



# Diplomarbeit

Höhere Technische Bundeslehranstalt Leonding  
Abteilung für Informatik

## Mesh Netzwerk

Eingereicht von: **Tim Untersberger, 5BHIF**  
**Stefan Waldl, 5BHIF**

Datum: **4. April 2020**

Betreuer: **Thomas Stuetz**

Projektpartner: **HTBLA Leonding**

## **Eidesstattliche Erklärung**

Hiermit erkläre ich an Eides statt, dass ich die vorgelegte Diplomarbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Gedanken, die aus fremden Quellen direkt oder indirekt übernommen wurden, sind als solche gekennzeichnet.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Leonding, am 4. April 2020

Tim Untersberger, Stefan Waldl

## **Declaration of Academic Honesty**

Hereby, I declare that I have composed the presented paper independently on my own and without any other resources than the ones indicated. All thoughts taken directly or indirectly from external sources are properly denoted as such.

This paper has neither been previously submitted to another authority nor has it been published yet.

Leonding, April 4, 2020

Tim Untersberger, Stefan Waldl

## **Zusammenfassung**

Zusammenfassung in Deutsch + Bild am Ende

## **Abstract**

Zusammenfassung in Englisch + Bild am Ende

## **Autoren der Diplomarbeit**

## **Danksagung**

If you feel like saying thanks to your grandma and/or other relatives.

# Inhaltsverzeichnis

<b>1</b>	<b>IstZustand</b>	<b>3</b>
<b>2</b>	<b>SollZustand</b>	<b>4</b>
<b>3</b>	<b>Verwendete Technologien</b>	<b>5</b>
3.1	Over The Air Update . . . . .	5
3.2	Mesh Netzwerk . . . . .	6
3.3	Nodejs . . . . .	6
3.4	Platform IO . . . . .	6
3.5	Docker . . . . .	6
3.6	Docker Compose . . . . .	6
3.7	ESP IDF Utility lib . . . . .	6
3.8	React . . . . .	6
3.9	Yarn . . . . .	6
3.10	Webpack . . . . .	6
<b>4</b>	<b>Ausgewählte Aspekte</b>	<b>7</b>
4.1	NGINX sichern mit letsencrypt . . . . .	7
4.2	EspWifiManager Implementation . . . . .	7
4.3	Die Wichtigkeit von Erase Flash . . . . .	7
4.4	ELF vs Bin . . . . .	7
<b>5</b>	<b>Summary</b>	<b>8</b>
<b>A</b>	<b>Additional Information</b>	<b>13</b>
<b>B</b>	<b>Individual Goals</b>	<b>14</b>

## Kapitel 1

# IstZustand

## Kapitel 2

# SollZustand



## Kapitel 3

# Verwendete Technologien

### 3.1 Over The Air Update

#### Problemstellung

Der Workflow mit Mikrocontroller in Produktivsystemen ist aufwändig, da sie sich meist an ungünstigen Orten befinden. Bis jetzt musste man seine Datenquelle physisch mit dem Chip verbinden um neue Firmware auf den Mikrocontroller zu spielen. OTA ermöglicht es den Mikrocontroller mit neuer Firmware zu versorgen, dabei muss der ausgewählte Microcomputer lediglich mit einem Wlan-Router verbunden sein.

#### Wie OTA funktioniert

Am wichtigsten ist es, dass man die Partitionstabelle an das ausgewählte Programm anpasst. Die Standard-Partitionstabelle ist je nach Hersteller anders. Um OTA zu ermöglichen ist es notwendig im Partitionstabelle mindestens eine, ausreichend große, Partition zu vergeben. Dabei ist es wichtig, dass die OTA-Partition ausreichend Speicher für die gewünschte Firmware hat.

Damit der OTA-Vorgang funktionieren kann, muss zuerst ein lauffähiges Programm auf dem Mikrocontroller gestartet sein. Danach muss z.B. der ESP32 die neue Firmware über die bestehende Internet oder Intranet-Verbindung in die zuvor definierten Partition herunterladen. Anschließend muss nur noch der ESP32 mit der aktualisierten Partition neu gestartet werden.

#### Implementation

Die ausgewählte Struktur für die Diplomarbeit kann man im folgenden Bild sehr gut sehen.

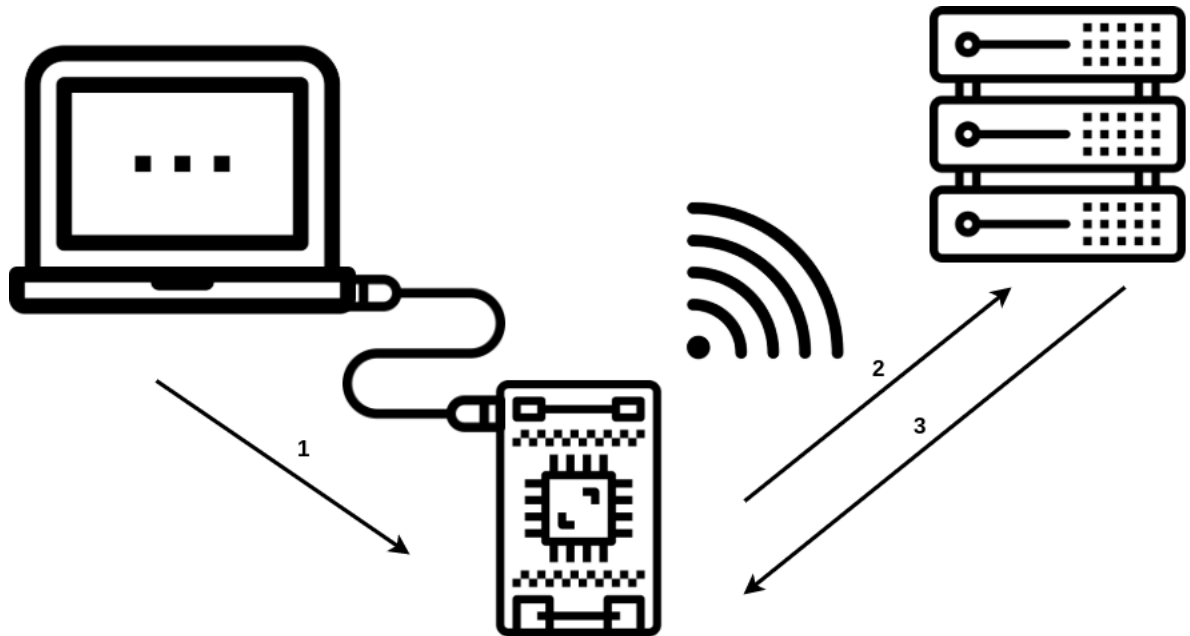


Abbildung 3.1: OTA Erklärung (Quelle: eigene Darstellung)

### 3.2 Mesh Netzwerk

### 3.3 Nodejs

### 3.4 Platform IO

### 3.5 Docker

### 3.6 Docker Compose

### 3.7 ESP IDF Utility lib

### 3.8 React

### 3.9 Yarn

### 3.10 Webpack

## Kapitel 4

# Ausgewählte Aspekte

- 4.1 NGINX sichern mit letsencrypt
- 4.2 EspWifiManager Implementation
- 4.3 Die Wichtigkeit von Erase Flash
- 4.4 ELF vs Bin

## Kapitel 5

### Summary

Here you give a summary of your results and experiences. You can add also some design alternatives you considered, but kicked out later. Furthermore you might have some ideas how to drive the work you accomplished in further directions.

# Literaturverzeichnis

# Abbildungsverzeichnis

3.1	OTA Erklärung (Quelle: eigene Darstellung) . . . . .	6
-----	--	---

# Tabellenverzeichnis

# Project Log Book

Date	Participants	Todos	Due
------	--------------	-------	-----



## Anhang A

# Additional Information

If needed the appendix is the place where additional information concerning your thesis goes. Examples could be:

- Source Code
- Test Protocols
- Project Proposal
- Project Plan
- Individual Goals
- ...

Again this has to be aligned with the supervisor.

## Anhang B

# Individual Goals

This is just another example to show what content could go into the appendix.