

DHBW Karlsruhe

TINF12B5

Studienarbeit

**Entwicklung eines Komplettsystems zur
Überwachung und Beleuchtung von Innen-
und Außenbereichen mit Raspberry Pi und
iOS App**

Autor:
Timo Höting
2185611

Betreuer:
Stefan Lehmann

Erklärung

Gemäß §5(3) der „Studien- und Prüfungsordnung DHBW Technik“ vom 22. September 2011.

Ich habe die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Ort, Datum

Unterschrift

Inhaltsverzeichnis

1	Einleitung	5
1.1	Vorwort	5
1.2	Projektbeschreibung	5
1.3	Teilprojekte	5
2	Hauptteil	6
2.1	LED-Pixel	6
2.1.1	Bewertungskriterien	6
2.1.2	Evaluiierung	6
2.1.3	Teststellung	7
2.2	Bewegungssensor	9
2.2.1	Bewertungskriterien	9
2.2.2	Evaluiierung	9
2.2.3	Fazit	10
2.2.4	Teststellung	11
2.3	Python-Server und Protokoll	11
2.3.1	Protokoll	11
2.3.2	Framework	11
2.3.3	Testcode	11
2.3.4	Implementierung	11
2.3.5	Klassen und ihre Funktionen	11
2.3.6	Hashfunktion	11
2.4	Verschlüsselung	11
2.4.1	SSL vs. TLS	11
2.4.2	Vor- und Nachteile TLS	11
2.4.3	TLS Handshake	11
2.4.4	Zertifikat und Key	11
2.4.5	Beispielcode Server	11
2.4.6	Wireshark Trace	11
2.5	Kamera	11
2.5.1	PI-Kamera vs. Netzwerkkamera	11
2.5.2	Ansteuerung	11
2.6	iOS App	11
2.6.1	Konzept	11
2.6.2	11
2.6.3	11
3	Praktische Umsetzung	12
4	Kostenaufstellung	13

5 Fazit

14

1 Einleitung

Es soll ein Komplettsystem entwickelt werden, dass sowohl die Überwachung als auch die Steuerung der Beleuchtung von Innen- und Außenbereichen ermöglicht. Das System soll nach Entwicklung universell einsetzbar und leicht konfigurierbar sein.

1.1 Vorwort

1.2 Projektbeschreibung

Für die Beleuchtung sollen adressierbare LED-Pixel eingesetzt werden, welche möglichst leicht in ihrer Anzahl variiert werden können. Es müssen passende Bauteile und Produkte evaluiert und getestet werden. Diese müssen vom Raspberry Pi ansteuerbar sein. Die Überwachung findet über eine Kamera statt. Ob diese direkt am Raspberry Pi angeschlossen wird oder sich nur im selben Netzwerk befindet wird im Laufe dieses Projekts erarbeitet. Für die Erkennung von Aktivitäten werden Bewegungsmelder eingesetzt. Gesteuert wird das System über einen Raspberry Pi. Von diesem aus werden die LEDs angesteuert, die Sensorsignale ausgewertet und die Befehle der App empfangen. Um dem User eine einfach Ansteuerung zu ermöglichen wird eine iOS App implementiert. Die hierfür genutzten Sprachen sind Swift und Objective-C. Die Serverfunktionalitäten werden in Python implementiert. Es gibt drei verschiedene Modi in denen sich das System befinden kann:

- Beleuchtung wird durch Bewegungsmelder ausgelöst (Reaktion darauf kann vom User definiert werden)
- Beleuchtung wird manuell vom Benutzer über App gesteuert (Color Chooser, Bereichsauswahl, Leuchteffekte)
- Bewegungsmelder als Alarmanlage, beim Auslösen wird der Benutzer benachrichtigt und Bild der Kamera als Notification auf dem Smartphone angezeigt

1.3 Teilprojekte

- LED-Pixel evaluieren / ansteuern
- Implementierung der Ansteuerung / des Protokolls (mit den drei verschiedenen Modi)
- Implementierung der iOS App
- Vollständige praktische Umsetzung an einem Beispielobjekt

2 Hauptteil

2.1 LED-Pixel

2.1.1 Bewertungskriterien

Die Beleuchtung soll durch einzelne LED-Pixel stattfinden. Ein Pixel bedeutet ein Chip auf dem sowohl die LED und der nötige Treiber sitzt. Für die Evaluierung werden folgende Kriterien gewählt:

- RGB-Farbraum
Die LED muss den gesamten RGB-Farbraum darstellen können.
Gewichtung: 5, KO-Kriterium
- Ansteuerung
Da der Raspberry Pi an einigen seiner Pins Pulsweitenmodulation (PWM) bietet, sollten die LED-Pixel ohne extra Hardware ansteuerbar sein. Eine extra Stromversorgung ist aber bei größerer Anzahl an LEDs unabdingbar.
Gewichtung: 10
- Framework
Hier wird bewertet ob der jeweilige Hersteller ein fertiges Framework zu seinen Produkten anbietet.
Gewichtung: 10
- Kosten
Es werden nur die reinen Produktkosten, also ohne Versand und Zoll, bewertet.
Gewichtung: 5
- Extras
An dieser Stelle können mögliche Extras eines Herstellers einfließen.
Gewichtung: 5

2.1.2 Evaluierung

- Adafruit, Neopixel
<https://www.adafruit.com/neopixel>
LED-Pixel in unzähligen Ausführungen.
Sitz der Firma in Tampa, Florida, USA
RGB: Chip ist der WS2801, <http://www.adafruit.com/datasheets/WS2801.pdf> ->
Hat volle Abdeckung des RGB-Farbraums
Ansteuerung: Findet über PWM-Pin des Raspberry Pi statt.
Framework: Framework von Adafruit, welches eine sehr leichte Ansteuerung ermöglichen soll.
Kosten: 4 LEDs 7\$, 25 LEDs zusammen 39\$, durch Lieferung aus USA sehr hohe

Versandkosten (50\$)

Extras: Händler bietet verschiedene Formen und fertige Ketten an.

- LED-Emotion GMBH, LED Streifen
<http://www.led-emotion.de/de/LED-Streifen-Set.html>
LED-Streifen, keine Einzelpixel, nur mit Controller, keine API
RGB: Voller RGB-Farbraum
Ansteuerung: Nur mit Controller
Framework: Keine öffentliche Api, möglicherweise mit Raspberry Pi ansteuerbar
Kosten: 30 LEDs mit Netzteil 79€
Extras: keine
- DMX4ALL GmbH, MagiarLED Solutions
<http://www.dmx4all.de/magiar.html>
Spezialisiert auf DMX-Ansteuerung, keine öffentliche API
RGB: Voller RGB-Farbraum
Ansteuerung: Wird über DMX-Controller angesteuert, dieser setzt die Signale um.
Vermutlich wird auch der WS2801 Chip verwendet.
Framework: DMX-Ansteuerung über DMX-Controller
Kosten: Streifen mit 72 LEDs = 99€
Extras: viele verschiedene Varianten
- TinkerForge, RGB LED-Pixel
<https://www.tinkerforge.com/de/shop/accessories/leds.html>
Scheinen die gleichen wie von Adafruit zu sein, allerdings werden hauptsächlich Controller im Shop angeboten
RGB: Chip ist der WS2801, <http://www.adafruit.com/datasheets/WS2801.pdf> -> Hat volle Abdeckung des RGB-Farbraums
Ansteuerung: Nach Anfrage an den Anbieter sollen die LEDs baugleich zu denen von Adafruit sein.
Framework: keins
Kosten: 50 LEDs = 59€
Extras: Lieferung aus Deutschland

Fazit: In der Evaluierung schneiden die Produkte von Adafruit und TinkerForge am besten ab. Für eine erste Teststellung werden die einzelnen LED-Pixel von Adafruit aus den USA bestellt (Neopixel). An diesen soll vor allem die Ansteuerung getestet werden. Falls diese sich bewähren wird für den endgültigen Aufbau auf die LED-Ketten von Tinkerforge zurück gegriffen.

2.1.3 Teststellung

Für einen ersten Test wurde das in XXX ausgewählte Produkt als einzelne Pixel bestellt. Der Hersteller Adafruit bietet hier 4er-Packungen an. Diese können leicht in eigene Schaltungen eingelötet oder auf Experimentier-Boards gesteckt werden. Bei geringer Anzahl LEDs reicht die 5V-Stromversorgung des Raspberry Pi aus.

Technische Daten Neopixel:

- Maße: 10.2mm x 12.7mm x 2.5mm
- Protokollgeschwindigkeit: 800 KHz
- Spannung: 5-9VDC (bei 3,5V gedimmte Helligkeit)
- Strom: 18,5mA / LED, 55mA / Pixel

Framework:

- RPI_WS281X (https://github.com/jgarff/rpi_ws281x)
- Sprache: Python
- Entwickelt für Raspberry Pi
- Voraussetzung: Python 2.7

Ablauf des Tests:

- **Aufbau der Schaltung**

An die einzelnen LED-Pixel wurden Stecker angelötet, damit sie auf das Experimentierboard aufgesteckt werden können. Dann wird die Schaltung nach folgendem Schaltbild verbunden. Wichtig ist, dass beim Raspberry Pi nur Pins verwendet werden können, welche PWM bieten.

- **Installation des Frameworks**

```
1 wget https://github.com/tdicola/rpi_ws281x/raw/master/python/dist/  
   rpi_ws281x-1.0.0-py2.7-linux-armv6l.egg  
2 sudo easy_install rpi_ws281x-1.0.0-py2.7-linux-armv6l.egg
```

Listing 2.1: Installation Framework ws281x

- **Testcode**

```
1 from neopixel import *  
2  
3 LED_COUNT = 4 # Number of LED pixels.  
4 LED_PIN = 18 # GPIO pin connected to the pixels (must support PWM!).  
5 LED_FREQ_HZ = 800000 # LED signal frequency in hertz (usually 800khz)  
6 LED_DMA = 5 # DMA channel to use for generating signal (try 5)  
7 LED_INVERT = False # True to invert the signal (when using NPN)  
8  
9 strip = Adafruit_NeoPixel(LED_COUNT, LED_PIN, LED_FREQ_HZ,  
   LED_DMA, LED_INVERT)  
10  
11 strip.begin()  
12 strip.setPixelColor(0, Color(255, 255, 255))  
13 strip.setPixelColor(1, Color(255, 255, 255))  
14 strip.setPixelColor(2, Color(255, 255, 255))
```



```
15 strip.setPixelColor(3, Color(255, 255, 255))  
16 strip.show()
```

Listing 2.2: Testcode zur Ansteuerung der LEDs

2.1.4 Fazit

Die einzelnen Pixel sind sehr leicht anzusteuern, unterstützen auch das automatische Abschalten nach einer bestimmten Zeit und haben eine sehr hohe Leuchtkraft. Die Evaluierung hat zu einer guten Produktwahl geführt. In der Endgültigen Projektstellung werden keine einzelnen LEDs eingesetzt, sondern eine fertige Kette des Herstellers.

2.2 Bewegungssensor

In einem der Modi soll die Beleuchtung durch den Bewegungsmelder ausgelöst werden. Hierfür sind zuverlässige und weitreichende Bewegungssensoren notwendig.

2.2.1 Bewertungskriterien

- **Ansteuerung**
Die Anbindung an den Raspberry Pi soll möglichst leicht realisierbar sein. Wünschenswert ist, dass der Sensor einfach ein High-Signal bei Bewegungserkennung ausgibt.
Gewichtung: 5, KO-Kriterium
- **Reichweite**
Die Reichweite oder Sensivität des Sensors soll ausreichend und regelbar sein.
Gewichtung: 3
- **Kosten**
Es werden nur die reinen Produktkosten, also ohne Versand und Zoll, bewertet.
Gewichtung: 1
- **Extras**
An dieser Stelle können mögliche Extras eines Herstellers einfließen.
Gewichtung: 3

2.2.2 Evaluierung

- **PIR (MOTION) Sensor, Adafruit**
Link: <http://www.adafruit.com/product/189>
Ansteuerung: Gibt High-Signal an einem Pin aus.
Reichweite: 7m, 120 Grad
Kosten: 9,95\$ + Versand aus USA
Extras: Kabel inklusive
- **PIR Infrared Motion Sensor (HC-SR501)**
Link: <https://www.modmypi.com/pir-motion-sensor>

Ansteuerung: Gibt High-Signal an einem Pin aus.
Reichweite: 5-7m, 100 Grad
Kosten: 2,99\$ + Versand aus UK
Extras: keine

- **Infrarot PIR Bewegung Sensor Detektor Modul**

Link: <http://www.amazon.de/Pyroelectrische-Infrarot-Bewegung-Sensor-Detektor/dp/B008AESI>
Ansteuerung: Gibt High-Signal an einem Pin aus.
Reichweite: 7m, 100 Grad
Kosten: 5 Stück = 7,66€
Extras: keine

2.2.3 Fazit

Die meisten Infrarot-Bewegungssensoren sind von der Bauweise nahezu identisch. Die Unterschiede liegen meist nur in der Empfindlichkeit. Da die Reichweite in diesem Fall nicht von großer Bedeutsamkeit ist, kann eigentlich jedes der Produkte bestellt werden. Auf Ebay und Amazon ist die Anzahl angebotener Sensoren nahezu unbegrenzt, es wurde für die Teststellung also die oben evaluierte Variante von Amazon bestellt.

2.2.4 Teststellung

2.3 Python-Server und Protokoll

2.3.1 Protokoll

2.3.2 Framework

2.3.3 Testcode

2.3.4 Implementierung

2.3.5 Klassen und ihre Funktionen

2.3.6 Hashfunktion

2.4 Verschlüsselung

2.4.1 SSL vs. TLS

2.4.2 Vor- und Nachteile TLS

2.4.3 TLS Handshake

2.4.4 Zertifikat und Key

2.4.5 Beispielcode Server

2.4.6 Wireshark Trace

2.5 Kamera

2.5.1 PI-Kamera vs. Netzwerkkamera

2.5.2 Ansteuerung

2.6 iOS App

2.6.1 Konzept

2.6.2 ...

2.6.3 ...

3 Praktische Umsetzung

4 Kostenaufstellung

5 Fazit