SMART LED LAMP

A

MINOR PROJECT REPORT

Submitted by

Gagan Chawla (12614802714)

Naveen Bhardwaj (12714802714)

Nikhil Kumar Sharma (11814802714)

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

Under the Guidance

of

Ms. Savita Sharma

Department of Computer Science and Engineering

Maharaja Agrasen Institute of Technology,

PSP area, Sector – 22, Rohini, New Delhi – 110085

(Affiliated to Guru Gobind Singh Indraprastha,

New Delhi)

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering

CERTIFICATE

This is to Certified that this minor project report of "Smart LED Lamp" is submitted by Naveen Bhardwaj (12714802714), Gagan Chawla (12614802714), Nikhil Kumar Sharma (11814802714) who carried out the project work under my supervision.
I approve this minor project for submission.

Dr. Namita Gupta
(HOD, CSE)

Ms. Savita Sharma
(Project Guide)

ACKNOWLEDGEMENT

It gives us immense pleasure to express our deepest sense of gratitude and sincere thanks to our highly respected and esteemed guide Ms. Savita Sharma, CSE Department MAIT Delhi, for their valuable guidance, encouragement and help for completing this work. Their useful suggestions for this whole work and co-operative behaviour are sincerely acknowledged.

We also wish to express our indebtedness to our parents as well as our family member whose blessings and support always helped us to face the challenges ahead.

Place: Delhi

Date:

Naveen Bhardwaj(12714802714)

Gagan Chawla(12614802714)

Nikhil K. Sharma(11814802714)

ABSTRACT

The project aims to develop a smart LED lamp, which is remotely controlled by Android apps via handheld devices, e.g., smartphones, tablets, and so forth. A color slider will be shown on the app from which the user can select a color combination to be glowed by LED Bulb. The development of this project requires an Arduino UNO micro-controller, ESP8266 module and RGB LEDs.

Table of Content

1. Introduction

Light-emitting diodes (LEDs) have numerous advantages over conventional lamps or compact fluorescent lamps (CFLs), such as long lifetime, large color gamut, small size, and absence of mercury vapour. Lamps based on high brightness red, green and blue (RGB) light-emitting diodes (LEDs) can produce light of nearly any color. Since LEDs have a short rise and fall times, such array of RGB and white LEDs can be controlled electronically by means of varying the duty cycle of ON time to generate a particular color. Thus, the resultant lamp can be called as a "smart lamp" or intelligent lamp. The ability to change the color point of the lamp provides a new feature to general illumination that has the potential to generate new applications.

In this manuscript, we present our work on developing an algorithm to select a particular color from an RGB palette on a hand held appliance like a cellular phone. The objective is to design a control code that requires least amount of resources in a mobile platform. The description of human color vision and measurement of color in lighting applications are traditionally defined by the CIE tristimulus values, XYZ whereas RGB is a way of specifying color and a way of viewing color in graphics. The chosen color is converted into Hex Color data and the RGB LED intensity is computed to produce a color mixer to yield the selected color. The color palette and the software are written in Java and Android.

1.1 Development in LED Industry

The developments in LED components over the past few years have resulted in generating a new market for illumination engineering. Nowadays, high power LEDs of the order of 3 to 5 Watts are available with very high luminous efficiency in a very small form factor package. There are several approaches of using LEDs to generate lighting products. The advantages of LED lighting include long life, high reliability, stable poweroutput, high luminous efficiency. We focus our work on the use of Red, Green, Blue and white LED combinations to produce a light source that can have a variable color point while producing relatively high luminous efficiency.

1.2 Lifetime of LED

The useful life of LED lighting products is defined differently than that of other light sources, such as incandescent or compact fluorescent lighting (CFL). LEDs typically do not “burn out” or fail. Instead, they experience ‘lumen depreciation’, wherein the brightness of the LED dims slowly over time. Unlike incandescent bulbs, LED “lifetime” is established on a prediction of when the light output decreases by 30 percent.

1.3 How are LEDs Used in Lighting

LED**s** are incorporated into bulbs and fixtures for general lighting applications. Small in size, LEDs provide unique design opportunities. Some LED bulb solutions may physically resemble familiar light bulbs and better match the appearance of traditional light bulbs. Some LED light fixtures may have LEDs built in as a permanent light source. There are also hybrid approaches where a non-traditional “bulb” or replaceable light source format is used and specially designed for a unique fixture. LEDs offer a tremendous opportunity for innovation in lighting form factors and fit a wider breadth of applications than traditional lighting technologies.

1.4 LEDs and Heat

LEDs use heat sinks to absorb the heat produced by the LED and dissipate it into the surrounding environment. This keeps LEDs from overheating and burning out. Thermal management is generally the single most important factor in the successful performance of an LED over its lifetime. The higher the temperature at which the LEDs are operated, the more quickly the light will degrade, and the shorter the useful life will be.

LED products use a variety of unique heat sink designs and configurations to manage heat. Today, advancements in materials have allowed manufacturers to design LED bulbs that match the shapes and sizes of traditional incandescent bulbs. Regardless of the heat sink design, all LED products have been tested to ensure that they properly manage the heat so that the light output is properly maintained through the end of its rated life.

1.5 How is LED lighting different than other sources, such as incandescent and Compact Fluorescent (CFL)?

LED lighting differs from incandescent and fluorescent in several ways. When designed well, LED lighting is more efficient, versatile, and lasts longer.

LEDs are “directional” light sources, which means they emit light in a specific direction, unlike incandescent and CFL, which emit light and heat in all directions. That means LEDs are able to use light and energy more efficiently in a multitude of applications. However, it also means that sophisticated engineering is needed to produce an LED light bulb that shines light in every direction.

Common LED colors include amber, red, green, and blue. To produce white light, different color LEDs are combined or covered with a phosphor material that converts the color of the light to a familiar “white” light used in homes. Phosphor is a yellowish material that covers some LEDs. Colored LEDs are widely used as signal lights and indicator lights, like the power button on a computer.

2. INTRODUCTIONS TO ANDROID

Android is a mobile operating system developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets. Android's user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a virtual keyboard for text input.

In addition to touchscreen devices, Google has further developed Android TV for televisions, Android Auto for cars, and Android Wear for wrist watches, each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras, PCs and other electronics.

Initially developed by Android Inc., which Google bought in 2005, Android was unveiled in 2007, along with the founding of the Open Handset Alliance – a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices. Beginning with the first commercial Android device in September 2008, the operating system has gone through multiple major releases, with the current version being 7.0 "Nougat", released in August 2016.

Android applications ("apps") can be downloaded from the Google Play store, which features over 2.7 million apps as of February 2017. Android has been the best-selling OS on tablets since 2013, and runs on the vast majority of smartphones. As of May 2017, Android has two billion monthly active users, and it has the largest installed base of any operating system.

2.1 Features of Android:

After learning what is android, let's see the features of android. The important features of android are given below:

1) It is open-source.

2) Anyone can customize the Android Platform

3) There are a lot of mobile applications that can be chosen by the consumer.

4) It provides many interesting features like weather details, opening screen, live RSS (Really Simple Syndication) feeds etc.

It provides support for messaging services(SMS and MMS), web browser, storage (SQLite), connectivity (GSM, CDMA, Blue Tooth, Wi-Fi etc.), media, handset layout etc.

2.2 Categories of android applications:-

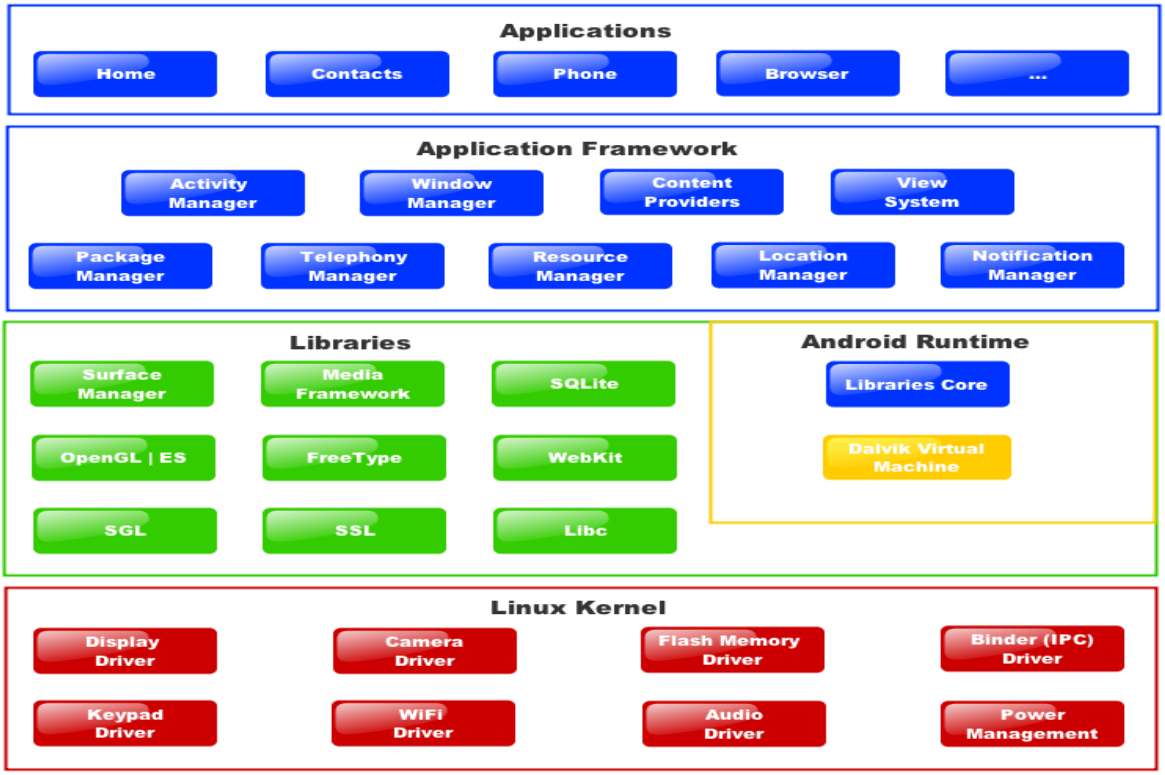
There are many android applications in the market. The top categories are:

- Entertainment
- Tools
- Communication
- Productivity
- Personalization
- Music and Audio

2.3 Android Architecture

Android architecture or Android software stack is categorized into five parts:

1. linux kernel
2. native libraries (middleware),
3. Android Runtime
4. Application Framework
5. Applications



2.4 PLATFORM USED

ANDROID STUDIO 2.2

Android Studio is the official integrated development environment(IDE) for Android platform development.

It was announced on May 16, 2013 at the Google I/O conference. Android Studio is freely available under the Apache License 2.0.

Android Studio was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.

Based on JetBrains IntelliJ IDEAsoftware, Android Studio is designed specifically for Android development. It is available for download on Windows, Mac OS X and Linux, and replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development.

WINDOWS 10

Windows edition

Windows 10 Home Single Language

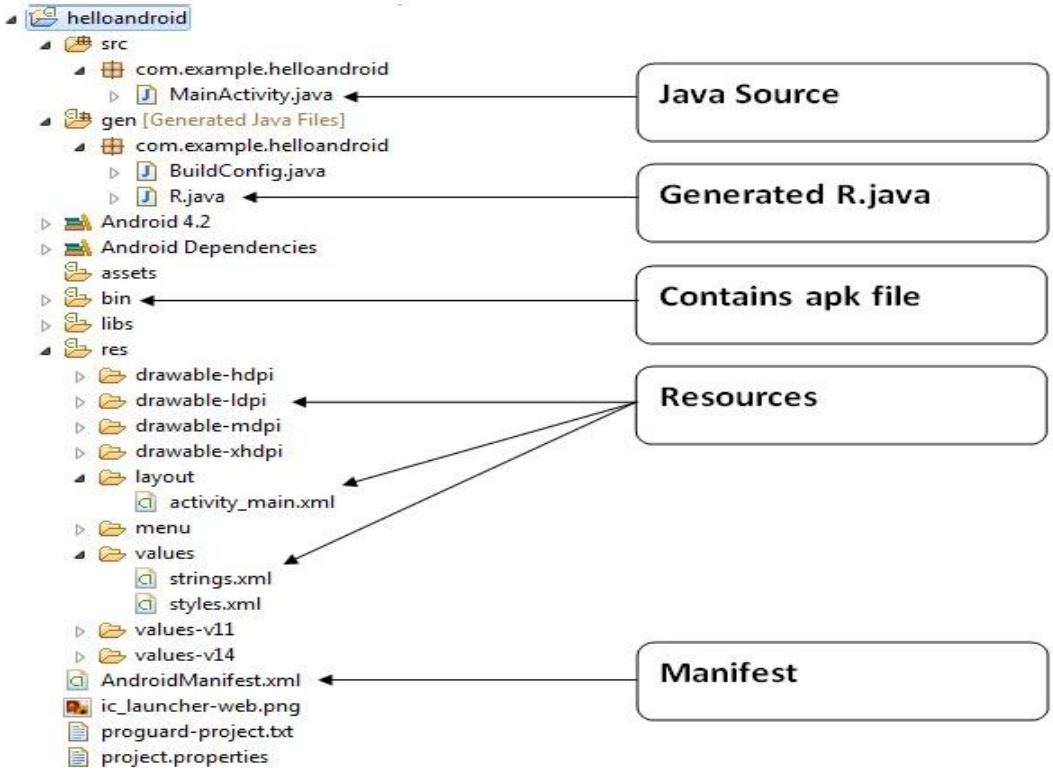
Windows 10

HARDWARE SPECIFICATIONS

HARDWARE	SPECIFICATION
CPU	Intel core i5
SPEED	2.20GHz
RAM	4GB
HARD DISK	1TB

2.5 INTERNAL DETAILS OF ANDROID PROJECT

Android application contains different components such as java source code, string resources, images, manifest file, apk file etc.



2.6 COMPILATION

To build and run your app, click **Run**. Android Studio builds your app and asks you to select a deployment target (an emulator or a connected device), and then deploys your app to it. You can customize some of this default behavior, such as selecting an automatic deployment target, by changing the run configuration.

If you want to use the Android Emulator to run your app, you need to have an Android Virtual Device (AVD) ready. If you haven't already created one, then after you click **Run**, click **create new emulator** in the **select deployment** dialog.

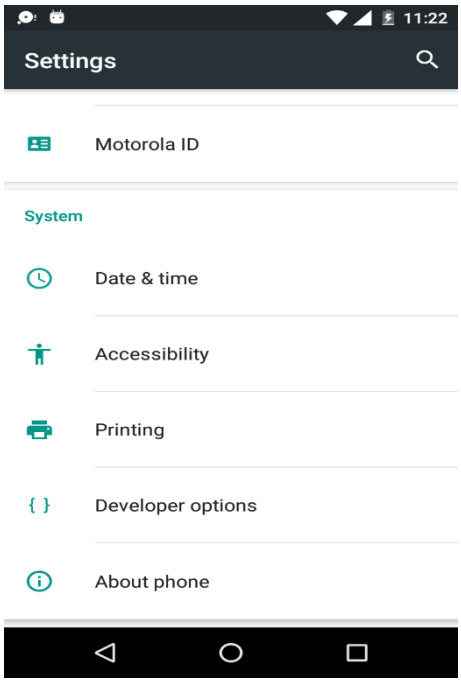
2.6.1 PROCESS OF COMPILATION

Step 1. Enable USB debugging in the connecting device.

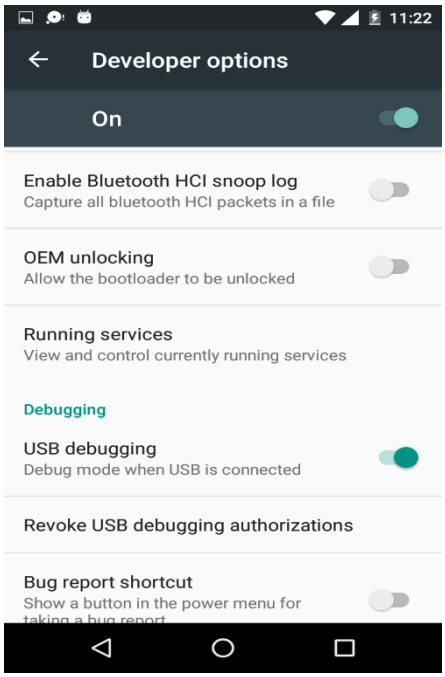
Step 2. Connect the devise with your computer using data cable.

Step3. Run the source code using connected device as an emulator.

STEP 1.a Go to settings.

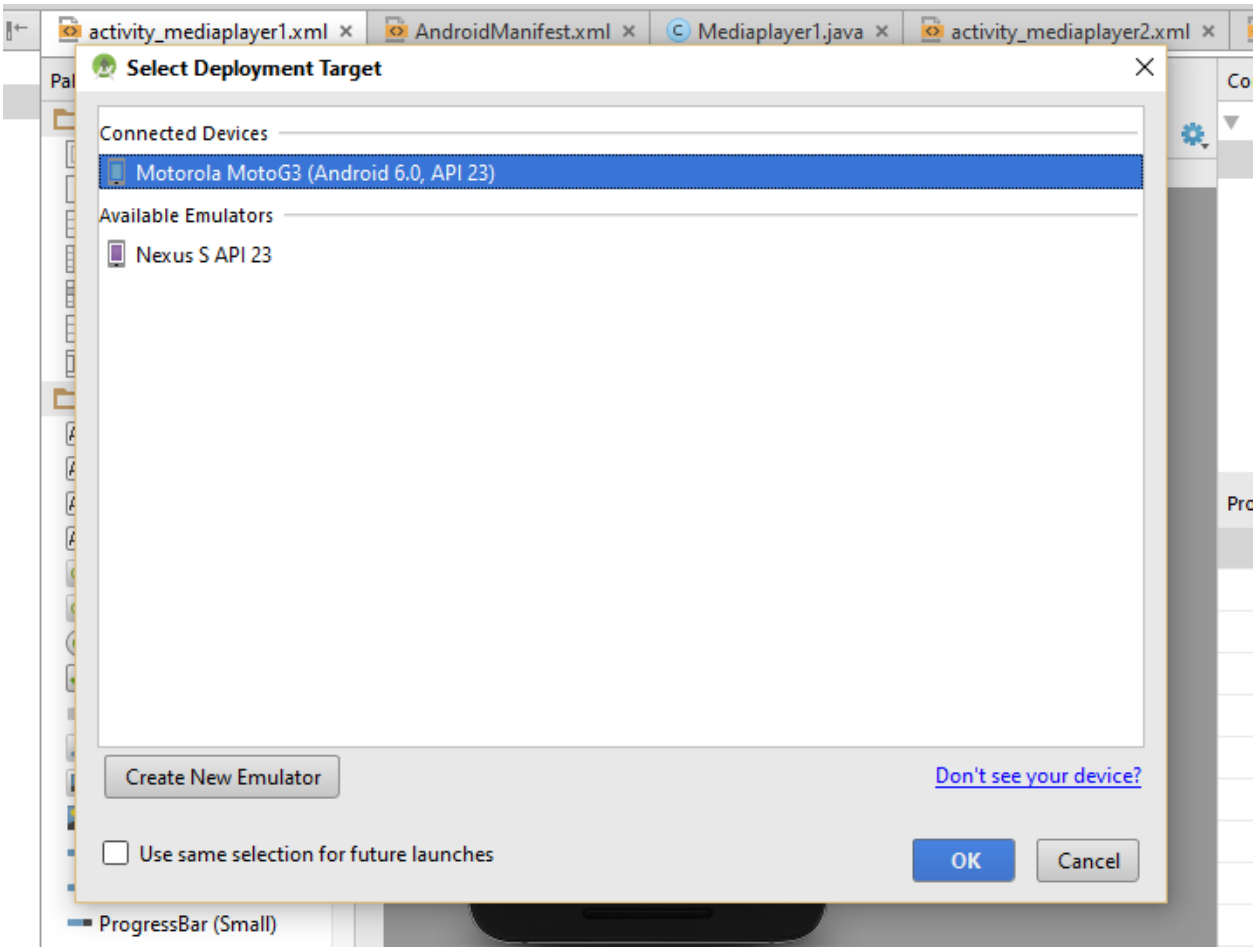


STEP 1.b. Click on developer option and enable USB debugging.



STEP 2. Connect your android device using data cable.

STEP 3. Run source file using android devise as emulator.



3. Internet of Things

The **Internet of things** (**IoT**) is the network of physical devices, vehicles, and other items embedded with electronics, software, sensors, actuators, and network connectivity which enable these objects to collect and exchange data. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure. Experts estimate that the IoT will consist of about 30 billion objects by 2020.

The IoT allows objects to be sensed or controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention. When IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, virtual power plants, smart homes, intelligent transportation and smart cities.

"Things", in the IoT sense, can refer to a wide variety of devices such as heart monitoring implants, biochip transponders on farm animals, cameras streaming live feeds of wild animals in coastal waters, automobiles with built-in sensors, DNA analysis devices for environmental/food/pathogen monitoring, or field operation devices that assist firefighters in search and rescue operations. Legal scholars suggest regarding "things" as an "inextricable mixture of hardware, software, data and service".

These devices collect useful data with the help of various existing technologies and then autonomously flow the data between other devices. The quick expansion of Internet-connected objects is also expected to generate large amounts of data from diverse locations, with the consequent necessity for quick aggregation of the data, and an increase in the need to index, store, and process such data more effectively.

Drawing representing the Internet of things (IoT)

3.1 Applications of IoT

The applications for internet connected devices are extensive. Multiple categorizations have been suggested, most of which agree on a separation between consumer, enterprise (business), and infrastructure applications. The former British Chancellor of the Exchequer George Osborne, posited that the Internet of things is the next stage of the information revolution and referenced the inter-connectivity of everything from urban transport to medical devices to household appliances.

The ability to network embedded devices with limited CPU, memory and power resources means that IoT finds applications in nearly every field. Such systems could be in charge of collecting information in settings ranging from natural ecosystems to buildings and factories, thereby finding applications in fields of environmental sensing and urban planning.

Intelligent shopping systems, for example, could monitor specific users' purchasing habits in a store by tracking their specific mobile phones. These users could then be provided with special offers on their favorite products, or even location of items that they need, which their fridge has automatically conveyed to the phone.Additional examples of sensing and actuating are reflected in applications that deal with heat, water, electricity and energy management, as well as cruise-assisting transportation systems.Other applications that the Internet of things can provide is enabling extended home security features and home automation. The concept of an "Internet of living things" has been proposed to describe networks of biological sensors that could use cloud-based analyses to allow users to study DNA or other molecules.

4. Smart Home Application

IoT devices are a part of the larger concept of Home automation, also known as domotics. Large smart home systems utilize a main hub or controller to provide users with a central control for all of their devices.

One application of the smart home is to provide assistance for disabled and elderly individuals. These home systems utilize assistive technology to accommodate an owner's specific disabilities. Voice control can assist users with sight and mobility limitations while alert systems can be connected directly to Cochlear implants worn by hearing impaired users. They can also be equipped with additional safety features. These features can include sensors that monitor for medical emergencies such as falls or seizures. Smart home technology applied in this way can provide users with more freedom and a higher quality of life.

Smart Led Lamp

5. Introduction to Arduino

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License(GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment(IDE) based on the Processing language project.

The Arduino project started in 2003 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

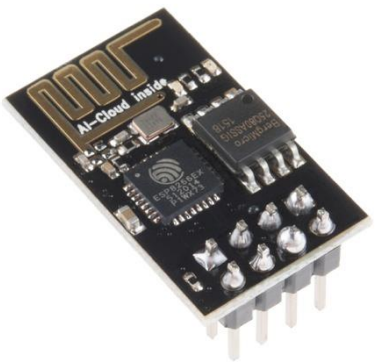
Arduino Uno R3

6. INTRODUCTION TO ESP8266

The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existance interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support. In the Documents section below you will find many resources to aid you in using the ESP8266, even instructions on how to transforming this module into an IoT (Internet of Things) solution.



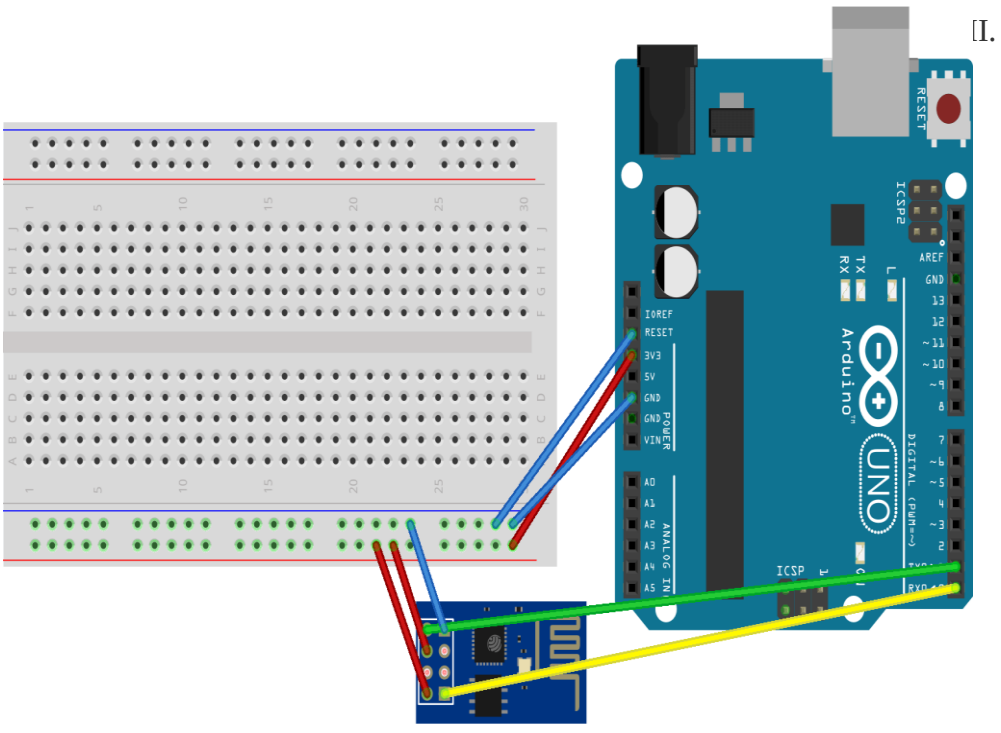
An ESP8266 Module

6.1 Connecting the ESP8266 to an Arduino

Connect the Arduino's **3v3** (3.3V) output to the red line on a breadboard. The ESP8266 works with 3.3V and not 5V, so this is necessary. If you want to connect other components that use 5V, you can connect the 5V output to the other red line of the breadboard, just make sure you don't connect the two.

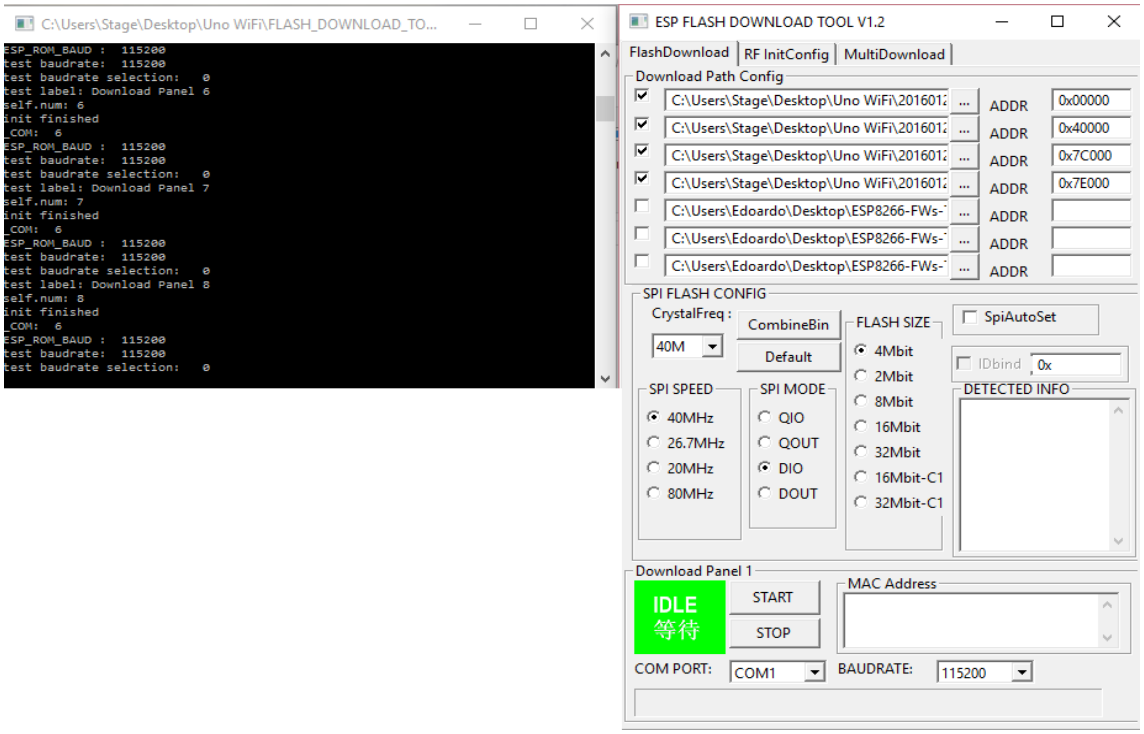
I. Connect **GND** (ground) to the blue line.
II. Connect the **RES** or **RESET** pin to the blue line. When you ground the reset pin, the Arduino works as a dumb USB to serial connector, which is what we want to talk to the ESP8266.
III. Connect the **RXD** pin of the Arduino to the **RX** pin of the ESP8266 (yellow color in the picture).
IV. Connect the **TXD** pin of the Arduino to the **TX** pin of the ESP (green color in the picture). Usually, when we want two things to talk to each other over serial, we connect the **TX** pin of one to the **RX** of the other (send goes to receive and the opposite). Here we do not have the Arduino talk to the ESP8266 though, our computer is talking to it *via*the Arduino.
V. Connect the **GND** pin of the ESP to the blue line and the **VCC** pin to the red line.
VI. Finally **CH_PD** goes to the red line, supposedly it will not work if you do not connect this

[I.



7. Arduino IDE

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 while saving 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 righthand 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.



Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

7.1 File Options

1. **New:** Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.
2. **Open:** Allows to load a sketch file browsing through the computer drives and folders.
3. **Open Recent:** Provides a short list of the most recent sketches, ready to be opened.
4. **Sketchbook:** Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.
5. **Examples:** Preferences Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.
6. **Close:** Closes the instance of the Arduino Software from which it is clicked.
7. **Save:** Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as.." window.
8. **Save as:** Allows to save the current sketch with a different name.
9. **Page Setup:** It shows the Page Setup window for printing.
10. **Print:** Sends the current sketch to the printer according to the settings defined in Page Setup.
11. **Quit:** Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE.

7.2 Edit

- **Undo/Redo:** Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.
- **Cut:** Removes the selected text from the editor and places it into the clipboard.
- **Copy:** Duplicates the selected text in the editor and places it into the clipboard.

- **Copy for Forum:** Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.
- **Copy as HTML:** Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.
- **Paste:** Puts the contents of the clipboard at the cursor position, in the editor.
- **Select All:** Selects and highlights the whole content of the editor.
- **Comment/Uncomment:** Puts or removes the // comment marker at the beginning of each selected line.
- **Increase/Decrease Indent:** Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.
- **Find:** Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.
- **Find Next:** Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.
- **Find Previous:** Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

7.3 Sketch

- **Verify/Compile:** Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.
- **Upload:** Compiles and loads the binary file onto the configured board through the configured Port.
- **Upload Using Programmer:** This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a Tools -> Burn Bootloader command must be executed.
- **Export Compiled Binary:** Saves a .hex file that may be kept as archive or sent to the board using other tools.

- **Show Sketch Folder:** Opens the current sketch folder.
- **Include Library:** Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.
- **Add File:** Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side o the toolbar.

8.RGB LED LIGHTS

How do RGB LEDs work?

LED is a combination of 3 LEDS in just one package.

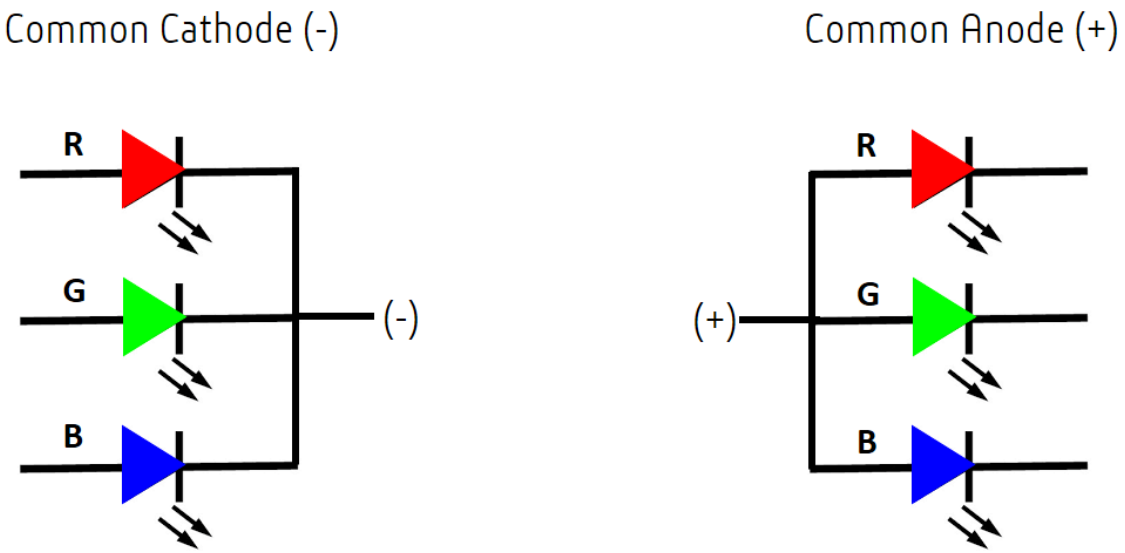
- 1x **R**ed LED
- 1x **G**reen LED
- 1x **B**lue LED

The color produced by the RGB LED is a combination of the colors of each one of these three LEDs. An RGB LED looks like this:

8.1 Types of RGB LED

There are common anode RGB LEDs and common cathode RGB LEDs. As you can see, the 3 LEDs can share the cathode or the anode. This results in an RGB LED that has 4 pins, one for each LED, and one common cathode or one common anode.

The common anode RGB LED is the most popular type.



How to create different colors?

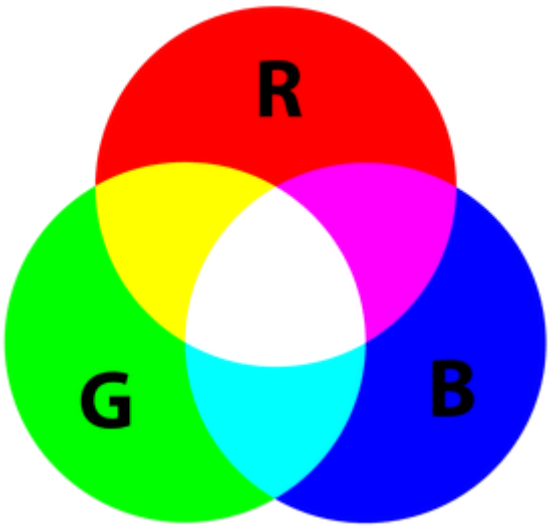
You can create one of those three colors – red, green or blue – by activating just one LED.
For example, if you want to produce blue, you activate the blue LED and turn off the other two.

8.2 Mixing colors

To produce other colors, you can combine the three colors in different intensities. To generate different colors you can use PWM to adjust the brightness of each LED.

As the LEDs are very close to each other, we can only see the final colors result rather than the three colors individually.

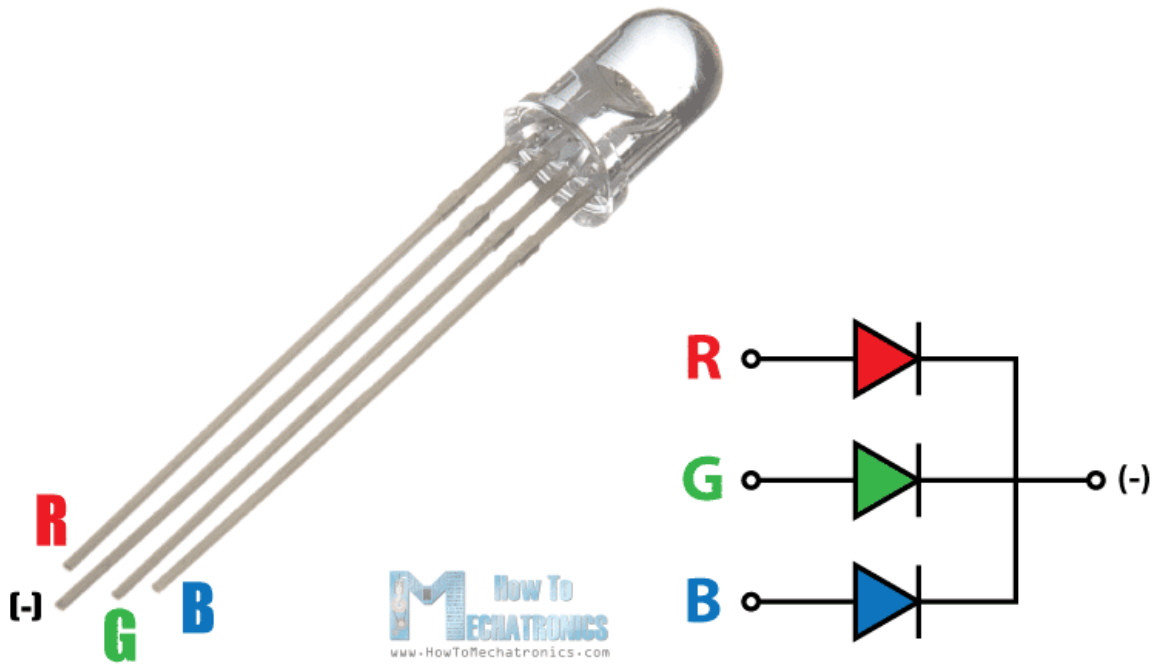
To have an idea on how to combine the colors, take a look at the following chart. This is the simplest color mixing chart, there are more complex color charts on the web.



8.3 RGB LED Pins

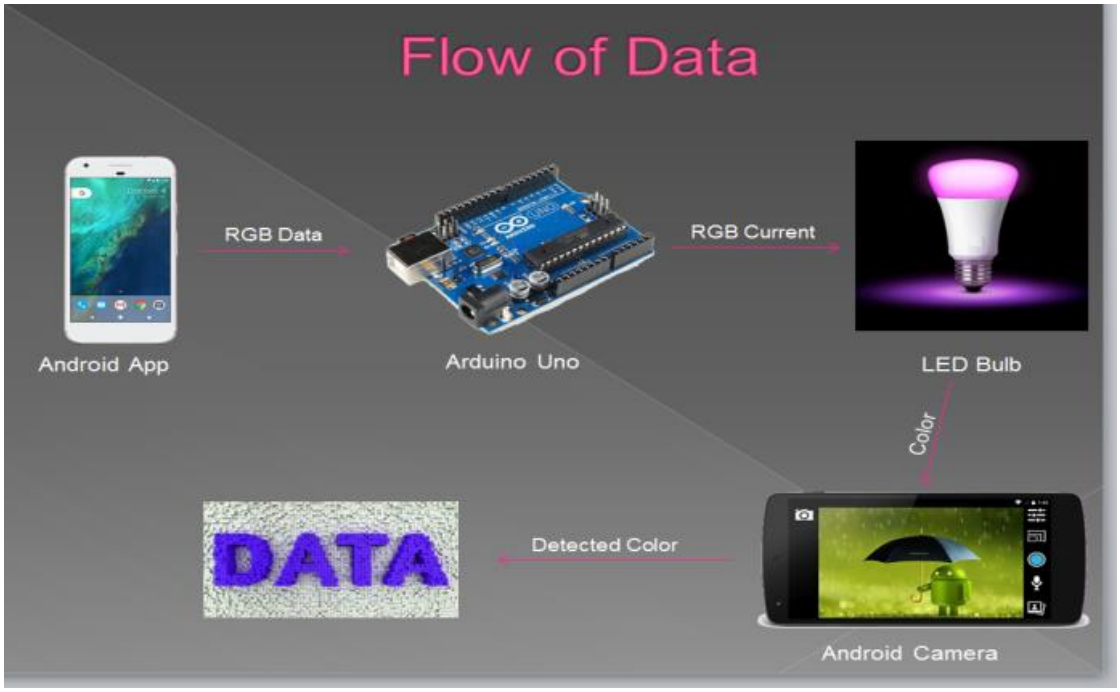
RGB LEDs have 4 pins which can be distinguished by their length. The longest one is the ground (-) or voltage (+) depending if it is a common cathode or common anode LED, respectively.

The other three legs correspond to red, green, and blue, as you can see in the figure below:



9. FUTURE ENHANCEMENT

- An intensity slider could be used by which the user can select the intensity of the light emitted by the LED.
- This project can also be used as a data transfer tool by using the transition of the color emitted by the LED.



10. CONCLUSION

The project “Smart LED lamp” has been implemented successfully. LED lamp can be controlled using an Android app via Wi-Fi. The LED lamp and ESP8266 are connected to the Arduino. The ESP8266 module is connected to the home network. An HTTP Web Server is programmed into the Arduino which accepts red, green, and blue values from the HTTP URL. The Android app takes user input for color value and extract RGB values and then send these values to the ESP8266 module via an HTTP request. Here it is to be noted that both the devices should be in the same network.

11.REFRENCES

- http://www.ledsmagazine.com/led-design-and-manufacturing/research-and-development.html
- http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?reload=true&punumber=6488907
- http://www.Arduino.stackexchange.com
- http://www.wikipedia.com/esp8266

2.