IOT Based Digital Color Control for Multicolor LED Illumination System

Prof. P Y Kumbhar[1], Anuj Pandharpatte[2], Rutuja Jadhav[3], Aparna Kalpande[4]

[1]Faculty, Department of Electronics and Telecommunication Engineering, Walchand Institute of Technology, Solapur, Maharashtra, INDIA
[1] pravinkumbhar03@gmail.com

[2]Student, Department of Electronics and Telecommunication Engineering, Walchand Institute of Technology, Solapur, Maharashtra, INDIA
[2] anuj9545195030@gmail.com

[3]Student, Department of Electronics and Telecommunication Engineering, Walchand Institute of Technology, Solapur, Maharashtra, INDIA
[3] rutuja.jadhav8@gmail.com

[3]Student, Department of Electronics and Telecommunication Engineering, Walchand Institute of Technology, Solapur, Maharashtra, INDIA
[4] aparnakalpande2996@gmail.com

ABSTRACT

LED's are great. But with every project comes a point where flashing LED is just not enough. For such cases RGB(Red, Green, Blue) LED is the answer. With an RGB LED you would be able to glow any colour your heart desires. A single LED die can only emit monochromatic light which could be one of the three primary colors - red, green and blue, known as RGB. To realize more colors, three LED dies need to be used together for RGB color mixing. Commonly 7 colors can be produced by controlling the switch of the channel for each primary color. To produce more than 7 colors, each color channel should be able to change in brightness, not just switched on or off. A popular control method is PWM, of which the cycle duty range determines the available brightness levels. The more the levels are available, the more colors can be produced. Apart from the popularity in applications like outdoor decoration lighting in cities, stage lighting designs, home decoration lighting and LED display matrix, RGB color mixing technology can also be found recently in LCD backlighting and projectors

Index Term— PWM, Intensity, RGB LED, Duty Cycle

1. INTRODUCTION

Internet of Things (IoT) is an ecosystem of connected physical objects that are accessible through the internet. The 'thing' in IoT could be a person with a heart monitor or an automobile with built-in-sensors, i.e. objects that have been assigned an IP address and have the ability to collect and transfer data over a network without manual assistance or intervention. The embedded technology in the objects helps them to interact with internal states or the external environment, which in turn affects the decisions taken. This concept of IoT has been used here along with two different ecosystems ie. Android and Arduino. Ultimately a digital colour controller is to be developed with the help of these.

2. OBJECTIVE

The objective of this project is to pick any desired colour using smart phone with the help of an Android Application, convert the sensed colour into its corresponding RGB value and make the Arduino turn on the RGB led with the same colour we picked using the Android Application. This task uses Node MCU-module as a receiver, which accepts the sent data and provides it to the Arduino

3. PROBLEM STATEMENT

The major shortcoming of the current existing systems performing this task is limited color range and its accessing techniques. The ultimate aim is to display a wide range of desired colors on a distantly placed RGB LED strip, with concatenation of IoT, Android Application, and other elements.

4. BREIF DESCRIPTION OF PROJECT

The entire system can be subdivided into 3 major sections namely

- The transmitter
- Cloud
- Receiver

These are described briefly as follows

4.1. The Transmitter

The smart phone itself works as the transmitter.The Android Application performs the task Of sensing the colour and converting it into the corresponding values of Red , Green and Blue colour intensities respectively.

4.2 CLOUD

ThingSpeak is an open source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP protocol over the Internet or via a Local Area Network. Thing Speak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates. ThingSpeak was originally launched by ioBridge in 2010 as a service in support of IoT applications.

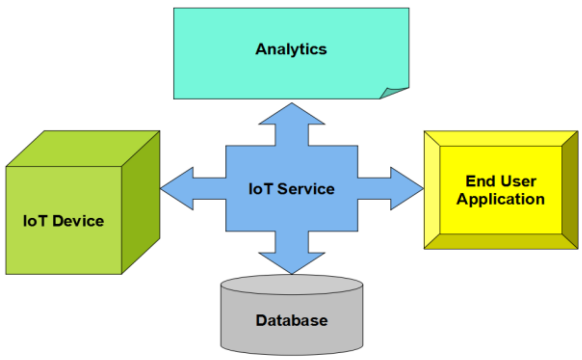

Fig 1:Block Diagram of cloud

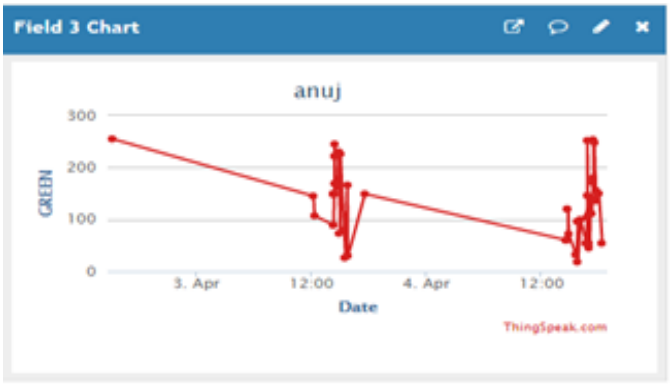

Fig 2: Field chart of Green colour

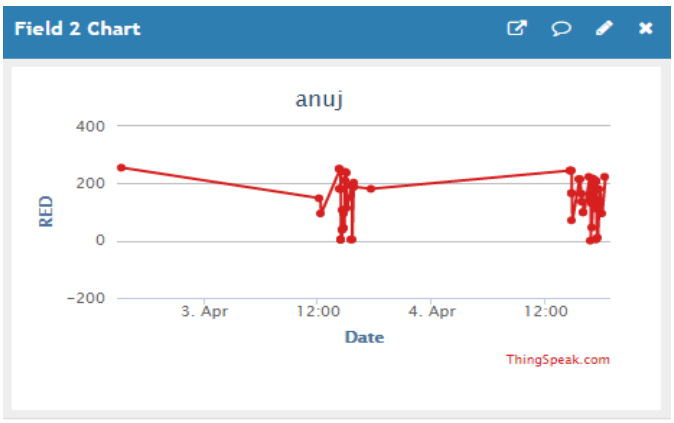

Fig 3: Field chart of Red colour

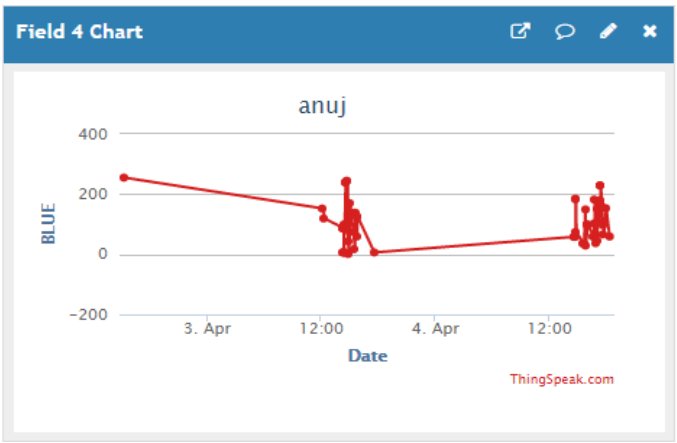

Fig 4: Field chart of Blue colour

4.3 RECIEVER

Here Node-MCU acts as a receiver which is used to retrieve the values of RGB which are plotted on the fields of thing Speak channel.

5. HARDWARE REQUIREMENT

1. Node MCU
2. Controller
3. Current Boosting Circuit.

1. Node MCU

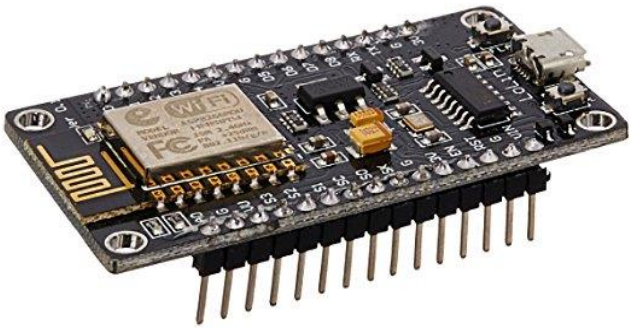

Fig 5: Node MCU-ESP8266 WiFi Development Board

The NodeMCU (Node MicroController Unit) is an open source software and hardware development environment that is built around a very inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains all crucial elements of the modern computer: CPU, RAM, networking (wifi), and even a modern operating system and SDK. When purchased at bulk, the ESP8266 chip costs only $2 USD a piece. That makes it an excellent choice for IoT projects of all

2. CONTROLLER

Fig6: AtMega 328 IC

The high-performance Microchip 8-bit AVR RISC-based microcontroller combines 32KB ISP flash memory with read-while-write capabilities, 1KB EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts,serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts.

3. CURRENT BOOSTING CIRCUIT

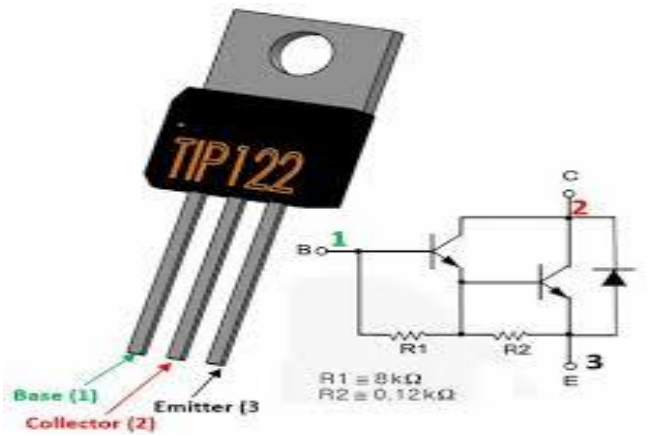

Fig7: TIP 122 Transistor

TIP-122 is used to boost current.

Features:

1. Darlington Medium- power NPN transistor.
2. High DC current gain (hFE) typically 1000.
3. Contineous collector current(IC) is 5A.
4. Collector- Emitter voltage (VCE) is 100V.
5. Collector- Base voltage (VCB) is 100V.
6. Emitter Base Voltage (VBE).

6. SOFTWARE REQUIREMENTS

1. Android studio
2. Arduino IDE

ARDUINO IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them. The Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

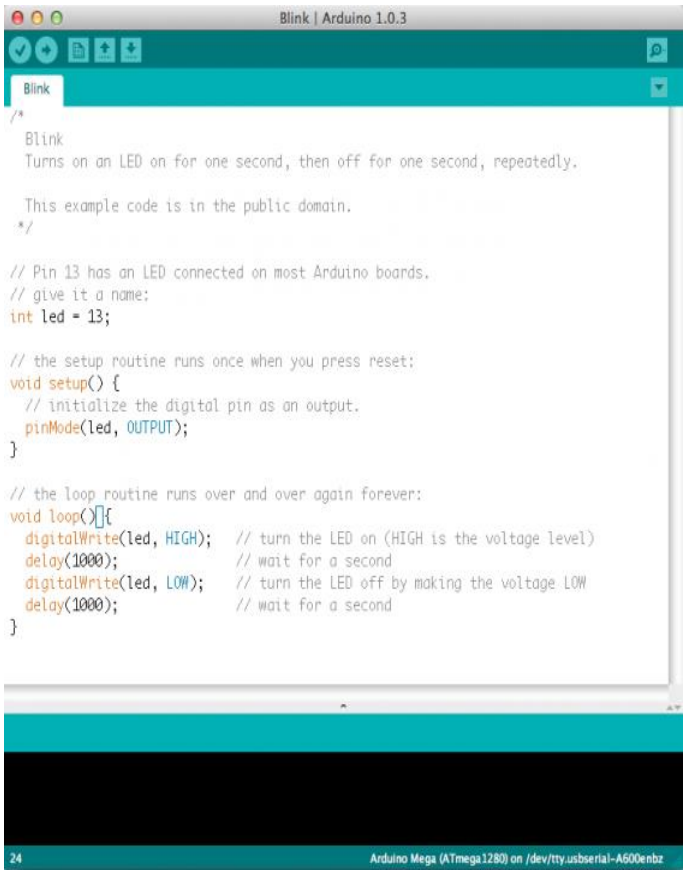

Fig-8:Arduino IDE Window

The Arduino integrated development environment (IDE) isa cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 2.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

ANDROID STUDIO

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as primary IDE for native Android application development Android Studio was announced on May 16, 2013 at the Google I/O conference It 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. The current stable version is 3.1 released in March 2018 The following features are provided in the current stable version Gradle-based build support Android-specific refactoring and quick fixes Lint tools to catch performance, usability, version compatibility and other problems ProGuard integration and app-signing capabilities Template-based wizards to create common Android designs and components A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations Support for building Android Wear apps Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine

7. IMPLEMENTATION

STEP 1:

Fig-9:ColorFetcher App View

The initial step is to open the colour fetcher android application from the smart phone.

STEP.2

Fig 10:Colour Selection in ColorFetcher App

Next step is to click an image and tap on the colour you desire

STEP.3

The next step is to send the sensed R,G,B colour values to the cloud via.internet.

STEP.4

The corresponding R,G,B values present at the cloud are to be fetched and provided to the controller. This task is performed by the NodeMCU module configured as receiver.

STEP.5

The input received from Node MCU is converted to corresdsponding PWM output by controller and given to current boosting circuit.

STEP.6

The output of current boosting circuit is given to the RGB LED strip.

8. RESULT

The following images show the colour being picked through the Android Application is being displayed on the RGB LED strip. The accuracy of the displayed colour depends on the quality of the Smartphone camera

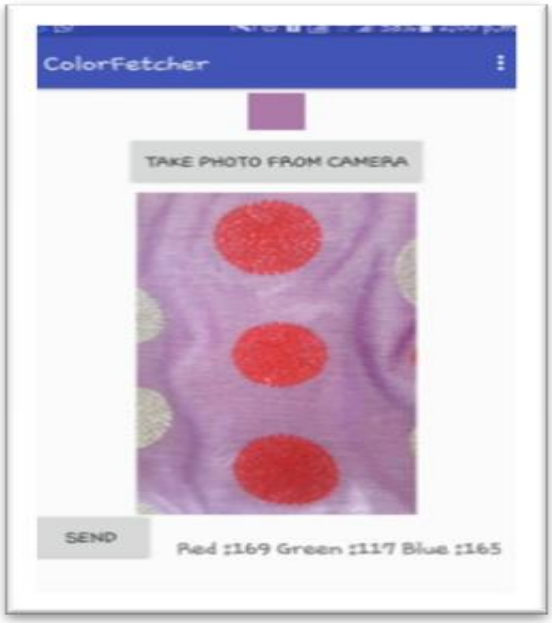

Fig-11.Desired Colour Selection

Fig-12.Desired colour Output

REFERENCES

[1] Zion market research, LED Lighting Market for Residential, Architectural and Outdoor Applications: "Global Industry Perspective, Comprehensive Analysis and Forecast, 2016 - 2022". Accessed 24.1.2017

[2] Lech Grodzki, Bialystok University of Technology, "The Comparison of the Pulse and Constant-current LED

Driving", PRZEGLĄD ELEKTROTECHNICZNY, ISSN 0033-2097, R. 89 NR 11/2013.

[3] A.Wilkins, J. Veitch, and B. Lehman, "LED lighting flicker and potential health concerns:IEEE standard PAR1789 update," in Proc. IEEE Energy Convers. Congr. Expo., 2010, pp. 171–178.

[4] R. Cordeiro, A. Cardoso, R. Duarte, D. Soares, G. Pereira, W. Vizzotto, V. Bender and T. Marchesan, "Indirect control of luminous flux and chromatic shift methodology applied to RGB LEDs", 2014 11th IEEE/IAS International Conference on Industry Applications, 2014..

[5] Tan, Y.K.; Huynh, T.P.; Wang, Z.Z. Smart personal sensor network control for energy saving in DC grid powered LED lighting system. IEEE Trans. Smart Grid 2013, 4, 669–676..

[6] Ding, Y.S.; Jin, Y.L.; Ren, L.H.; Hao, K.R. An intelligent self-organization scheme for the internet of things. IEEE Comput. Intell. Mag. 2013, 8, 41–53

[7] Choi, S.J.; Kim, T.H. Symmetric current-balancing circuit for LED backlight with dimming. IEEE Trans. Ind. Electron. 2012, 59, 1698–1707

[8] Liu, T.H.; Yi, S.C.; Wang, X.W. A fault management protocol for low-energy and efficient wireless sensor networks. J. Inf. Hid. Multimedia Signal Process. 2013, 4, 34–45.

