**Brief Progress Report**

**Colour Detection Based Lighting Control System**

**16 April 2018**

**Yarmouk University**

**Submitted by: Mohammed Hawamdeh**

**As a part of Hajjawi’s Smart E-Learning Room Project**

**Contents**

**Introduction…………………………………………………………………………………. 3**

**System Objectives ………………………………………………………………………...… 3**

**System Overview**

System Block Diagram**…………………………………………………………………..…… 4**

Hall’s Computer Behaviour**…………………………………………………………….….… 5**

Raspberry Pi Behaviour**………………………………………………………………...……. 5**

**Software System**

Colour Detecting Application**…………………………………………………….……..…… 6**

Screen Capture**…………………………………………...……………………….……..…… 6**

Major Colour Detecting**…………………………………………………….……..……….… 7**

Server-Side Operations**…………………………………………………….……...……….… 7**

Philips Hue API**…………………………………………………………….……..……….… 8**

**Hardware Tools …………………………………………………..……….……..……….… 8**

Raspberry pi 3 module B

Philips Hue Colour Ambiance Smart Bulb.

Philips Hue Smart Bridge

Conclusion **…………………………………………………..……….……..…………….… 8**

**Introduction**

This report documents the work and progress of Lighting Control Team regarding enhancing guest’s picture experience using colour detection system. This system is proposed to be implemented in Hajjawi’s Smart E-learning Room project.

**System Objectives**

* Less hardware.

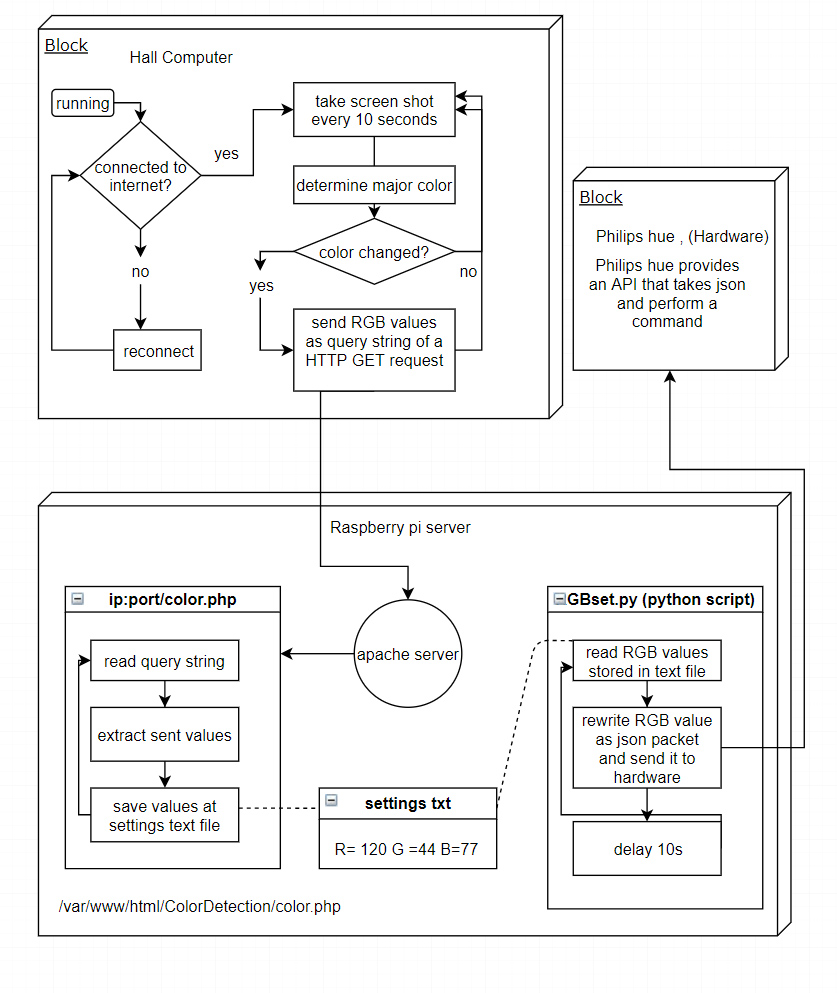
Accomplishing task without engaging sensors while getting more accurate results by image processing and clustering algorithms.

* Boosting attendees visual experience using colours

Being able to use colours consciously and harmoniously helps to attract attention, make a statement or even setting the mood and the tone of the audience.

**System Overview**

* System Block Diagram



* Hall’s Computer Behaviour

The computer which is connected to the hall projector need install a custom colour detecting application built by the team and which to be discussed in Software System section ( page 6).

As long as the computer’s running and connected to the network the screen shot application should take a screen shot , determine major colour RGB values and pass it to server as a query string in the form of HTTP GET Request as long as the colour is changed.

Only one computer is allowed to control the colour this can be accomplished in several ways , an ID can be generated by the C# application to be sent as a part of the Query string in the URL .The Raspberry pi should save the first received ID and only allow this ID to control Lights until a quit command is sent too.

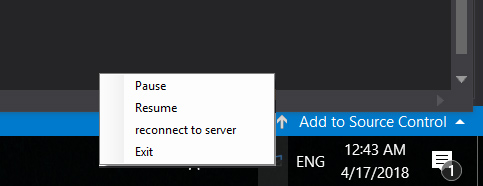
* Raspberry Pi Behaviour

The module is used as server, it receives the request and if it pass the conditions the command will be executed and output will be passed to hardware, more details in Software System section (page 6).

**Software System**

* Colour Detecting Application

The application is built using Visual Studio 2018 as a Windows Form Application, no visual GUI is displayed on screen to avoid appearing in the screenshot, but the application itself is always minimized to system tray which provides a way to control the application without the need of GUI as shown below.



Windows is expected to be the operating system of the device controlling the colours in order to run the C# Windows Forms Application. Support for other systems should be considered as a future goal.

* Screen Capture

To ensure the computability with different devices, the captured image width and height are multiplied with the scaling factor used on the device. Screen shots are passed to next stage to determine major colour without being saved on disk.

* Detecting Major Colour

Captured images can include many colour degrees. AForge.NET, a framework designed for developers in the fields of computer vision and AI is used to perform a posterization filter.

Posterization filters are used for the conversion of a continuous gradation of tone to several regions of fewer tones as shown below.

The filter is applied to approximating colour ratios of adjacent pixels.

Major colour is detected by sorting a list of pixel colours.

RGB values are calculated and sent as Query string parameters of HTTP GET Request:

Ex. ip/ColorDetect/color.php?R=222&G=100&b=0 .

* Server-Side Operations

Two main tasks is done on Raspberry pi:

1. A background process is started after booting to Raspbian OS, and kept running until shutdown. This process checks for a text file used for storing RGB values to be passed for hardware by the same process .Setting file is stored locally on the Raspberry pi.
2. Receiving requests is done using Apache HTTP Server Project and PHP, a typical Colour Control request should have at least one of The letters “R , G ,B” with a corresponding value from 0 to 255 denoting the ratio. Other colour value if not passed will be set to zero.

Regular Expressions are used to extract colour parameters without boilerplate code.

Other parameters can be used such as ID to limit the control for one user, also this can be used to perform other commands or light effects.

* Philips Hue API

Philips Hue can be controlled or integrated with various applications using API. The bridge has a powerful RESTful interface, and the full API Documentation is available online.

RGB values must be converted to TSL form in order to be received at the API.

**Hardware Tools**

* Raspberry Pi 3 module B.
* Philips Hue Colour Ambiance Smart Bulb.
* Philips Hue Smart Bridge.