

Gliederung

- es ist wichtig, strukturiert an soetwas heranzugehen, damit funktioniert und keine plötzlichen Strukturprobleme auftreten, deswegen etwas längere Einführung

Vorwort

- doch größeres und komplexeres Projekt als gedacht -> nicht möglich, alles zu erklären
- Ich versuche, ein paar Konzepte genauer zu erklären, der Rest ist in der Seminararbeit

Idee

- Raspi von Ferne steuern, auch von mehreren Geräten gleichzeitig ->
- nicht auf Motor/ Lampen/ Taster festlegen, jeder Benutzer soll eigene Hardware an die Pins anschließen können

Anforderungen

- erweiterbar, nicht nur auf Hardwareebene (also mehr als Lampen/ Taster) auch auf Softwareebene (also App etc)
- mehrere Geräte -> Synchronisation (RasPi)
- keine unnötigen Anfragen an den Server, auch wenn python an sich langsam ist, aber schnell zu schreiben

Umsetzung

Erweiterbar

- Webserver: Im Endeffekt über jeden Browser aufrufbar -> überall nutzbar
- Bibliothek flask: einfach nutzbar, sehr schnell und klein -> Nutzung
- Import der Bibliothek (ähnlich wie beim normalen RPi.GPIO)
- Erstellung einer neuen App (Webserver), **name** ist der Name des Programmes (Erklärung)
- Webserver starten, macht zur Zeit nichts
- Hallo Welt erstellen
 - neue Funktion -> index
 - diese soll Hallo Welt zurückgeben (Flask wandelt String in richtige Antwort um)
 - @app.route('/') sagt wo dann das zu finden ist (/ = Standart, Decorator, eigentlich nicht wichtig)

Mehrere Geräte

Speichern der Konfiguration

- dictionary, hat mehrere Sachen -> Modus, den Modus der Raspberry Pi und die einzelnen Pins in einem weiteren dict gespeichert
 - so sieht es bei dem Start des Webservers aus
- Beispiel konfiguration mit einem Pin (BCM ist ja der normale modus), genauso mit mehreren Pins

Nutzer bei Änderungen informieren

- Socket.IO erklären
 - http: client/ Benutzer fragt immer server an, server hat keine Möglichkeit den Benutzer zu benachrichtigen
 - * sehr ineffizient
 - Alternative: Socket.IO -> Kommunikation in beide Richtungen möglich
 - (intern: auch http, Timeout/ Waiting)
- wichtig, alle Benutzer zu informieren, damit alle auf dem gleichen Stand sind
- *Tafelbild* 1 Server, 3 Clients

Möglichst effizient

- Wiederholung von Vorhergehendem
- es werden nicht alle Daten gesendet, sondern immer nur das was sich ändert, (mobile Daten sparen)
- alles geschafft!

Vorführung

- nicht nur pins/ Lampen sondern auch Ventilator
- selbst steuern

Fazit

- habe sehr viel dabei gelernt
 - ich musste die Dokumentation von GPIO lesen (Benachrichtigung input)
 - eigenen Ersatz für GPIO gebaut, testing
 - design von Webseiten (ist leider hässlich)
- erneut festgestellt: programmieren macht mir sehr viel spaß (Arbeit, Studium)
- mMn funktioniert es sehr gut, könnte für Physik (Schaltkreise) und Informatik (Beginn mit Raspi) genutzt werden