System Requirements Specification

for

Web-controlled LED light-suit

**Version 1.4**

**Created by Mike Schoder**

**FHNW**

**15.10.2019**

**Table of Contents**

[1. Introduction 3](#_Toc23261846)

[1.1 Purpose 3](#_Toc23261847)

[1.2 Intended Audience and Reading Suggestions 3](#_Toc23261848)

[1.3 Project Scope 3](#_Toc23261849)

[2. Overall Description 4](#_Toc23261850)

[2.1 Product Perspective 4](#_Toc23261851)

[2.2 Product Features 4](#_Toc23261852)

[2.2.1 GUI (Graphical User Interface) 4](#_Toc23261853)

[2.2.2 Communication 5](#_Toc23261854)

[2.2.3 Customization 5](#_Toc23261855)

[2.3 Operating Environment 5](#_Toc23261856)

[2.4 Design and Implementation Constraints 6](#_Toc23261857)

[2.5 User Documentation 6](#_Toc23261858)

[3. System Features 7](#_Toc23261859)

[3.1 GUI (Graphical User Interface) 7](#_Toc23261860)

[3.2 Communication 7](#_Toc23261861)

[4. Other Nonfunctional Requirements 8](#_Toc23261862)

[4.1 Safety Requirements 8](#_Toc23261863)

[4.2 Security Requirements 8](#_Toc23261864)

[4.3 Software Quality Attributes 8](#_Toc23261865)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Mike Schoder | 15.10.2019 | Creation of the document | 1.0 |
| Mike Schoder | 22.10.2019 | Supplement of div. Requirements | 1.1 |
| Mike Schoder | 29.10.2019 | Completion of the first Draft | 1.2 |
| Mike Schoder | 05.11.2019 | Corrections caused by the feedback | 1.3 |
| Mike Schoder | 12.11.2019 | Corrections based on feedback | 1.4 |

# Introduction

## Purpose

This Document was created to give a product specification over a new product. This product is a web-controlled LED light suit. Main purpose of this suit is to look interesting, with different animations and adjustable brightness. The components of the suit should be as function-based and limited as possible, likewise with microcontrollers. Those microcontrollers should be able to communicate with each other over a Wi-Fi, which will be hosted from a more capable and efficient microcomputer. The communication should be broadcasted from the microcomputer, so there can be multiple suits running the same animation at once.

## Intended Audience and Reading Suggestions

This Document is intended for different audiences. First, it is for the technical engineers who have to implement the product into a real-life possibility. Second, it is for the readers who want to know where the origin of the project comes from and what was specified primarily. Those people are also the testers, as they will try and copy the project and maybe run into some errors on their path to success.

## Project Scope

The project should be finished by end of the year, as the suit can be used as a carnival-suit. While outside, the suit should be powered with batteries, as the people who are using it want to walk around and do not want to hang on a power strip.

The goal is to stand out of other people who do not have as nice clothes as you have, and while working on the project, the engineer should learn a lot new stuff and document it, to improve the overall skills.

# Overall Description

## Product Perspective

This product is a standalone project. It is a new product for people who want to stand out of the masses of people at festivals, carnivals or everyday life.

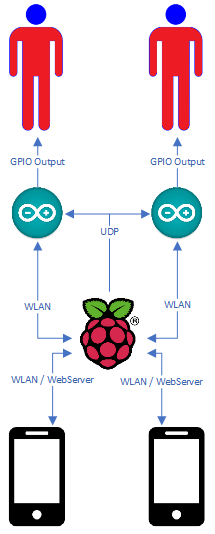


Figure : Product Overview of the Product with Interfaces and used Protocols

As the picture above shows us, we want to host a webserver on a raspberry-Pi, over which the Arduino’s get the commands to switch the current animation. The webserver is available only on the Wi-Fi hosted directly on the PI. This Wi-Fi is hosted without a Broadcast, so with scanning for available Wi-Fi’s, it will not show up. Connections can only be made if you know the Wi-Fi, so it is relatively safe. (when you do not know something exists, you will not try to get in). Once connected, you get to open the browser on the device you just connected. There, type in the IP of the Raspberry, and you get a Website with many Buttons. By a push of a Button, the Raspberry sends a UDP-Packet to the Arduino that includes a code, which animation now should be played instead of the current.

## Product Features

### 2.2.1 GUI (Graphical User Interface)

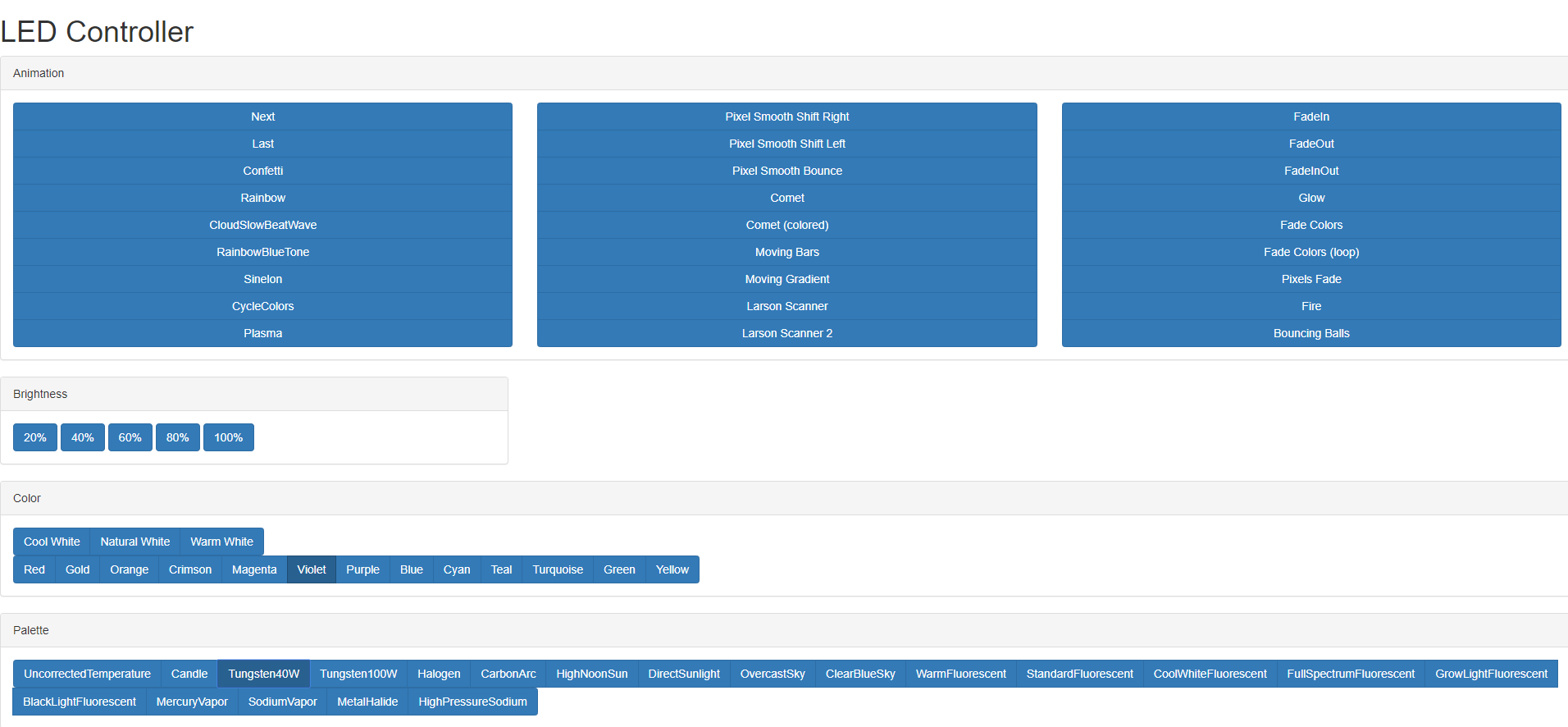


Figure : A sample GUI of the Webserver, divided into animations, brightness, color and color Palette

### 2.2.2 Communication

A major feature of the product is its communication between the devices, which allows fast changes on the visible part, the suit itself.

### 2.2.3 Customization

A significant function of the product is its customization. By using the GUI and the communication, you can rapidly change your look and react to outside influences. The suit’s Strobe-Effect gives someone epileptic seizures? Turn the brightness down or change the animation instantly to something more smoothly. The suit gathers too much attention? Turn it off instantly without damaging something, by just one click on a button.

## Operating Environment

The System will operate in following conditions:

|  |  |  |  |
| --- | --- | --- | --- |
| Category | Name | Quantity | Special |
| Microcontroller | Raspberry PI (newest Raspbian by 22.10.2019) | 1 | Installed Flask Webserver for Python 3 |
| Microcontroller | Arduino NodeMCU 1.0 with built-in ESP8266 | 1 | Installed Library is FastLED 3.0 |
| End-Device | Does not matter; as example a mobile with Android 9 | 1 | Fitted with a webserver |
| LED | NeoPixel Strip of LED | 5m | Quantity depends on the size of the Cloth you want to fit it on |
| Cloth | Plain black Suit (could be an Overall, a full body suit or anything) | 1 |  |

## Design and Implementation Constraints

The Developer of the product must use the specifications defined in the previous chapter, he is ought to use UDP as the communication module between the webserver and the Arduino, and is free in choosing and designing his own GUI.

**Important**: The Arduino, which is a single core processor, must be capable of listening to incoming commands, while it is showing an animation on the suit at the same time.

## User Documentation

With the product, there should come a user manual, a progress report, a documentation how it is implemented, and a tutorial that shows how optional suits can be added to show the same animations at the same time.

# System Features

## GUI (Graphical User Interface)

* + 1. Description and Priority

The GUI has a high priority, as it is the interface between a user, who probably has not the same technical know-how as the developer, and the system itself. It should be self-explanatory. What happens when I press this button? How can I achieve my goal? The cost of this feature is rated as low, as in this document there is already a template that could be used.

* + 1. Stimulus/Response Sequences

A User connects to the Wi-Fi of the Raspberry PI. The user types in the IP of the Raspberry PI. The User sees the GUI. The User wants to change the animation and presses a button in the GUI. The underlying Flask-webserver parses the button-press to an UDP-packet and broadcasts it in his net. All listening Arduinos pick up the UDP-packet and play the animation chosen by the user.

## Communication

* + 1. Description and Priority

The communication has a high priority, as it is the interface between the GUI and the suit itself. when there is no communication, the Arduino gets no commands from the user and does not know when the user wants something new. The cost of this feature is rated low, as it can be implemented relatively effortlessly. It can be accomplished over the Flask-webserver with a simple WebSocket over Python3.

3.2.2 Stimulus/Response Sequences

By using a button on the GUI, a command in form of an integer is sent from the Raspberry PI to the broadcast address of the Network. The Arduino gets this command and chooses the animation according to the integer. This means, the Arduino always has to listen to not skip a command.

# Other Nonfunctional Requirements

## Safety Requirements

For safety reasons, the common regulations in the handling with electrical appliances must be used, and everything must be insulated. The battery must be checked regularly.

Possible damage when handling electrical appliances may be burns or electric shocks.

## Security Requirements

For security reasons, the Wi-Fi cannot be publicly accessible, else someone could connect to the net and change your currently running animation.

The PI itself must be password-protected, else everybody who got access to the net of the PI could connect to the PI and alter settings, as example lock the current user out and change the login.

## Software Quality Attributes

Software quality is measured by the time it takes for changing a running animation. If it takes longer, or does not respond, something in the communication or the calculation from the animation is not optimized. If the animation is instantly changed by the click of a button, all components are matched to each other.

Software quality also is measured in the flexibility of the product code, if a user is able to add a new animation effortlessly, the software quality is rated high, as the code itself is easy to understand.