

SmartHome Controller

Beskrivelse:

Projektet er bygget på at jeg gerne selv kunne tænke mig et smart hjem, og der skulle være mulighed for at have en central styreenhed i hjemmet. Selve hjernen i systemet skulle være enheden der havde viden omkring hvor alle de forskellige komponenter der kan interageres med er placeret. Den bør være tilgængelig fra flere enheder og er derfor blevet lavet som et api i en LAMP-stack. Planen er at alle enheder bliver gemt i en database så man kan lave opslag på enhedsnavn og få en GPIO placering. Derefter er det api'et der kan skyde nogle python scripts afsted med eller uden parametre. I api'et er der mulighed for at bruge alle former for request, men har kun implementeret 2 indtil videre.

Ofte vil man nok bruge sin telefon til at styre sit hjem, men da man ikke altid bærer rundt på den kunne det være fedt at have en central styreenhed i hjemmet. Her kommer SmartHome Client delen ind. Der har jeg brugt et STM32 board, der kan sende request til mit api og derfor gøre det samme som en telefon ville kunne. Denne enhed ville skulle monteres på en fast placering og kun stå for det meste nødvendige ting og intet kompliceret.

Jeg har taget udgangspunkt i at kunne tænde og slukke lys, og har derfor endpoints til at tænde og slukke alt lys, men også til hvert enkelt rum. For at tænde de enkelte rum skal man lave et post request med data omkring hvilket rum man ønsker at tænde i bodyen. Jeg har også lavet et endpoint til at trække ud hvor mange rum der er, for at kunne lave noget dynamisk UI på STM32 boardet, men har ikke kunne implementere det korrekt endnu på clienten. Det er med til at give mulighed for tilføjelse af flere rum uden at noget vil stå hardcoded, og gøre skaleringen nemmere.

Teknologier:

Raspberry OS Lite er installeret på raspberry pi'en, for at minimere pladsforbrug og optimere hastigheden. Systemet bygger på linux og man arbejder kun i en CLI samt det er open source. Link til Raspberry OS Lite: <https://www.raspberrypi.org/downloads/raspberry-pi-os/>

Api'et er udviklet i php, bygget på et framework der hedder Lumen, der også bygger på Laravel. Det er et let framework med en god kodebase, da laravel er en af de mest brugte php frameworks. Link til Lumen: <https://lumen.laravel.com/>

Webserver er der brugt **Apache 2**, som er en meget anvendt webserver på alle miljøer. Der er god dokumentation og god hjælp at hente hvis man skulle støde på problemer. Det er et stabilt og opdateret system, og godt til brug af de teknologier jeg har valgt at benytte. Link til Apache: <https://www.apache.org/>

Den installeret database er en **MariaDB**, den er endnu ikke blevet opsat til brug, men bliver forhåbentlig taget i brug inden demonstration af systemet. MariaDB er valgt fordi det er et

pålideligt og meget anvendt. Det giver en sikkerhed i forhold til oppe tid og hvilke problemer der skulle kunne opstå.

Link til MariaDB: <https://mariadb.org/>

Jeg har brugt **Visual Code** som udviklingsværktøj til mit api, da det både understøtter PHP og python. Det er et super lækkert værktøj og bliver brugt af flere og flere udviklere for hver dag. Programmet er gratis og er nemt at bruge. Det er også open source, hvilket åbner op for mange spændende muligheder.

Linkt til Visual Code: <https://code.visualstudio.com/>

Til udvikling af STM32 software har jeg brugt **C++**, det lav niveau kode som giver super hurtig eksekverings tider, men til tider godt kan være svært at arbejde med. Grunden til at det ikke altid er lige nemt er at man er vant til en masse givet funktioner i andre sprog, hvor der her ikke er så mange.

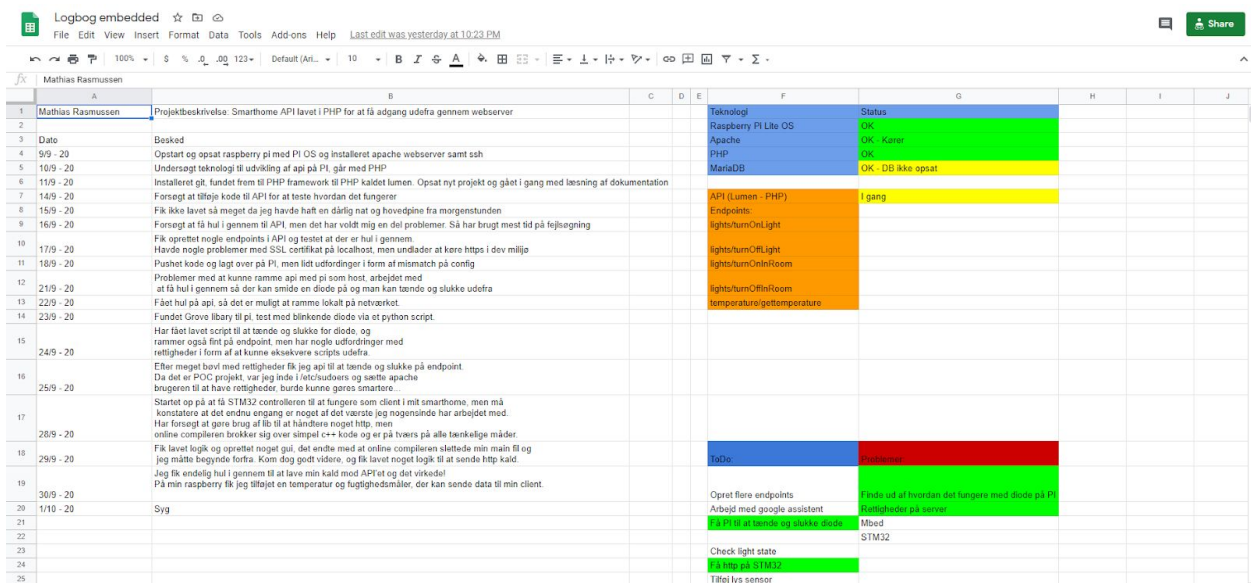
Link til C++: <https://isocpp.org/>

For at compile c++ koden har jeg brugt **Mbed Online Compiler**. Grunden til jeg har holdt det i online compileren er for at spare tid på opsætning af flere udviklingsmiljøer. Det gode ved at køre det online er også at koden kan findes flere steder og at der indbygget versionering historik.

Link til Mbed Online Compiler: <https://ide.mbed.com/compiler/>

Logbog:

Her er et indsat billede af min logbog.



The screenshot shows a web-based logbook application. The title bar reads 'Logbog embedded'. The interface includes a menu (File, Edit, View, Insert, Format, Data, Tools, Add-ons, Help) and a toolbar with various icons. The main content area is a table with columns A through J. The table contains a detailed log of development progress, including dates, tasks, and status updates. The log entries are color-coded: blue for general notes, green for successful completions, and red for errors or warnings. The log entries are as follows:

A	B	C	D	E	F	G	H	I	J
1	Mathias Rasmussen	Projektbeskrivelse: SmartHome API lavet i PHP for at få adgang udføre gennem webserver			Teknologi	Status			
2					Raspberry Pi Lite OS	OK			
3		Besked			Apache	OK - Kører			
4	5/9 - 20	Opstart og opsat raspberry pi med Pi OS og installeret apache webserver samt ssh			PHP	OK			
5	10/9 - 20	Undersøgt teknologi til udvikling af api på Pi, går med PHP			MariaDB	OK - DB ikke opsat			
6	11/9 - 20	Installeret git, fundet frem til PHP framework til PHP kaldet lumen. Opsat nyt projekt og gået i gang med læsning af dokumentation							
7	14/9 - 20	Forsøgt at tilføje kode til API for at teste hvordan det fungerer			API (Lumen - PHP)	I gang			
8	15/9 - 20	Fik ikke lavet så meget da jeg havde haft en dårlig nat og hovedpine fra morgenstunden			Endpoints				
9	16/9 - 20	Forsøgt at få hul i gennem til API, men det har været mig en del problemer. Så har brugt mest tid på fejlsøgning			lights/humCtrlLight				
10	17/9 - 20	Fik oprettet nogle endpoints i API og testet at der er hul i gennem			lights/humORLight				
11	18/9 - 20	Harde nogle problemer med SSL certifikat på localhost, men undlader at køre https i dev miljø			lights/humCtrlRoom				
12	21/9 - 20	Pushet kode og lagt over på Pi, men lidt udfordringer i form af mismatch på config			lights/humCtrlRoom				
13	22/9 - 20	Problemer med at kunne ramme api med pi som host, arbejdet med at få hul i gennem så der kan smide en diode på og man kan tænde og slukke udføre			temperature/gettemperature				
14	23/9 - 20	Fået hul på api, så det er muligt at ramme lokalt på netværket							
15	24/9 - 20	Fundet Grove library til pi, test med blinkende diode via et python script							
16	25/9 - 20	Har fået lavet script til at tænde og slukke for diode, og rammer også fint på endpoint, men har nogle udfordringer med rettigheder i form af at kunne eksekvere scripts udføre							
17	28/9 - 20	Efter meget hovl med rettigheder fik jeg api til at tænde og slukke på endpoint. Da det er POC projekt, var jeg inde i let/sudoers og sætte apache brugeren til at have rettigheder, burde kunne gøres smartere.							
18	29/9 - 20	Startet op på at få STM32 controlleren til at fungere som client i mit smartHome, men må konstatere at det endnu engang er noget af det værste jeg nogensinde har arbejdet med.							
19	30/9 - 20	Har forsøgt at gøre brug af til at håndtere noget http, men online compileren brokker sig over simpel c++ kode og er på tværs på alle tænkelige måder							
20	1/10 - 20	Fik lavet logik og oprettet noget gui, det endte med at online compileren slættede min main fil og jeg måtte begynde forfra. Kom dog godt videre, og fik lavet noget logik til at sende http kald.							
21		Jeg fik endelig hul i gennem til at lave min kald mod APIet og det virkede! På min raspberry fik jeg tilføjet en temperatur og fugtighedsmåler, der kan sende data til min client.							
22									
23									
24									
25									

Logbogen har jeg brugt til at kunne huske hvilke ting jeg skal have med og nogle af de tanker jeg har haft løbende. Så har jeg også brugt den til at opsætte nogle af delmålene vi har fået

givet. Det er ikke altid gået lige godt med at holde den opdateret og gøre brug af den, men jeg kan godt se formålet med at have dette værktøj i sin dagligdag.

Link til logbog:

<https://docs.google.com/spreadsheets/d/1MZjzMXAxiRjS2JC2chLj6PEB6qLOHmo3uwvxJMhxAUA/edit?usp=sharing>

Kommentarer i kode:

Jeg mangler stadigvæk at kommentere koden, men har gjort mig tanker omkring kommentarer i koden.

Normalt mener jeg ikke at kommentarer bør være tilstede, da koden gerne skulle være skrevet selvsigende. Hvis man skulle kommentere i koden er det for at hjælpe den næste der kigger i form af hvis der noget opsætning eller noget der er udover hvad koden gør.

Dokumentation af kode:

Min plan var at installere Swagger på api'et for at kunne se mulige endpoint i en pæn grafisk visning, men jeg har valgt at prioritere det ned i form af jeg er lidt bagud på client delen. Det samme gælder dokumentation af koden på client delen, hvor jeg ville benytte doxygen til at lave en pæn visning af koden. Jeg må bare erkende det ikke blev til noget i første omgang.

Link til swagger: <https://swagger.io/>

Link til doxygen: <https://www.doxygen.nl/index.html>