Embedded Controller III

TEC emhætte controller

af Morten & Virtus

2020

Indholdsfortegnelse

[Logbog 2](#_Toc40702203)

[Onsdag 2](#_Toc40702204)

Torsdag..................................................................................................................................................2

[Fredag 2](#_Toc40702205)

[Mandag 2](#_Toc40702206)

[Tirsdag 2](#_Toc40702207)

[Dokumentation 3](#_Toc40702208)

[Fritzing 3](#_Toc40702209)

[Problematikker 4](#_Toc40702210)

[Brugermanual 5](#_Toc40702211)

# Logbog

OnsdagEfter udlevering af opgaven brugte vi tiden på at brainstorme idéer til projektet og endte med at tage udgangspunkt i en controller til en emhætte. Morten læste om Google Home’s API og konkluderede at det var for meget at bide over. Der skulle laves et sikkert fundament i første omgang og vi brugte noget tid på at generindre undervisning fra sidste Embedded Controller forløb, så vi var forberedte på at starte på opgaven.  
  
TorsdagVi startede med at gennemgå vores tidligere projekt for at huske hvordan det fungerede, samt for at se om der var noget af koden der kunne genbruges. Efter dette lavede vi en simpel Arduino-konfiguration som skulle simulere de forskellige funktioner på emhætten:

* 1 LED til tændt / slukket
* 1 LED til om lyset er tændt / slukket
* 1 LED der skulle blinke hurtigt / langsomt for at simulere udsugnings hastighed

Vi lavede endpoints til at modtage kommandoer fra hjemmesiden. Når man tilgår diverse endpoints fx (http:xxxx/4/on) bliver output4 tændt. Hver knap refererer til forskellige endpoints og outputs.

### Fredag

Vi undersøgte mulighederne for stemmestyring og fandt et Web Speech API, som kunne præcis det vi skulle bruge det til:

API: <https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API>

Interface: <https://developer.mozilla.org/en-US/docs/Web/API/SpeechRecognition>

API’et bruger Jspeech Grammer Format, som vi bruger til at hjælpe med at vægte vores keywords højere end tilfældige ord og prioritere vores commands: <https://www.w3.org/TR/2000/NOTE-jsgf-20000605/>  
  
Med dette implementeret kunne vi via stemmestying tilgå de samme endpoints som knapperne når vores kommandoer blev hørt. Efter dette lavede vi css, html og et embedded billede for at give hjemmesiden et flot finish.

### Mandag

Mandag brugte vi på at skrive rapport og lave dokumentationen.

### Tirsdag

Fremvisning og aflevering af opgave.

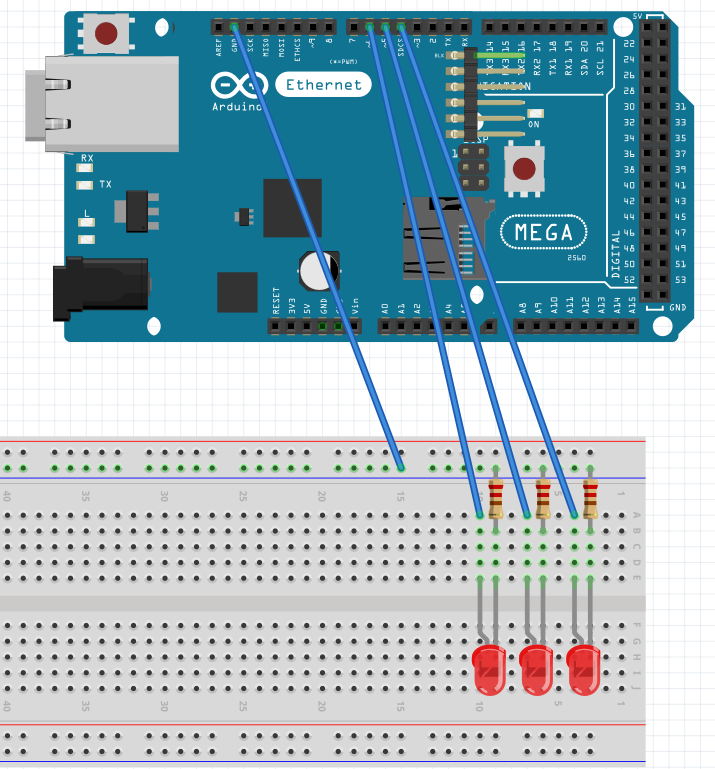
# Dokumentation

Kodedokumentation via Doxygen kan findes i **docs-mappen** og kan tilgås via **index.html** filen.

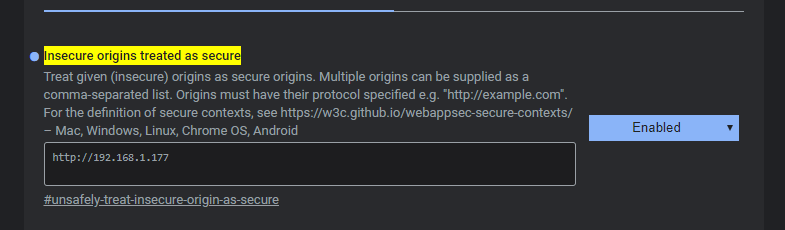
Selve koden er dokumenteret med kommentarer og beskrivelser i **Emhaette.ino**.   
  
Link til github: <https://github.com/VirtusB/extractorhood-arduino>

### Fritzing

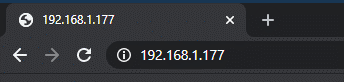
Her kan man se et billede af opstillingen af vores Arduino projekt:



# Problematikker

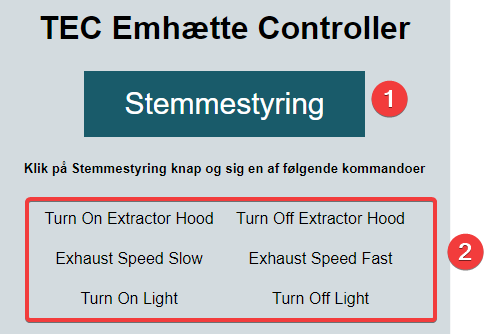
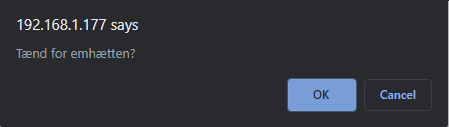
Under projektet stødte vi på et par problematikker som vi skulle arbejde os uden om.  
  
Det første problem var at nogle af vores LED’er ikke virkede, som fik det til at ligne at vores kode ikke virkede, men efter at have vendt og drejet dem et par gange for derefter at udskifte LED’erne virkede det og vi kunne udvikle videre.   
  
Da vi prøvede at implementere vores HTML, CSS og embedded billede stødte vi ind i den første begrænsning på Arduinoen. Da vores metode krævede at man skulle skrive meget af HTML og JS koden direkte til arduinoen som WebServer, løb vi tør for hukommelse. Dette resulterede i at vi måtte finde et andet billede, komprimere det og optimere koden endnu mere for ikke at overstige hukommelsen da hver byte optog dyrebar plads.  
  
En måde at komme uden om dette på ville være at hoste Web Serveren et andet sted, og så bare tilgå den via Arduionen. Alternativ kunne man også sætte et Micro-SD kort i Arduinoen i for at opbevare JS, CSS, HTML og eventuelle billeder, så de kun optog fysisk plads og ikke hukommelse. Dette vil vi have til eftertanke næste gang.  
Vi prøvede også at tilgå Javascriptet fra internettet, men dette kunne Arduinoen ikke hente og bruge, da den ikke havde adgang til internettet.   
  
Vi stødte på et mindre problem med stemmestyring, da det kræver god udtale og et stille miljø for at kunne finde den korrekte kommando. En af måderne vi udbedrede det på var at lave en funktion i JS, **tryToFindCommandBetter**. Vi har f.eks. en kommando som hedder "Turn On Light", og som standard skal Web Speech API have præcis den kommando for at den kan forstås, men ved hjælp af **tryToFindCommandBetter** funktionen som vi lavede, blev det også muligt f.eks. at sige "Turn On Bright Light", da alle 3 ord i "Turn On Light" også findes i "Turn On Brigt Light". tryToFindCommandBetter funktionen løber brugerens tale igennem, f.eks. "Turn On Bright Light", looper hver af vores kommandoer igennem og tjekker om vi har en kommando som kan matche.  
  
Da google anser hjemmesiden som usikkert domæne måtte man ikke bruge mikrofonen til stemmestyring. For at få dette til at fungere skal man bruge chrome og kopiere dette ind i ens browser:  
**chrome://flags/#unsafely-treat-insecure-origin-as-secure**og tilføje WebServerens IP-adresse, klikke **Enabled** og derefter genstarte chrome.Overstående kan også fixes ved at køre via HTTPS eller/og bruge et gyldigt certifikat som chrome stoler på.

# Brugermanual

1. Tilgå hjemmesiden **192.168.1.177** på mobil eller computer   
   
2. Når hjemmesiden er loadet er det muligt at se emhættens nuværende status i teksten over knapperne. Det er muligt at se om emhætten er tændt, hastigheden udsugningen er sat til og om lyset er tændt eller slukket:  
   
3. **Knappernes funktioner:**  
   Knap 1 tænder og slukker for emhætten og udsugningen

Knap 2 ændrer på udsugnings hastighed (Hurtig / langsom)

Knap 3 tænder og sluker for emhættes lys  
  


1. **Stemmestyring:**
   1. Tryk på knappen stemme styring og indtal en af de nedestående kommandoer for at udføre kommandoen. 
   2. Hvis kommandoen bliver hørt korrekt vil du modtage en popup for at bekræfte at det er dette du ønsker at udføre:  
      
   3. Hvis stemmeindtalelsen ikke matcher en af de kommandoer der er defineret i kasse nr. 2, er det muligt at se hvad controlleren hørte i bunden af siden:  
      