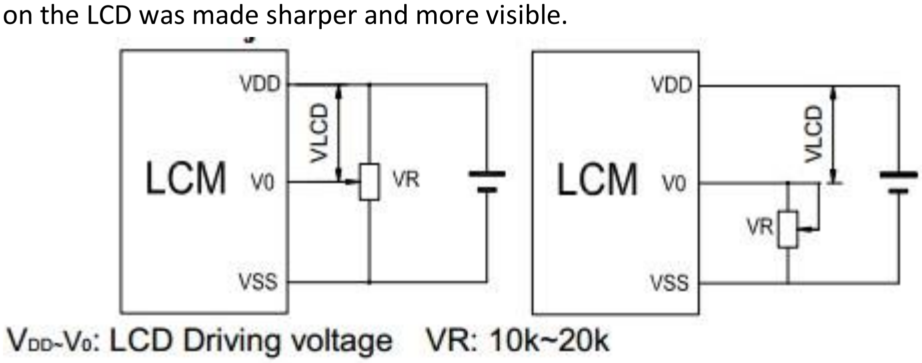
7.4.3. Liquid Crystal Display

Table.7.2. LCD Features and Characteristics

Pin no.	Symbol	External connection	Function
1	Vss	Power supply	Signal ground for LCM
2	V_{DD}		Power supply for logic for LCM
3	V_0		Contrast adjust
4	RS	MPU	Register select signal
5	R/W	MPU	Read/write select signal
6	E	MPU	Operation (data read/write) enable signal
7~10	DB0~DB3	MPU	Four low order bi-directional three-state data bus lines. Used for data transfer between the MPU and the LCM. These four are not used during 4-bit operation.
11~14	DB4~DB7	MPU	Four high order bi-directional three-state data bus lines. Used for data transfer between the MPU
15	LED+	LED BKL power supply	Power supply for BKL
16	LED-		Power supply for BKL

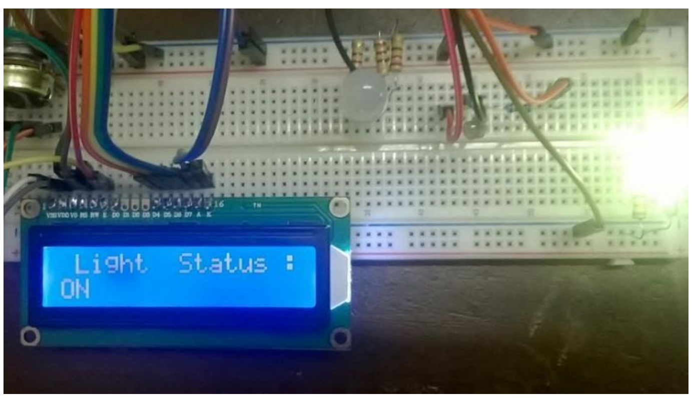
Figure.7.7: Contrast Adjustment

Contrast adjustment was done using a potentiometer. By adjusting the value of its resistance, the display on the LCD was made sharper and more visible.



7.5. APPENDIX IV: PROJECT SNAPSHOTS

Figure 7.10: Project snapshots



This is how the webpage looks like on a smartphone/PC:

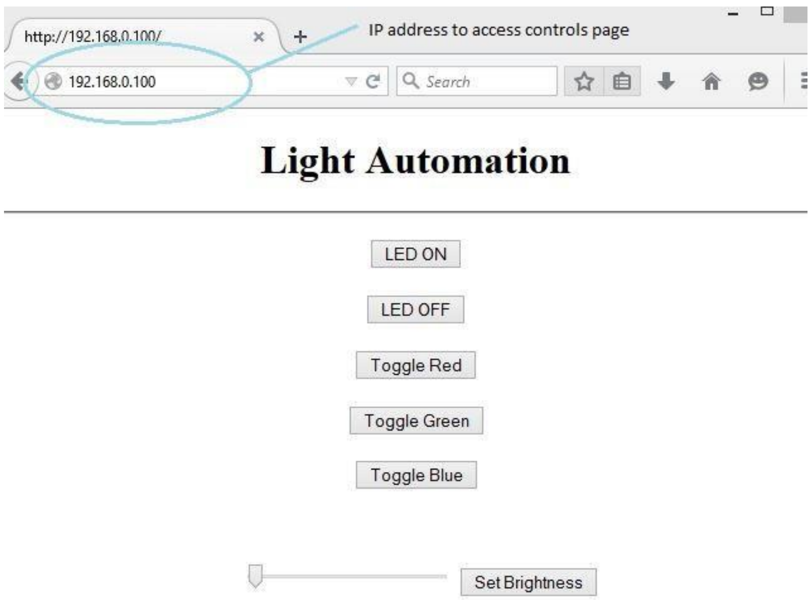
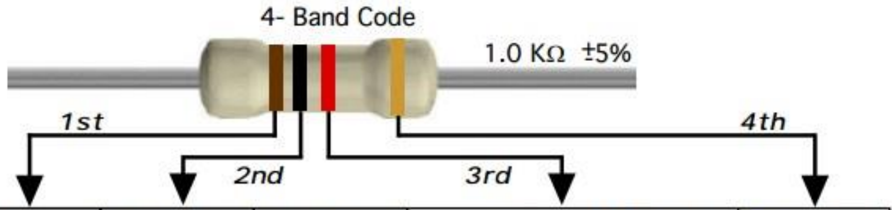


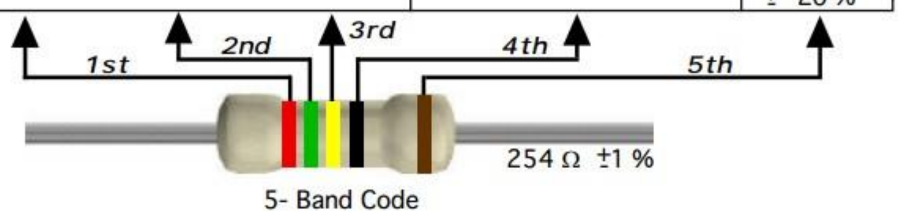
Figure 7.11: User Control Interface

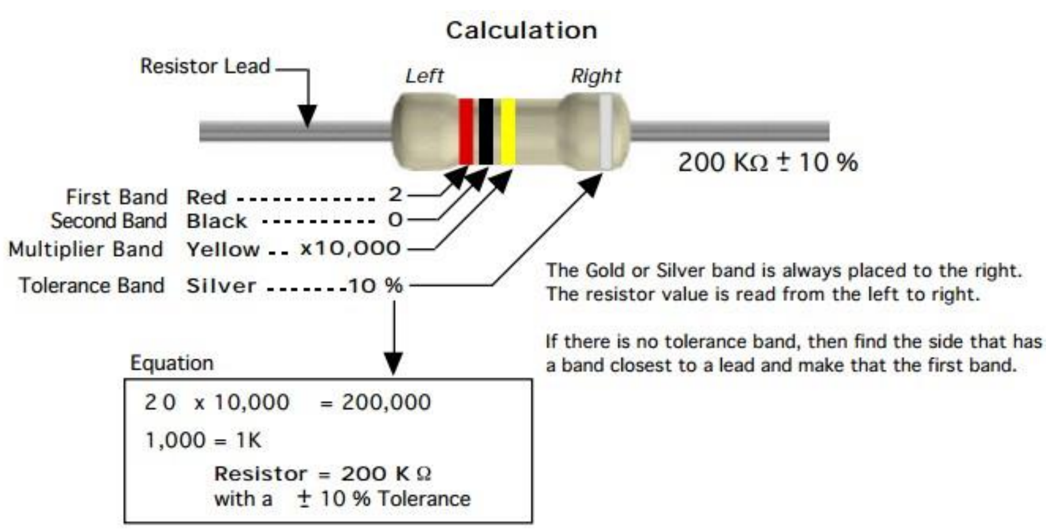
7.6. How to Read the Resistor Code

RESISTOR COLOR CODE GUIDE



Color	1st Band	2nd Band	3rd Band	Decimal Multiplier		Tolerance
Black	0	0	0	1	1	
Brown	1	1	1	10	10	± 1 %
Red	2	2	2	100	100	± 2 %
Orange	3	3	3	1K	1,000	
Yellow	4	4	4	10K	10,000	
Green	5	5	5	100K	100,000	
Blue	6	6	6	1M	1,000,000	
Violet	7	7	7	10M	10,000,000	
Gray	8	8	8		100,000,000	
White	9	9	9		1,000,000,000	
Gold					0.1	± 5 %
Silver					0.01	± 10 %
None						± 20 %





8. REFERENCES

[1]. Blum, Jeremy. *Exploring Arduino-Tools and Techniques for the Engineering Wizardy* (1998, May 19). *USA Patent No. 5752766 A.*

[2]. Booch, G., Rumbaugh, J., & Jacobson, I. (1998). *The Unified Modelling Language User Guide.* Addison Wesley.

[3]. Dorf and Bishop, Addison-Wesley Modern Control Systems

[4]. Mueller, Jameson. *Light Dependent Resistor Data-sheet(LDR_NSL19_M51.pdf)*

[5]. P.Lalanda, & S.Cherki. (1998). Object oriented methods and software architectures. *Position Paper in ECOOP 98, Workshop on Object Oriented Software Architecture.*

[6]. Qu, X., Wong, S. C., & Tse, C. K. (2007). Colour Control System for RGB LED Light Sources Using Junction Temperature Measurement. *The 33rd Annual Conference of the IEEE Industrial Electronics Society.* Taipei.

[7]. Shaikh, P. H., Nor, N. B., Nallagownden, P., & Irraivan. (2013). Building Energy Management through a Distributed Fuzzy Inference System. *International Journal of Engineering and Technology*, 3236-3242.

[8]. The Society of Light and Lighting. (2002). *Code for Lighting.* London: Butterworth Heinemann.

[9]. Umare, M. S., Jain, P. A., & Kushare, P. D. (n.d.). Development of computer controlled colour mixing illumination network using RGB LEDs. *IOSR Journal of Engineering (IOSRJEN)*, 44-47.

[10]. Internet Sources:

- www.electrical4u.com/photoresistors/
- www.sunroof.com/light-detection-devices/
- www.lightingcontrols.com