Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
CSE 4316: Senior Design I
Summer 2021

The Lighters
LightHouse

Dipika Giri
Dhruv Patel
Parker Skinner
David Trimino

Revision History

Revision	Date	Author(s)	Description
0.1	8.7.2021	DG,DP,PS,DT	document creation
0.2	8.16.2021	DG,DP,PS,DT	First draft complete

CONTENTS

List of Figures

List of Tables

1 Introduction

The overall product concept is a web application acts like a controller which has multitude of functionality. The main functionality that the system will need is to take in the user's customization preference and display them with the LED lights. The system should also store the user's customization bundles in a cloud repository for others to have and share.

2 System Overview

The system will essentially consist of two layers: one layer for the raspberry pi and the other for an ESP8266 micro-controller. The raspberry pi layer will encapsulate the front-end and the back-end of the web application. The raspberry pi is the only hardware system that interacts with the user and sends the changes or configurations that the user specifies to all other systems. The ESP8266 layer is essential for LED lights that span longer than 5 meters due to the power needed to maintain color accuracy. The ESP8266 enables the LED lights to adapt to the user's configurations.

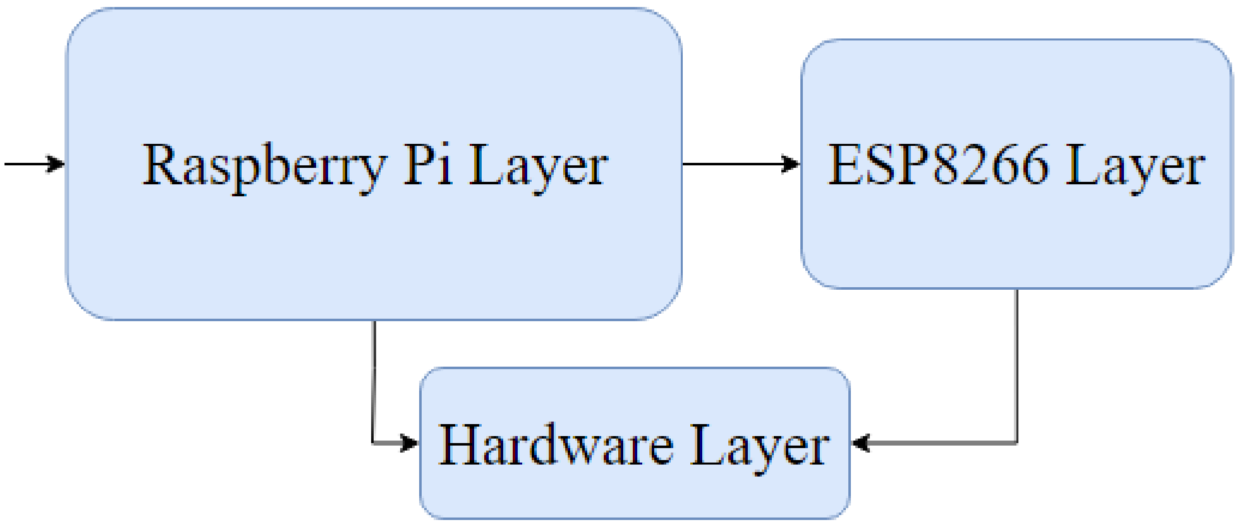

Figure 1: A simple architectural layer diagram

2.1 Raspberry Pi Layer Description

The raspberry pi layer is the user interface for the user using the LitHouse LED Web Application and the background processes that deliver the displays that the user requested. This layer consists of 5 subsections, web application, database, designer interface, pattern saved, and web server subsystem layers. Overall this will essentially be the application layer consists of two sub layers: user and back. The user layer is responsible for what the user sees and interacts with. The back layer is responsible for changing configuration of LEDs, sending data to an ESP8266 micro-controller, and handles any other background processes.

2.2 ESP8266 Layer Description

The ESP8266 layer is responsible for receiving data packets, from the web server on the raspberry pi, which contain configuration instructions on how to display LED lights. The ESP8266 is responsible for giving the LED lights that are not connected to the raspberry pi the smart functionality, for them to be able to change colors or patterns.

2.3 ESP8266 Layer Description

The hardware layer includes the power supply that will be injected every 5 meters as well as for the Pi and the micro controller and the WS281B LED lights. The lights receive inputs from the power supply as well as the saved patterns from the pi layer and output from the ESP8266 layer for what patterns to run.

3 SUBSYSTEM DEFINITIONS & DATA FLOW

This section breaks down your layer abstraction to another level of detail. Here you graphically represent the logical subsystems that compose each layer and show the interactions/interfaces between those subsystems. A subsystem can be thought of as a programming unit that implements one of the major functions of the layer. It, therefore, has data elements that serve as source/sinks for other subsystems. The logical data elements that flow between subsystems need to be explicitly defined at this point, beginning with a data flow-like diagram based on the block diagram.

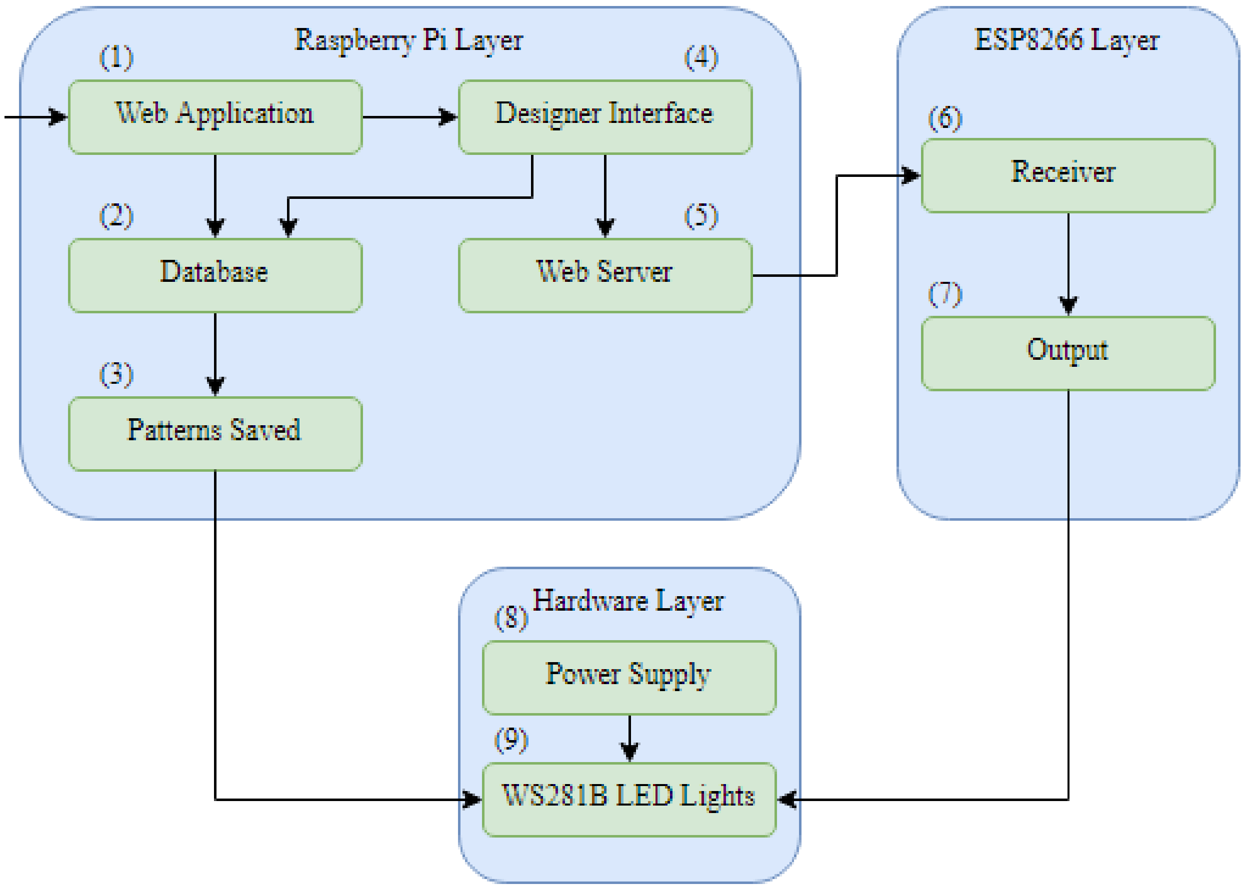

Figure 2: Data flow diagram of system

4 Raspberry Pi Layer Subsystem

The layer consists of web application, database, designer interface, pattern saved, and web server subsystem layer. This layer communicates with the micro-controller and provides a user interface. The user data is stored in the database and when the patterns are determined, the layer will communicate with web server or the LED strip. This layer should apply user data and create light patterns in the LED strips.

4.1 Web Application

Web Application subsystem layer is the interaction with the user. The goal is to get user data needed to run the simulation on the Raspberry Pi and create various light pattern. The data will be then be used to store it in Database subsystem and Designer Interface subsystem layer.

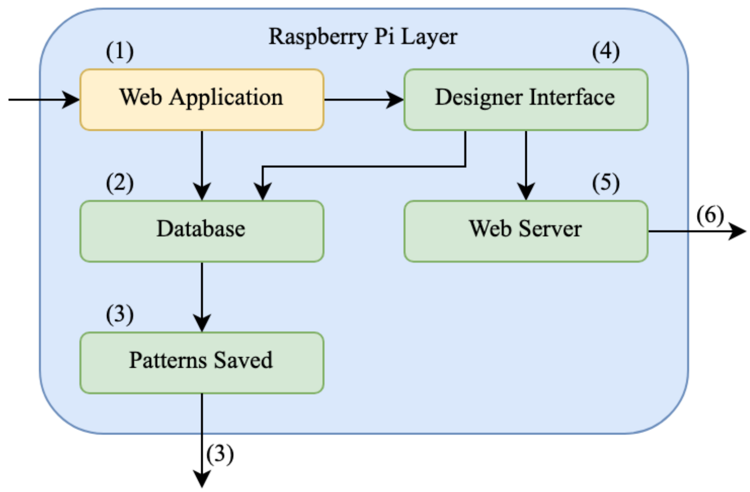

Figure 3: Raspberry Pi Subsystem 1

4.1.1 Assumptions

The user will be able to connect the device which is the Raspberry Pi the user is using. The user will be able to select an effect from various options and if the effect is custom, the layer will communicate with the Designer Interface to create a custom effect which are also, given in form of various option. The light effect data will saved and back-end will use that data.

4.1.2 Responsibilities

Web application subsystem layer will include interface to connect the devices, select effects, customize detailed effects and patterns, and upload music. The responsibility of this layer will include user data and saving the data to use in the Database. The user interface will also have a custom effect selection; the layer will communicate with Designer Interface to create a custom effect.

4.1.3 Subsystem Interfaces

Table 2: Web Application Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Select Device	IP Address	Send UDP packet
#2	Select Effects	Select Option	N/A
#3	Sync Music	Music	Audio Analysis

4.2 Database

Database subsystem layer stores user data that are need to run and create the design pattern for the LED. The database layer interacts with the Pattern Saved subsystem layer and analyzes the user data.

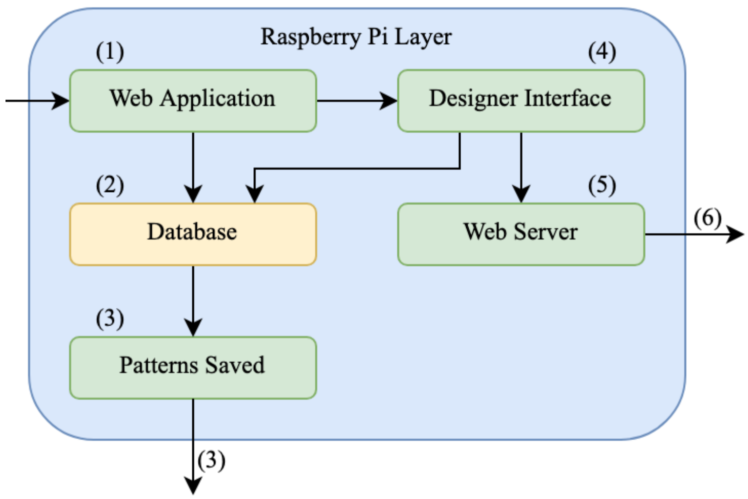

Figure 4: Raspberry Pi Subsystem 2

4.2.1 Assumptions

The layer stores the data from web application layer and designer interface layer. The data will be used in pattern saved layer.

4.2.2 Responsibilities

This layer is responsible for adding elements to the database. Every key is analyzed and managed. Any changes to the data should be managed using the web application and designer interface subsystem layer.

4.2.3 Subsystem Interfaces

Table 3: Database Subsystem interfaces

ID	Description	Inputs	Outputs
#1 Manage user data	Store user data user data	user data updated data table	data table #1

4.3 Pattern Saved

The pattern saved subsystem layer will analyze and store the data that is specifically used to design the light patterns or access the light pattern chosen.

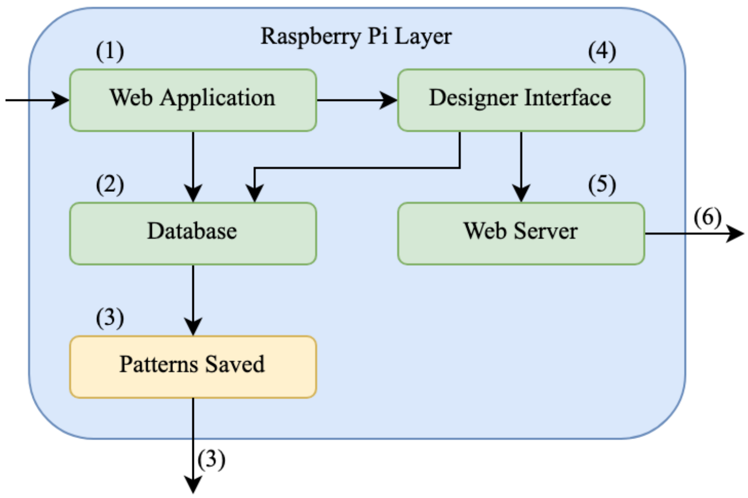

Figure 5: Raspberry Pi Subsystem 3

4.3.1 Assumptions

The layer uses the user data directly from the database to create and analyze the light patterns from the user data. This layer will then communicate with the hardware layer, more specifically, the LED strips.

4.3.2 Responsibilities

This layer is responsible for analyzing the user data to create the light pattern selected and execute it on the LED strips.

4.3.3 Subsystem Interfaces

Table 4: Patterns Saved Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Analyze user data	User data	Light pattern

4.4 Designer Interface

Designer Interface subsystem layer will provide various selection of effect a user can select. It is the extended version of the web application subsystem layer.

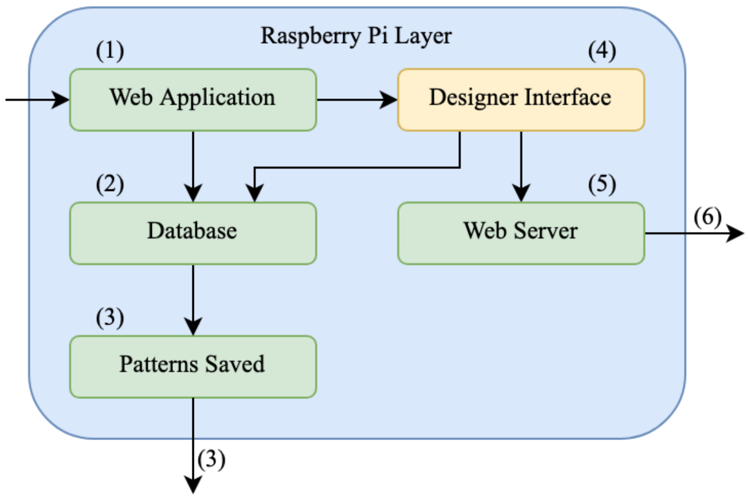

Figure 6: Raspberry Pi Subsystem 4

4.4.1 Assumptions

The layer provides custom light patter effect if the user has additional information. The layer includes a user interface with selection of light color, pattern, and time. The layer interacts with the database and the web server

4.4.2 Responsibilities

This layer is responsible for selecting custom light pattern design Using user interface, it shall create a pattern and save the data in the database.

4.4.3 Subsystem Interfaces

Table 5: Designer Interface Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Custom user data	Design pattern	Send Request
#2	Store data in Database	User Data Database Access	N/A

4.5 Web Server

The web server subsystem layer delivers web page to the user. This layer also stores and processes user data and interacts with the ESP8266 layer of the receiver end.

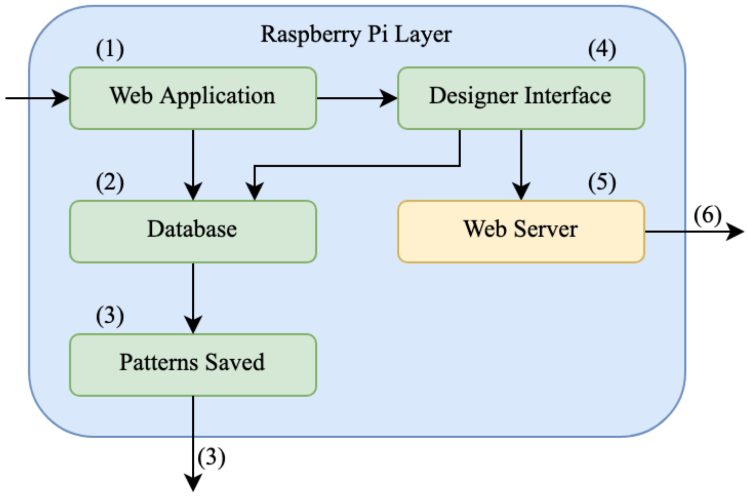

Figure 7: Raspberry Pi Subsystem 5

4.5.1 Assumptions

The layer receives the user data from the web application and designer interface subsystem layer and processes those data and sends all the data needed to the ESP8266 layer.

4.5.2 Responsibilities

This layer is responsible for display the website content and store, process and deliver web page to users . This layer sends packets to the ESP8266 layer

4.5.3 Subsystem Interfaces

Table 6: Web Server Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Send packets	data	Response
#2	Process webpage	User Data	Display content

5 ESP2866 Layer Subsystems

The ESP8266 Layer is the micro-controller that will directly interface with the led lights and send instructions to change the LED states

5.1 Receiver

The ESP8266 receives packets via a Wi-Fi network with simple TCP/IP connections using Hayes-style commands. These commands will be sent from the Pi directly to the Micro-Controller

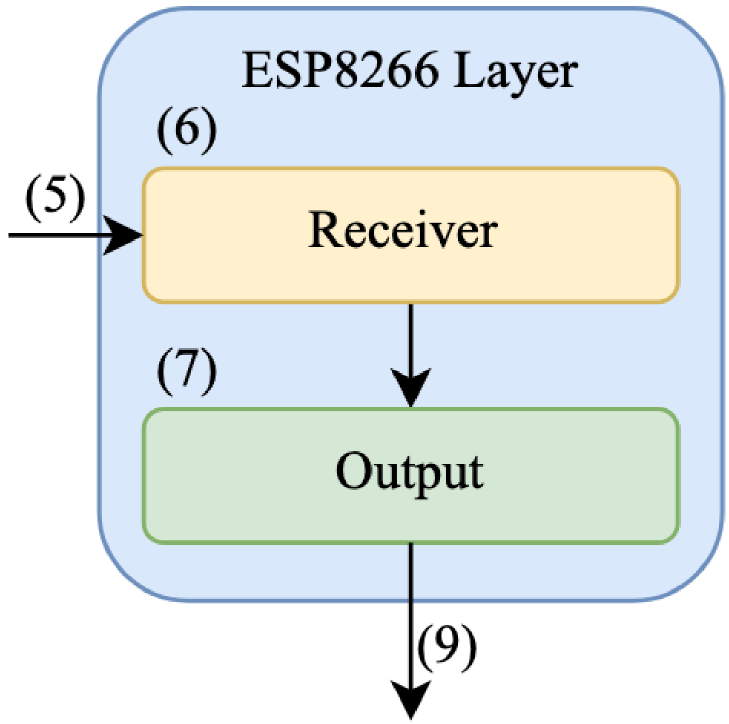

Figure 8: ESP8266 Subsystem 1

5.1.1 Assumptions

We are assuming that all packets sent to the micro-controller will be from the Pi only and nothing else will try to interface with it

5.1.2 Responsibilities

This subsystem has the responsibility of receiving packets and parsing the data contained in those packets

5.1.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

Table 7: Reciever Subsystem interfaces

ID	Description	Inputs	Outputs
#5	Incoming Network Packets	IP Packet	N/A
#6	Parsed Network Packet	N/A	Parsed Data

5.2 Output

The output will read the parsed data received by the receiver and send commands to the LED Strips based off of that data

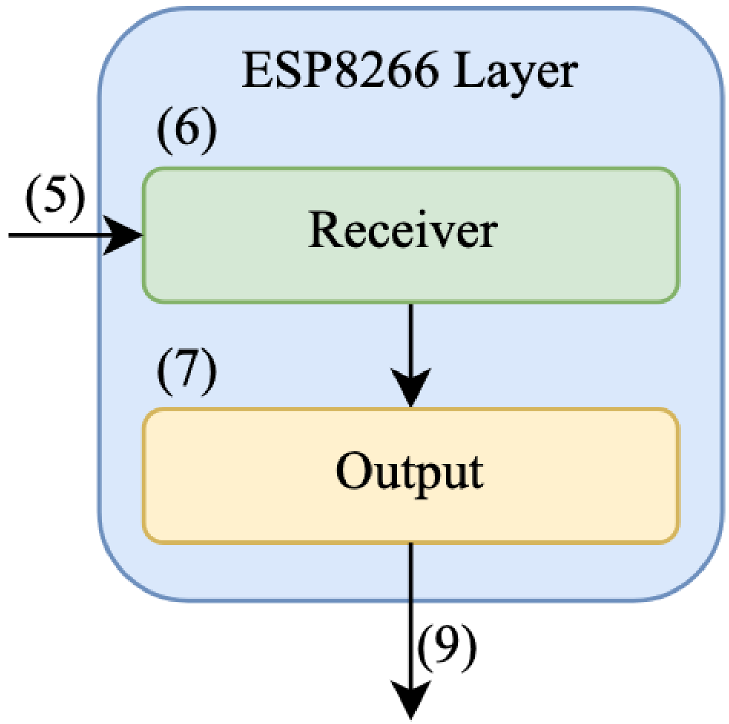

Figure 9: ESP8266 Subsystem 2

5.2.1 Assumptions

We are assuming here that the data coming from the receiver is already parsed and is in the correct format

5.2.2 Responsibilities

The Output must send the desired commands to the LED strips to properly change the color, brightness, or any other wanted state of the lights.

5.2.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

Table 8: Output Subsystem interfaces

ID	Description	Inputs	Outputs
#7	Incoming Parsed Data	Instruction Data	N/A
#9	Outgoing LED Commands	N/A	LED Commands

6 Hardware Layer Subsystems

In this hardware layer we have two subsystems including the power supply and the WS281B LED Lights. This layer is solely responsible for the LED lights and the lighting of them. The two arrows shown numbers 3 and 7 both come from the Raspberry Pi and ESP8266 layers respectively with commands on how the LEDs should be lit up.

6.1 Power Supply Subsystem

The power supply subsystem is what we have included to show the amount of power we will need to power the lights as this is one of the major components for the final product. The lights will need to be powered at fixed lengths to make sure that the lights receive ample power to stay lit the entire time. This subsystem communicates with the LED lights providing them power.

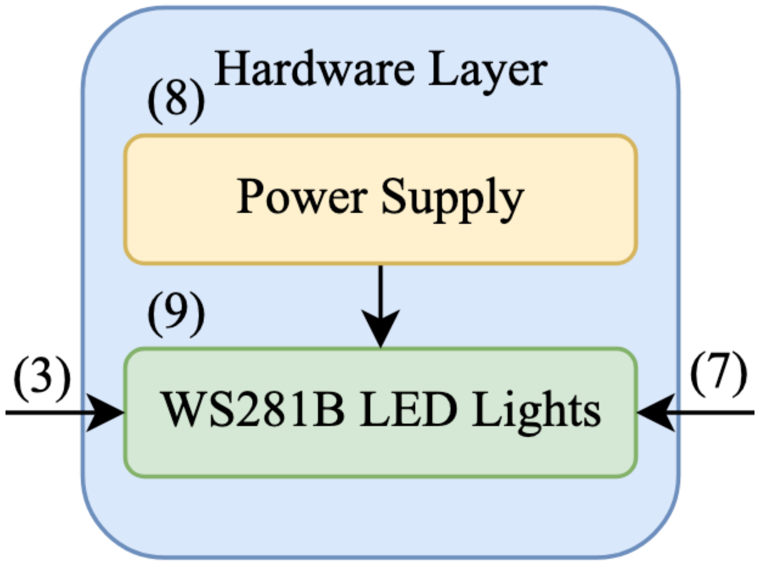

Figure 10: Hardware Subsystem 1

6.1.1 Assumptions

An ample amount of consistent power is provided at all times to the lights ensuring that they are lit at all times consistently and accurately the user needs them to be.

6.1.2 Responsibilities

The responsibility of this subsystem is simple, to provide power to the lights and making sure that the correct amount and enough power is provided at each interval needed for the lights. We plan to make it every 5 meters to keep the colors accurate for the entire length of the LED strip.

6.1.3 Subsystem Interfaces

The only inputs are power from a standard electrical socket and the output is power to the LED lights at the intervals.

Table 9: Power Supply Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Power Supply	Standard Electrical Socket(110 V)	Output to LED Lights

6.2 WS281B LED Lights

The WS281B subsystem simply contains the LED lights that the whole product is based upon. The lights receive power from the power supply subsystem and inputs on how to light up from the raspberry pi and ESP2866 layers.

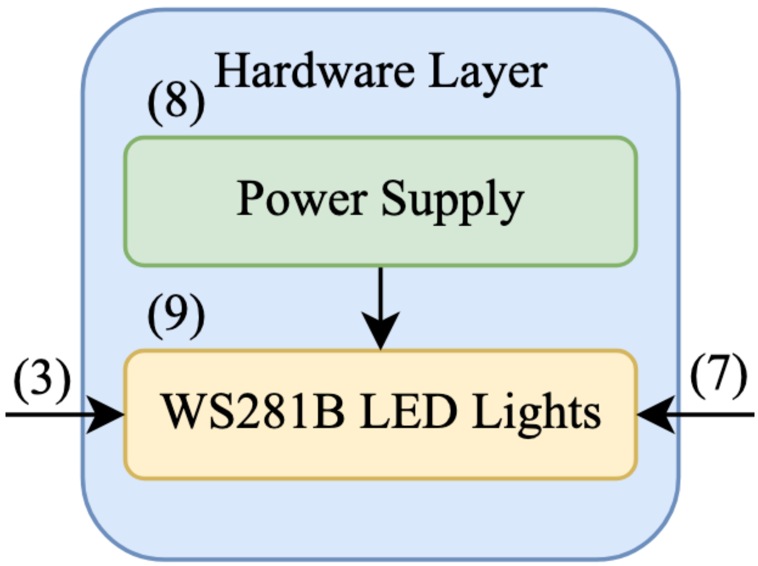

Figure 11: Hardware Subsystem 2

6.2.1 Assumptions

An ample amount of consistent power is provided at all times to the lights ensuring that they are lit at all times consistently and accurately as well as they will have a command set from the raspberry pi and ESP2866 layers on the pattern they are supposed to display.

6.2.2 Responsibilities

The responsibility of this subsystem is to carry out the pattern that the user would like, may it be a specific pre-saved pattern from the raspberry pi layer or specific LED lights to be given smart capabilities that are not connected to the Pi layer but the ESP8266 layer. The LED lights should display the lights with the same intensity throughout the usage.

6.2.3 Subsystem Interfaces

The inputs are from the power supply subsystem within the hardware layer, and from the patterns saved subsystem from the raspberry pi layer and the output subsystem from the ESP8266 layer. The outputs for all are the display of the LED lights that are supposed to run.

Table 10: WS281B LED Lights Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Power	Power Supply	LED Lights Display
#2	Saved Patterns	Saved patterns in Pi database	LED Lights Display
#3	Smart Lights	ESP2866	LED Lights Display

REFERENCES

Gieser, Shawn. "Computer System Design Project I", University of Texas at Arlington.