

Weekly Reports

Luftqualität in Innenräumen - Gruppe 1

31. Januar 2022

Name	Matrikel Nr.	Arbeitsaufwand (h)
Friedrich Just	1326699	22,00
Stipe Knez	1269206	20,00
Lucas Merkert	1326709	15,00
Achim Glaesmann	1309221	
Max-Rene Konieczka	1211092	20,00
Can Cihan Nazlier	1179244	19,00

Tabelle 1: Arbeitsaufwand dieser Woche

1 Überblick

1.1 Friedrich Just

Diese Woche wurde sich mit der TiKZ Bibliothek auseinandergesetzt, um für die Dokumentation ein State-Diagramm zu zeichnen. Danach wurde recherchiert, wie man die Eingangs Spannung messen kann. Hier wurde in dem AT-MEL256RFR2 Datenblatt[1] zwei Möglichkeiten gefunden. Der sogenannten Battery Monitor(BATMON) und die Spannung regelmäßig Messen. Vorteile von BATMON ist das hier ein Interrupt ausgelöst wird, wenn die Spannung unter einen bestimmten Wert fällt. Dieser Wert kann zwischen 1.7V - 3.675V liegen. Allerdings ist der Nachteil, dass die genaue Spannung unbekannt ist und so die Lebensdauer der Batterie nicht abgeschätzt werden kann. Wenn man die Spannung Messen will muss man den A/D converter benutzen. Dazu muss er mit den Einstellungen wie in der Abbildung 1 zusehen ist konfiguriert werden. Der ADC wird angeschaltet wenn im Register ADCSRA das ADEN bit auf eins gesetzt wird. Danach kann die Messung gestartet werden in dem man im selben Register den ADSC Wert auf eins setzt. Nun wird ein 16bit Wert im Register ADCW gespeichert. Dieser muss nun mit der Formel umgerechnet werden aus Abbildung 2.

Table 27-10. Recommended ADC Setup for EVDD Voltage Measurements

Parameter	Register	Recommended Setup
ADC Channel	ADMUX, ADCSRB	Select MUX4:0 = 00110 to measure V_{DRTBBP} ; MUX5 = 1;
ADC Clock	ADCSRA	Select a clock frequency of 1 MHz or lower;
V_{REF}	ADMUX	Select the internal 1.6V reference voltage;
Start-up time	ADCSRC	Standard requirement of 20 μ s is sufficient;
Tracking time	ADCSRC	Value depends on ADC Clock, a minimum of 4 μ s is required;

Abbildung 1: Konfiguration zur Messung der Spannung

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$

Abbildung 2: Umrechnung

1.2 Stipe Knez

Vergangene Woche stand das Optimieren dreier Punkte bezüglich des Backends im Vordergrund. Das erste Problem, das behoben werden sollte war, dass wir in unserem Quellcode den COM-Port, über den das ZigBee Modul die Daten sendet, „gehardcoded“ war. Dies führte natürlich zu Problemen (bzw. einem Server crash), wenn man versucht hat, die Anwendung auf anderen Computern zu starten, da der COM-Port, über den das ZigBee Modul seine Daten schickt, ein anderer war. Dieses Problem haben wir durch das Programmieren einer „Port suchen“-Funktion gelöst, die anhand von spezifischen Daten wie „manufacturer“ und „vendor Id“ rausfiltert, an welchem COM-Port das Zigbee Modul angeschlossen ist und dann dementsprechend diesen zum Datenempfang verwendet.

Nachdem dieses Problem gelöst war, ging es an die Lösung eines Weiteren. Der Server stürzte immer dann ab, wenn vor dem Start des Backends kein Zigbee Modul angeschlossen war. Um dieses Problem zu lösen, integrierten wir des Weiteren eine Funktion, die die „Port suchen“-Funktion in einem bestimmten Intervall so lange ausführt, bis ein Modul angeschlossen und somit ein Port gefunden wird.

Die letzte Komplikation, die noch übrig blieb, war, dass der Empfang der Daten des ZigBee Moduls nicht mehr funktioniert hat, sobald man das Modul ausgesteckt und wieder angesteckt hat, während das Backend und somit der Server liefen. Stattdessen sollte es so sein, dass beim erneuten Anstecken das Modul wieder erkannt wird und der Datenempfang normal weiterläuft. Diese Funktionalität wurde dann auch umgesetzt durch das integrieren eines zusätzlichen „event listeners“, der im Fall eines „close“ Events (abstecken des Moduls mit

daraus resultierendem disconnect) die im vorherigen Absatz erwähnte „Port suchen“-Funktion wieder im vorher erwähnten Intervall ausführt, bis ein erneutes anstecken des Moduls stattfindet.

1.3 Lucas Merkert

In dieser Woche habe ich ein wenig weniger gemacht, da ich mit der Fertigstellung eines anderen Projekts beschäftigt war. Es wurde vor allem das weitere Vorgehen besprochen und eine Anpassung des Dashboards Konzepts vorgenommen. Zudem wurde die Dokumentation angefangen und ein besser lesbares Zustandsdiagramm (nicht fertig) erstellt.

1.4 Achim Glaesmann

1.5 Max-Rene Konieczka

Für die vergangene Woche war vorgesehen, die gesendeten Daten in einem Live-Graph anzuzeigen. Allerdings stand zu diesem Zeitpunkt noch nicht ganz fest, wie genau das Dashboard aussehen soll. Daher wurde sich zunächst noch um die Struktur gekümmert. Es werden vier Diagramme zu sehen sein, um die Werte von Temperatur, CO₂, relativer Luftfeuchte sowie der TVOC anzuzeigen. Im Verlauf dieser Woche wird sich nun hauptsächlich um das Anzeigen der Daten gekümmert.

1.6 Can Cihan Nazlier

Diese Woche gab es einige Serverseitige und Clientseitige bugfixes. Es wurde besprochen wie die Ampel schalten soll und nach welchen Werten man geht. Der Server fällt nicht mehr aus wenn kein zigbee set dran ist. Es wurde neue Endpoints fürs dashboard erstellt.

Literatur

- [1] *Datasheet ATmega*. URL: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8393-MCU_Wireless-ATmega256RFR2-ATmega128RFR2-ATmega64RFR2_Datasheet.pdf.