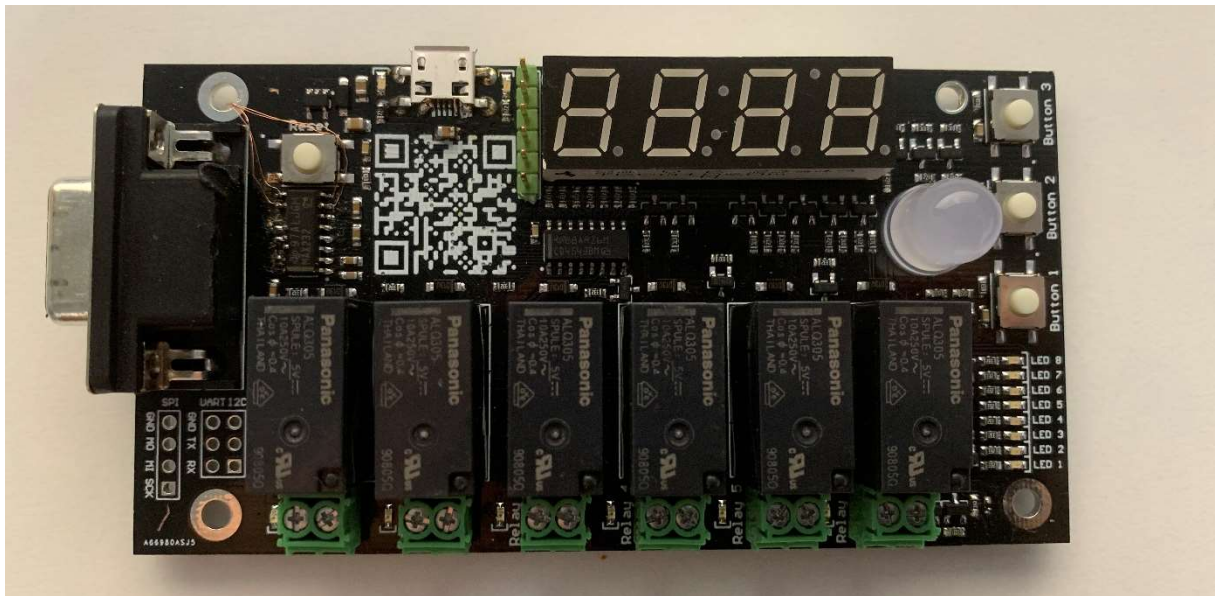


Erweiterung für das Node-Red

Gebäudeinformatiker mit Zertifikat SwissGIN / MMTS



Marco Stauber
Donnerstag 2020
TBZ Höhere Fachschule

Inhaltsverzeichnis

Einleitung	3
Zusatzinfos	3
Projekt	4
Erweiterungsboard	5
Control Panel.....	6
Master Control.....	6
Relais und LEDs	6
RGB LED	6
Taster/Knöpfe.....	6
7-Segment	6
Temperatur Sensor.....	6
Klimaanlage	7
Funktion.....	7
Dashboard	7
LED-Streifen	8
Funktion.....	8
Dashboard	8
Reflexion	9
Selbstständigkeitserklärung	10
Quellen	10
Anhang	11
Dashboards	11
Control Panel.....	11
Klimaanlage	12
LED-Streifen	13

Einleitung

Mein Ziel war es, Node-Red kennen zu lernen und es anzuwenden, sich die einzelnen Nodes anzuschauen und zu verstehen was diese machen.

Spezifisch wollte ich für mein selbst entwickeltes Erweiterungsboard eine Steuerung und eine Anwendung programmieren.

Zusatzinfos

Der folgende Link befindet sich auch auf dem Erweiterungsboard.

Der QR Code dazu befindet sich auf der Vorderseite.

Dort gibt es dafür die:

- Anleitung der Befehle
- Seriell Einstellungen
- Function-Node Beispiele
- Schema, Gerber und Stückliste des Erweiterungsboards
- Flows (Verfügbar ab Präsentation)

Alle Flows und Dashboard sind als Input Codes auf dem GitHub.

<https://github.com/MXACE/Edge-Device-Extension>

Die Bilder bei «Projekt/Klimaanlage» und «Projekt/LED-Streifen» dienen nur zur Darstellung. Die darin vorhandenen Objekte wurden mit Absicht nicht korrekt angeschlossen oder können sich noch bis zur Präsentation ändern, die Funktion bleibt aber die Gleiche.

Projekt

Ich wusste für eine lange Zeit nicht, was ich machen sollte. Irgendwann kam ich auf die Idee, eine Erweiterung des Edge Devices zu entwickeln, die mir mehr Möglichkeiten gibt, wie man Objekte anschliessen, steuern oder etwas darstellen kann. Diese Erweiterungen habe ich 5-mal herstellen lassen und diese an meine Kollegen aus dem Donnerstagskurs verkauft.

Nun ist das Edge Device auf der Hardwareseite erweitert. Für ein Projekt, dachte ich mir, wäre es doch interessant, wenn man ein Kontrollzentrum hat, um das Ganze zu steuern, zum Beispiel in Form eines Dashboards.

Da mir das noch nicht genug war, wollte ich zudem eine Funktion mit meinem Erweiterungsboard entwickeln, um darzustellen, dass dieses auch funktioniert.

In meinem Fall eine einfache Klimaanlage, die mit einem Temperatur Sensor arbeitet und einem Relais, um den Lüfter einzuschalten.

Das war mir aber immer noch zu wenig.

Als Option wäre es doch auch möglich wenn ich einen Arduino hätte, bei dem ein LED-Streifen angeschlossen ist. Das ganze steuere ich mit dem seriellen Anschluss der Erweiterung.

Erweiterungsboard

Da beim Edge Device die Anzahl an Hardwareanschlüssen ziemlich gering ist, habe ich meine eigene Printplatte entwickelt, die zahlreiche Erweiterungen und Anzeigemöglichkeiten anbietet.

Das PCB bietet:

6x Schaltbare Relais (Schliesser)

8x Grüne LEDs

1x RGB LED (7 verschiedene Farben)

1x 4-Stellige 7-Segment Anzeige

3x Knöpfe

1x I²C Anschluss

1x SPI Anschluss

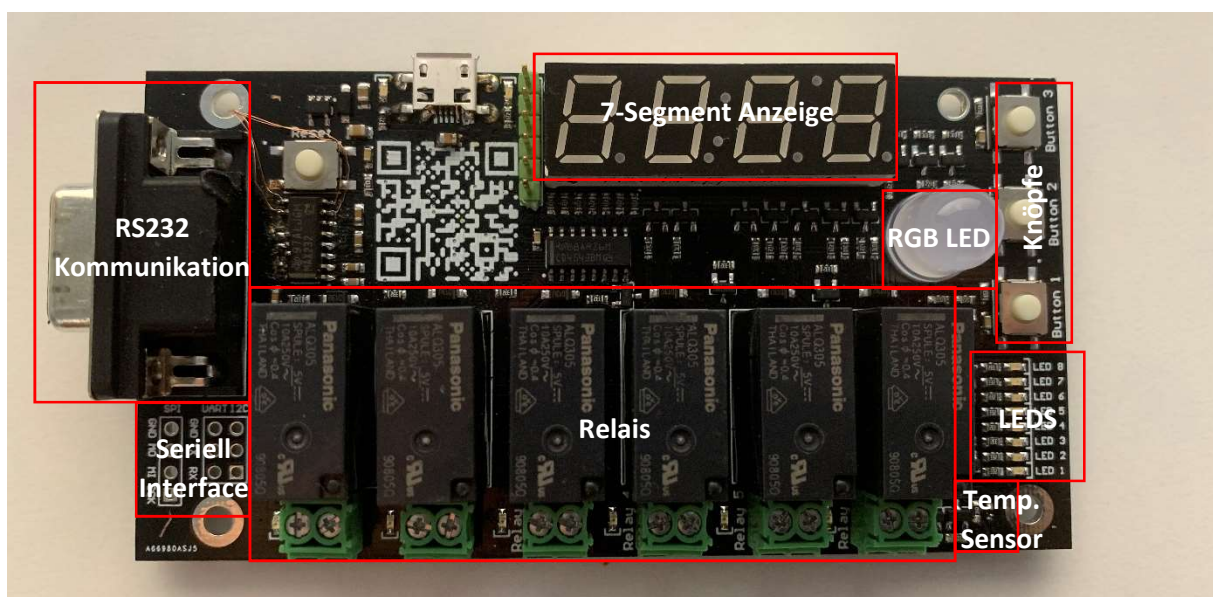
1x Uart Anschluss

Zwei der drei seriellen Anschlüsse (I²C & SPI) sind in der Software nicht ausprogrammiert, jedoch in der Hardware vorhanden.

Angesteuert wird das über den Seriell-Node und der RS232 Buchse am Edge Device. Alle Befehle, die für die Erweiterung benötigt werden, können in der Anleitung auf dem GitHub gefunden werden.

Das Board bekommt seine Speisung über den USB-Anschluss.

Die Befehle, Einstellungen und Beispiele für Function-Nodes findet man im GitHub im Ordner «Extension-Board».



Control Panel

Im Dashboard kann man jedes einzelne Modul des Erweiterungsboards ansteuern.

Master Control

Um andere Funktionen nicht zu unterbrechen oder sogar bis zum nächsten Software-neustart «kaputt» zu machen, gibt es einen Master-Switch.

Bevor dieser nicht aktiviert ist, lassen sich keine Knöpfe oder Schalter bedienen.

Die vorherigen Zustände der Schalter bleiben. Falls sich jedoch ein Modul geändert hat, übernimmt dies das Control Panel nicht.

Relais und LEDs

Mit Schaltern kann man alle Relais und LEDs ein- und ausschalten. Diese Relais könnten sogar 230V AC schalten, was man aber nicht machen sollte, da das Board nicht dafür designed wurde.

RGB LED

Die Farbe des RGB LEDs wird über einen Colorpicker ausgewählt. Dieser gibt mir einen HSL-Wert. Da ich die Helligkeit nicht einstellen kann, interessiert mich nur der Hue-Wert. In einem Function-Node überprüfe ich, welche die nächste Farbe ist, und setze das RGB LED auf die entsprechende Farbe.

Taster/Knöpfe

Die Knöpfe auf dem Board werden im Dashboard in Form von Schaltern dargestellt.

Da das Erweiterungsboard nur bei der negativen Flanke das Buttonsignal schickt, habe ich hier eine 200ms Verzögerung eingebaut damit man es besser sehen kann.

7-Segment

Die 7-Segment-Anzeige wird über ein Text-Input-Node mit der Konfiguration Number-input gesteuert. Man kann eine Ganzzahl hineinschreiben, welche nachher auf die ersten vier Stellen begrenzt wird.

Für den Doppelpunkt oder das Apostroph muss man die Schalter verwenden, da diese nicht im Text-Input akzeptiert werden.

Die Kommastelle wird über einen Schieberegler eingestellt. Die Zahl entspricht der Position des Kommas aus der Sicht der Einerstelle.

Wenn man alle seine Einstellungen gemacht hat, kann man auf den «SET» Knopf drücken und die 7-Segment-Anzeige zeigt das Eingebene an.

Temperatur Sensor

Die Temperatur wird alle fünf Sekunden gemessen und auf dem Dashboard als Text ausgegeben. Gleichzeitig wird der Wert in einem XY-Graph dargestellt, um den Verlauf anzusehen. Der Wert wird in einer globalen Variabel abgespeichert und läuft unabhängig vom Master-Switch.

Klimaanlage

Funktion

Über den Temperatursensor erhalte ich die Umgebungstemperatur des Erweiterungsboards. Ein Problem welches vorkommt ist, dass sich die Printtemperatur erhöht wenn andere Module angeschaltet sind, was auch den Sensor beeinflusst. Somit steigt die Temperatur bis zu etwa 32°C an.

Dieses Problem ist bei der Klimaanlage sogar sehr nützlich.

Sobald die Temperatur an einem gewissen Punkt ankommt, schaltet dies ein Relais an, an welchem ein Lüfter angeschlossen ist. Dieser Lüfter ist so ausgerichtet, dass er auf den Sensor bläst.

Wenn die Temperatur nun genug hoch ist, schaltet sich der Lüfter ein und kühlt den Sensor bis zu der Off-Schwelle hinunter. Danach beginnt das Ganze wieder von vorne.

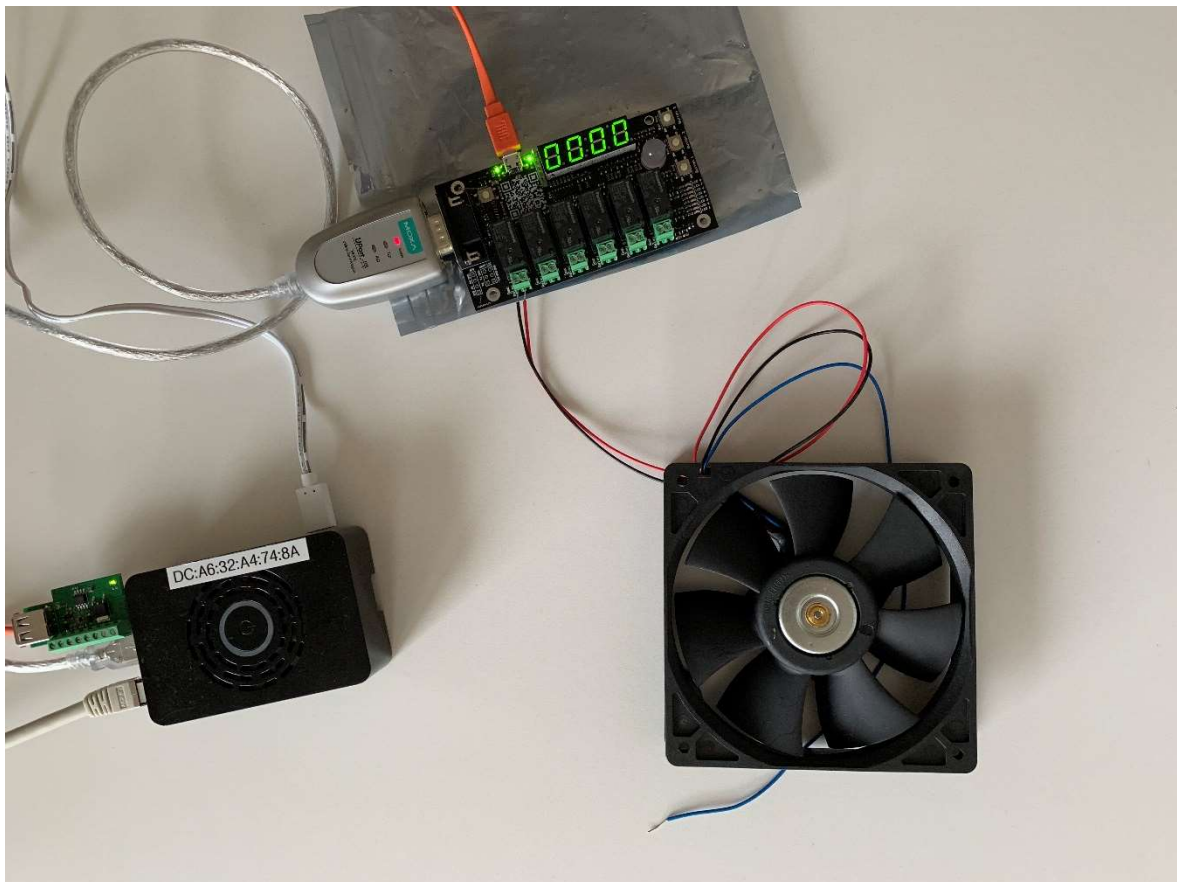
Gleichzeitig wird die aktuelle Temperatur auf der 7-Segment Anzeige dargestellt.

Dashboard

Im Dashboard kann man das Ganze über den Hauptschalter anschalten.

Man kann, wie im Control Panel, die Temperatur in einem Graph mitverfolgen und direkt mit einem Text Output-Node ablesen.

Ein Schalter gibt den Status des Lüfters an.



LED-Streifen

Funktion

Ein Arduino ist über die UART Schnittstelle angeschlossen. An diesem befindet sich ein Adafruit Neopixel LED Streifen. Über das Dashboard schaltet man den LED Streifen ein und aus und kann zwischen den vorgegebenen Szenen wechseln.

Dashboard

Über einen Slider bestimmt man welche Szene man haben will.

Schiebt man den Slider auf null oder auf das Maximum, stellt man den LED Streifen an oder aus.

Eine Textanzeige zeigt an, welche Szene gerade aktiv ist.



Reflexion

Persönlich habe ich von der Lehre als Elektroniker EFZ schon Erfahrung im Programmieren, also fällt es mir leicht, eine Lösung mit den Nodes zu finden.

Grösstenteils habe ich mit Function-Nodes gearbeitet, dabei fiel mir auf, dass die Grundfunktionen in JavaScript exakt gleich wie in C oder C++ sind. Eine Ausnahme sind die Variablen, bei denen ich nachschauen musste, wie man welche erstellt die Node oder Flow übergreifend sind, darunter auch die Globalen.

Ich wusste lange nicht was ich als Projekt machen wollte. In diesen Sinn hatte ich Glück mit der Quarantäne, die im März kam. Durch diese hatte ich mehr Zeit und konnte währenddessen mein Erweiterungsboard entwickelt.

Bei gewissen Nodes wie z.B. range oder switch habe ich Mühe, da deren Funktion für mich nicht besonders logisch ist. Bei range zum Beispiel, dass die Nummer nicht nur begrenzt, sondern auch noch skaliert wird.

Bei meinem Erweiterungsboard gab es auch endliche Komplikationen, da ich den Programmverlauf möglichst parallel gestalten musste. Eine der Herausforderungen war die Befehlsliste zu schreiben und diese umzusetzen.

Was mir auch noch auffiel ist, dass Node-Red standartmässig bei einer seriellen Kommunikation ein Echo zurückgibt. Das hatte die Folge, dass der Buffer auf dem Erweiterungsboard Overflows hatte und somit einen Hardfault-Fehler auslöste. Das heisst, es kam zu einem Bus-Fehler und das Programm blieb stehen.

Ein anderes Symptom war, dass die Knopfnummer auf der 7-Segment-Anzeige dargestellt wurde, obwohl es keinen Code dazu gibt, der dies auslösen könnte.

Das Node-Red ist eine sehr interessante Programmierplattform welche mit bis zu knappen 3000 Plugins erweitert werden kann. Darunter sehr nützliche, welche mir halfen, wie zum Beispiel die serielle Kommunikation oder das Dashboard.

Ich kann nun die Grundlagen von JavaScript sowie mit Node-Red umgehen und verstehen, was passiert.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich dieses Projekt selbstständig verfasst und keine anderen als die angegebenen Quellen benutzt habe.

Quellen

Node-Red, Writing Functions

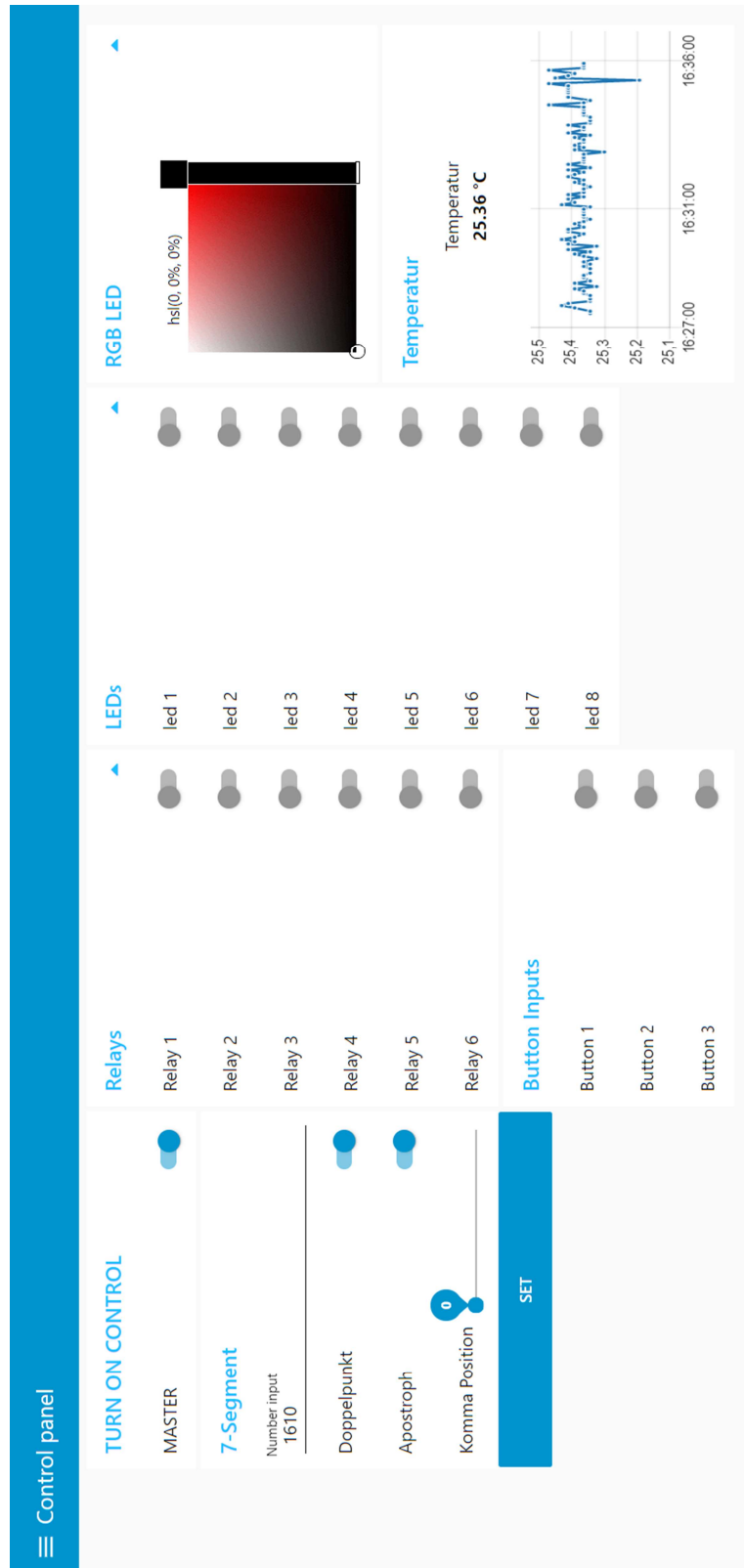
<https://nodered.org/docs/user-guide/writing-functions>

[Abrufdatum: 20.09.2020]

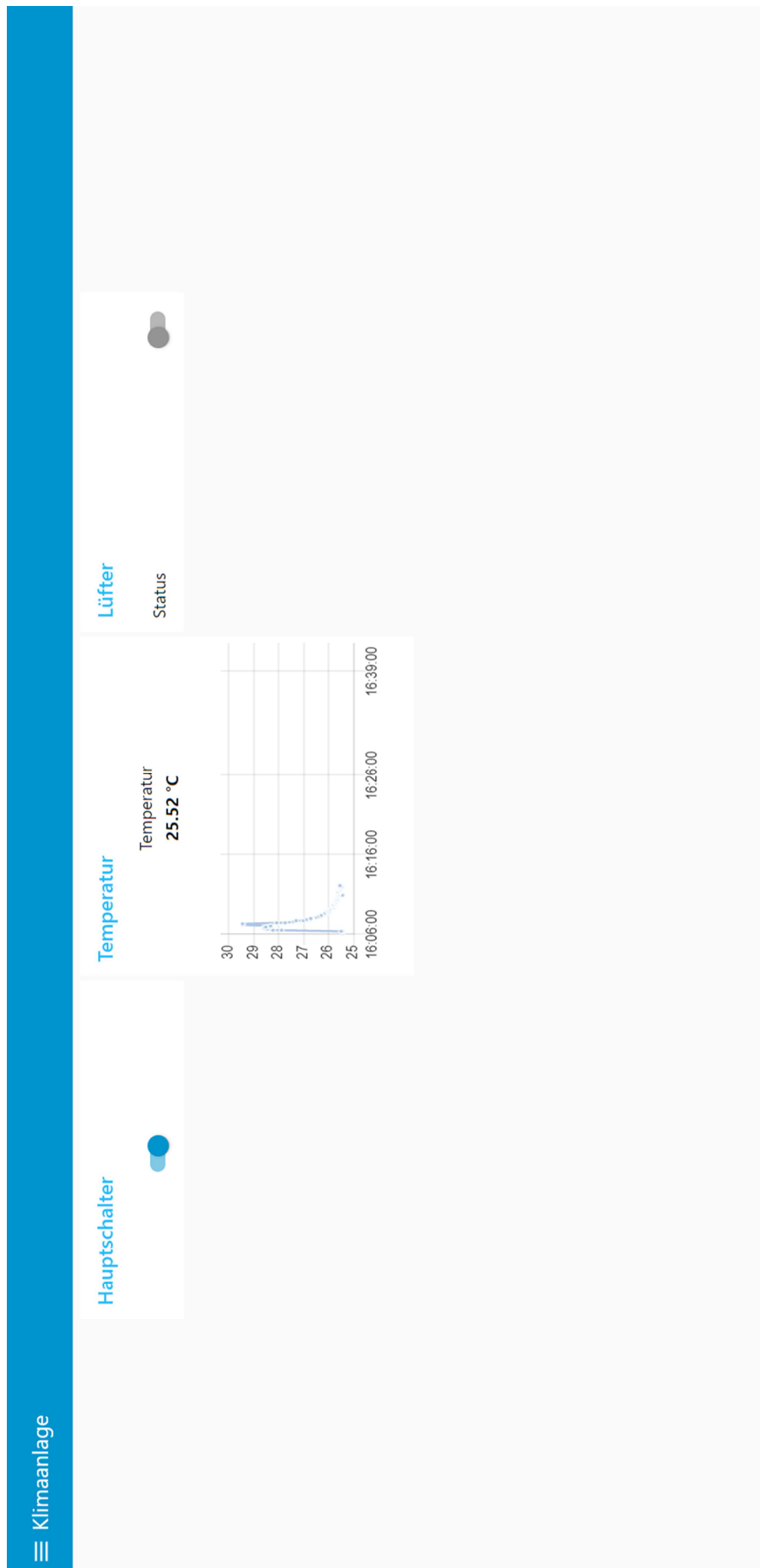
Anhang

Dashboards

Control Panel



Klimaanlage



LED-Streifen

