

# MQTT-Control-Panel

Fabian Schätzschock

Richard Krammer

13. November 2024

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>3</b>
<b>2</b>	<b>Konzept</b>	<b>4</b>
2.1	Software . . . . .	4
<b>3</b>	<b>Anforderungen</b>	<b>5</b>
3.1	Software . . . . .	5
3.1.1	MQTT . . . . .	5
3.1.2	Display . . . . .	5
3.1.3	UI-Library . . . . .	5
3.1.4	Input . . . . .	5
3.1.5	Display . . . . .	5
3.2	Hardware . . . . .	5
<b>4</b>	<b>Software</b>	<b>6</b>
4.1	Projektstruktur . . . . .	6
4.2	User Interface . . . . .	6
4.2.1	SquarelineStudio . . . . .	6
4.2.2	Touch Control . . . . .	7
4.3	Kommunikation . . . . .	7
4.3.1	MQTT . . . . .	7
4.3.2	ESP-NOW . . . . .	7
<b>5</b>	<b>Ergebnisse</b>	<b>8</b>

# 1 Einführung

## 2 Konzept

### 2.1 Software

Das Control-Panel soll mit dem Netzwerk eines Smarthomes verbunden werden und in der Lage sein, über den im Smarthome bereits integrierten MQTT-Broker mit integrierten Geräten zu kommunizieren und den Status dieser Geräte abzufragen und anzupassen.

## 3 Anforderungen

### 3.1 Software

Zur effizienten Realisierung des Panels sind zwei Hauptfunktionalitäten zu implementieren.

#### 3.1.1 MQTT

Um eine Kommunikation zu ermöglichen, ist eine MQTT-Library notwendig. Hierfür wurde "PubSubClient" von "knolleary" verwendet. Hierbei handelt es sich um eine simple Library, die nur das Nötigste implementiert, um einen überschaubaren Overhead zu gewährleisten.

#### 3.1.2 Display

Bei der Umsetzung des Displays sind zwei Funktionalitäten die es zu implementieren gilt. Zum einen die Anzeige von Informationen und zum anderen die Interaktion mit dem User Interface.

#### 3.1.3 UI-Library

Für die Ansteuerung des Displays wurde die "TFT\_eSPI" Library von "Bodmer" verwendet. Diese Library bietet Support für den am Display verwendeten Controller.

#### 3.1.4 Input

#### 3.1.5 Display

Bei der Umsetzung des Displays sind zwei Funktionalitäten die es zu implementieren gilt. Zum einen die Anzeige von Informationen und zum anderen die Interaktion mit dem User Interface.

### 3.2 Hardware

## 4 Software

### 4.1 Projektstruktur

Die grundlegende Struktur des Projekts folgt der von PlatformIO normalisierten Projektstruktur.

### 4.2 User Interface

#### 4.2.1 SquarelineStudio

SquarelineStudio ist eine Software, die es ermöglicht, ein User Interface mithilfe eines Drag-and-Drop-Editors zu erstellen und in C-Code zu exportieren. Dieser Code kann dann dann zusammen mit den beiden Libraries *TFT\_eSPI* und *lvgl* in PlatformIO integriert werden.

#### **TFT\_eSPI**

TFT\_eSPI ist eine Library für Grafik und Fonts auf einem TFT-Display. Sie ist mit vielen verschiedenen Controllern kompatibel und bietet viele Funktionen, um verschiedene TFT-Display anzusteuern. Sie ist eine Hälfte des Grundgerüsts für die Darstellung von Grafiken in SquarelineStudio.

#### **lvgl**

lvgl ist eine Library für die Darstellung von flexiblen Grafiken auf vielen Plattformen, darunter auch dem Arduino Framework. Zusammen mit TFT\_eSPI bildet sie das Grundgerüst für die Darstellung von Grafiken in SquarelineStudio.

#### 4.2.2 Touch Control

### 4.3 Kommunikation

#### 4.3.1 MQTT

#### 4.3.2 ESP-NOW

## 5 Ergebnisse