

Projektskizze für eine automatisch einrichtende Wetterstation mit einer grafischen Oberfläche

Fabian Frick, Wiktor Szalanski, Esat Özcelik

24. März 2017

Zusammenfassung

Kommerzielle Lösungen für Wetterstationen sind heutzutage teuer oder bieten nicht die Möglichkeiten, die ein Mikrocontroller mit Messkomponenten des gleichen Wertes bieten könnte. Deswegen soll ein System entwickelt werden, mit dem sich eine modulare Wetterstation einfach aufsetzen lässt. Der einzige Mehraufwand gegenüber den kommerziellen Optionen soll das Zusammenstecken der gewählten Komponenten und das aufrufen eines Skriptes sein. Des weiteren wird dieses Projekt Open-Source lizenziert; somit haben technisch versierte Nutzer die Möglichkeit ihr System anzupassen und zu erweitern.

1 Ausgangssituation

1.1 Kommerzielle Lösungen

Fertige Kommerzielle Lösungen können im Low-End-Bereich vom Funktionsumfang durch einen Thermometer ersetzt werden. Produkte mit dem für dieses Projekt vorgeschlagenem Funktionsumfang sind um ein vielfaches teurer als die Komponenten die gebraucht werden.

1.2 Community und Open-Source

Bestehende Lösungen bieten keine softwareseitige One-Click Funktionalität und erfordern Kenntnisse, die die Fähigkeiten eines durchschnittlichen Wetterstationennutzers übersteigen. Solche Nutzer können somit nicht auf die billigere Community-Projekte zurückgreifen.

2 Nutzen

Private Nutzer können eine High-End Wetterstation für einen Bruchteil des kommerziellen Preises einrichten und nutzen. Der durch die Modularität entstehende Mehraufwand halten wir für vernachlässigbar. Des Weiteren kann jeder die Software erweitern und die Komponenten nach Bedarf aussuchen. Durch die Wirtschaftlichkeit der Lösung ist es auch denkbar solche Wetterstationen als Wetter-Hotspot zu benutzen.

3 Projektmanagement

Durch die Modularität des Projekts und wegen der großen Anzahl an Soll und Kann Anforderungen bietet es sich an ein agiles Vorgehensmodell zu verwenden. Da das Team aus nur drei Entwicklern besteht, wäre ein weitgehend formloses Modell am sinnvollsten um unnötigen Mehraufwand zu vermeiden.

4 Anforderungen

Die Anforderungen an die Wetterstation und deren Komponenten werden in die Kriterien „Muss“, „Soll“ und „Kann“ unterteilt.

4.1 Muss-Kriterien

- Erfassung von Temperatur, Luftdruck und -feuchtigkeit
- Automatisiertes Setup (Einrichtung, Setup, Konfiguration erfolgt automatisch über ein bereitgestelltes Skript)
- Zyklische Erfassung von Messwerten in konfigurierbaren Intervallen
- Speicherung der Messwerte
- Die Wetterstation ist kabellos zugreifbar
- Die erfassten Daten werden in einem Frontend visualisiert und grafisch aufbereitet
- Prototypisches Gehäuse für die Wetterstation

4.2 Soll-Kriterien

- Auswertungs- und Filtermöglichkeiten im Frontend (Zeitfenster, Durchschnittswerte etc.)
- Anzeigeeinheit an der Wetterstation

4.3 Kann-Kriterien

- Erfassung von Niederschlagsmenge, Helligkeitsdauer, Windgeschwindigkeit und -richtung
- „Professionelles“ Gehäuse
- Prototyp: Backend für Raspberry Pi Wetterstationen (Community Projekt => „RPi Wetterkarte“)

5 Projektplanung

5.1 Gantt-Diagramm

				i	Name	Duration	Start	Finish	Predecessors
1					Abgabe Projektskizze		1T 24.03.17	24.03.17	
2					Feedback Projektskizze		7T 25.03.17	31.03.17	1
3					Abgabe Pflichtenheft		1T 07.04.17	07.04.17	
4					Grundsetup RPI		1T 11.04.17	11.04.17	
5					Feedback Pflichtenheft		5T 08.04.17	12.04.17	3
6					Abgabe finales Pflichtenheft		9T 13.04.17	21.04.17	5
7					Recherche Sensor-Komponenten		3T 22.04.17	24.04.17	6
8					Verkabelung der Sensoren + Konfiguration von I2C		1T 25.04.17	25.04.17	4; 7
9					Programmatische Anbindung der Sensoren + Auslesen von Messwerten		2T 26.04.17	27.04.17	8
10					Zyklisches Lesen in konfigurierbaren Intervallen (Cronjob)		5T 28.04.17	02.05.17	9
11					Speicherung der Messwerte		6T 28.04.17	03.05.17	9
12					Einrichtung Setup, Konfiguration automatisieren		23T 12.04.17	04.05.17	4
13					Visualisierung / Frontend		18T 22.04.17	09.05.17	6
14					Finalisierung Betaversion Auarbeitung		5T 11.05.17	15.05.17	
15					Abgabe Betaversion Ausarbeitung		1T 19.05.17	19.05.17	
16					Optimierung und Refactoring		1T 25.05.17	25.05.17	
17					Feedback Ausarbeitung		1T 26.05.17	26.05.17	
					Abgabe finale Ausarbeitung		1T 09.06.17	09.06.17	

