

WiFa Lock IoT 2020

5. semester 2020

Science and Technology, Ingeniørhøjskolen, Aarhus Universitet, EDE, Herning

Deltagere

Studienummer	Navn	Initialer
201808188	Dani Mølgaard Olsen	DMO
201808059	Nikolaj Just Morsing	NJM
11754	Jakob Aaboe Vestergaard	JAV

Undervisere

Morten Opprud Jakobsen

Klaus Kolle

Contents

Resumé.....	3
Abstract.....	3
Forord.....	4
Begreber og forkortelser	4
Navne	4
Indledning	5
Problemformulering.....	7
Kravspecifikation.....	7
Afgrænsning.....	8
Analyse.....	9
Lowpower overvejelser	10
Valg af kommunikationstype	11
Hvorfor en webservice.....	12
Web service.....	12
Edge computing.....	14
Valg af hardware	15
Estimering af energiforbrug.....	18
Protokoller	19
Beskedformater	22
Interface analyse	22
Arkitektur	23
Design.....	26
Kommunikation mellem Moduler	26
Kameramodul	27
RC-differentiator.....	29
RGB-board	31
Hovedmodul	32
Implementering	35
Udviklingsplatform.....	35

IDE	35
Sprog & compiler.....	35
Firmware.....	35
Hovedmodul	36
Kameramodul	36
Verifikation.....	37
Diskussion af resultater	39
Konklusion	39
Fremtidigt arbejde	40
Referencer	41

Resumé

Målet med denne rapport er at undersøge hvorvidt det er muligt at forbedre et hverdagsprodukt, ved at bruge principperne bag Internet of Things. Rapporten indeholder en analyse af hvordan disse principper kan benyttes til at udvikle en IoT-prototype.

Rapporten er opdelt i udviklingsfaser; analyse, arkitektur, design, implementering og verifikation. Disse faser giver udviklerne et godt grundlag at arbejde ud fra, fra idé til prototype.

Teamet bag rapporten er selv tilfredse med resultatet og har fået et stort og lærerigt udbytte af denne opgave.

Teamet har ligeledes været tilfredse med samarbejdet og processen.

Abstract

The goal with this paper is to examine if it is possible to improve an everyday product by using the principles behind the Internet of Things. This paper contains an analysis on how these principles can be used to develop an IoT-prototype.

The paper is divided in development phases; analysis, architecture, design, implementation and verification. These phases give the developers a good foundation to build on, from idea to functioning prototype.

The team behind the paper is satisfied with the results and the team has learned a lot during this process.

The team is satisfied with the teamwork and the process.

Forord

Denne rapport er en del af eksamen for kurset E5IOT på Elektronik ingeniøruddannelsen ved AU Herning. Bilag vedhæftes i .zip-format.

Alle dokumenter og source-code kan også findes på GitHub [34], hvor der er at finde en vejledning til læsning af projektet.

Hvem vi er: Team 4 består af:

Jakob Aaboe Vestergaard

Dani Mølgaard Olsen

Nikolaj Just Morsing

Uddannelsessted: Aarhus Universitet, Herning

Studieretning: Ingeniør - Elektronik

Semester: 5. semester

Begreber og forkortelser

Begreb	Definition
IoT	Internet Of Things
AWS	Amazon Web Services
BLE	Bluetooth Low Energy
BT	Bluetooth (classic)
MTMN	Model til detektering af menneskeansigter
MCU	MicroController Unit

Navne

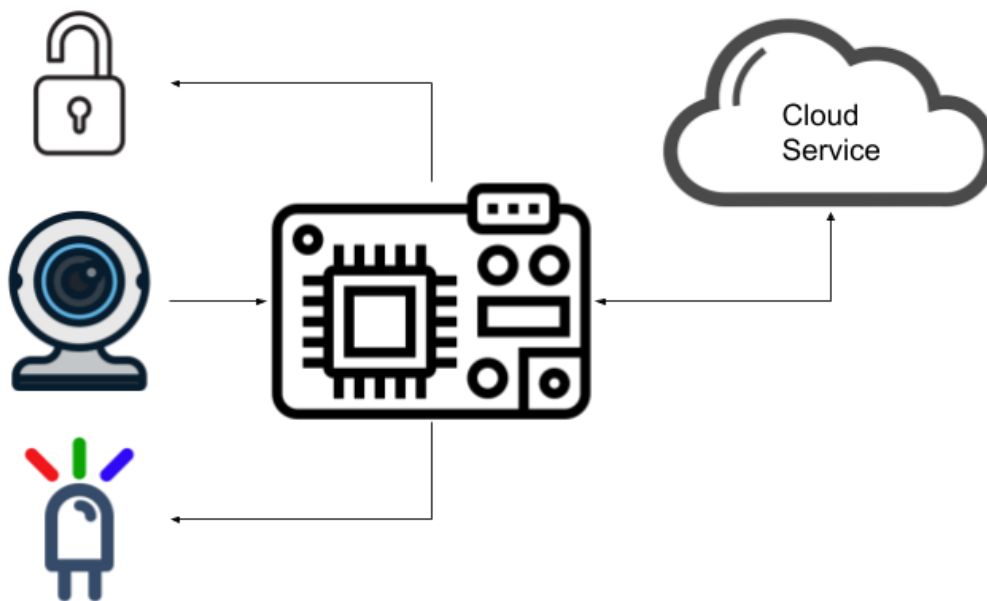
Forkortelse	Betydning
JAV	Jakob Aaboe Vestergaard
DMO	Dani Mølgaard Olsen
NJM	Nikolaj Just Morsing

Indledning

I en verden hvor adskillige IOT-enheder vrirler frem, kan det være svært at lave et produkt, der er brugbart og samtidig ikke allerede er på markedet. Da teamet ikke skal levere et salgbart produkt, vælges der at fokusere på funktionalitet.

Løsningen skal være et smart-produkt, der benytter sig af trådløs kommunikation og online services. Løsningen skal også kunne agere ud fra online-services ved at påvirke aktuatorer.

Med dette projekt arbejder teamet på at designe og udvikle en prototype, der låser f.eks. en dør op for registrerede brugere vha. ansigtsgenkendelse. Dele af produktet er tiltænkt at skulle forsynes vha. et batteri og skal derfor designes til at være så energi-effektivt som muligt.



Figur 1 – produkt idé

Projektbeskrivelse

Formålet med projektet er at anvende teoretiske emner fra kurset i projektet. Projektet løber over hele semestret.

Der skal designes et IOT-artefakt, bestående af embedded hardware, embedded software, sensorer og aktuatorer. IOT-artefaktet skal kunne udføre meningsfuld opgave, bedre end en lignende enhed, som ikke er forbundet til internettet.

Løbende i projektet skal der på GitHub tilføjes og opdateres relevante filer. Denne GitHub skal være offentlig tilgængelig, hvor det skal være muligt at følge forløbet i udviklingen af IOT-artifaktetartefaktet.

Ansvarsområder

Emne	Beskrivelse	Deltager
Analyse	Forudgående analyse samt general analyse	Alle
Design	Design af systemet som helhed	Alle
Arkitektur	Beskrivelse og udformning af arkitekturen	Alle
Kameramodul	Implementering af kameramodulet	DMO
Hovedmodul	Implementering af hovedmodulet	JAV + NJM
Cloud service	Analyse og design af Cloud Service	JAV+ NJM
Edge Device	Neuralt netværk til ansigtsdetektering	DMO
Test	Funktionalitets-test	Alle

Problemformulering

Hvordan kan man forbedre et hverdagsprodukt ved at bruge principperne bag IoT-devices?

Kravspekifikation

Ud fra den ønskede funktionalitet ønskes der, mulighed for få adgang til et låst objekt, vha. ansigtsgenkendelse. Dette skal foregå automatisk, når en person nærmer sig kameraet. Det ønskes at dele funktionaliteten op i 2 blokke, et kameramodul og et hovedmodul. Dette gøres da hovedmodulet, som styrer låsen, skal have mulighed for sidde et eksternt sted, der kan være svært og komme til. Ud fra dette og de krav stillet for projektet i kurset [03], er der udformet krav vha. EARS-metoden [04].

Ubiquitous

- Kameramodulet skal være low power og kunne holde strøm i op til 30 dage, ved normal brug
- Kameramodulet skal optage i høj nok opløsning til at foretage ansigtsgenkendelse og objektgenkendelse.
- Moduler skal kommunikere sammen vha. WiFi/bluetooth
- Hovedmodulet skal være forbundet til internettet vha. WiFi
- Kameramodulet skal aktiveres vha. en sensor
- Hovedmodulet skal aktiveres vha. netværkskommunikation
- Kameramodulet skal drives med 5v vha. batteri.
- Kameramodulet skal sende billeder til hovedmodulet
- Systemet skal automatisk oprette forbindelse mellem kameramodulet og hovedmodulet
- Kameramodulet skal implementeres som et edge device, hvor der udføres ansigtsdetektering på billeder

Event Driven

- Når et menneske stiller sig foran, indenfor 2 meter, af kameramodulet. Skal systemet tage et billede og foretage objektgenkendelse (Person).
- Når hovedmodulet modtager et billede fra kameramodulet, skal billedet sendes til (Online facial recognition service) og verificeres op mod reference billeder, hvis billedet matcher et referencebillede, skal låsen låses op.
- Når objektgenkendelse identificerer en person, sendes billede til hovedmodul.
- Når et billede afvises, skal LED lyse rødt.

State Driven

- Mens kameramodulet er i power saving tilstand, skal systemet vågne op ved input fra digital indgang.
- Mens Hovedmodulet er i power saving tilstand, skal systemet vågne op ved kald fra WiFi/bluetooth.
- Mens lås er låst op, skal LED på kameramodulet lyse grønt.
- Mens kameramodulet afventer verifikation, skal LED lyse blå.
- Mens kameramodulet har afsendt et billede til hovedmodulet, skal kameramodulet vente i max 10 sekunder på verifikation.
- Mens låsen er låst op, skal systemet ikke låse før objektet låsen er på, er lukket.

Unwanted behaviour

- Hvis kameramodulet vækkes af andet end et menneske skal billedet ikke sendes til hovedmodulet
- Hvis kameramodulet ikke kan forbinde til WiFi/bluetooth skal modulet gå i power saving tilstand.
- Hvis hovedmodulet ikke kan forbinde til WiFi/bluetooth skal modulet gå i error-tilstand, blink med LED.

Afgrænsning

For at kunne færdiggøre en prototype, er der valgt ikke at implementere følgende krav:

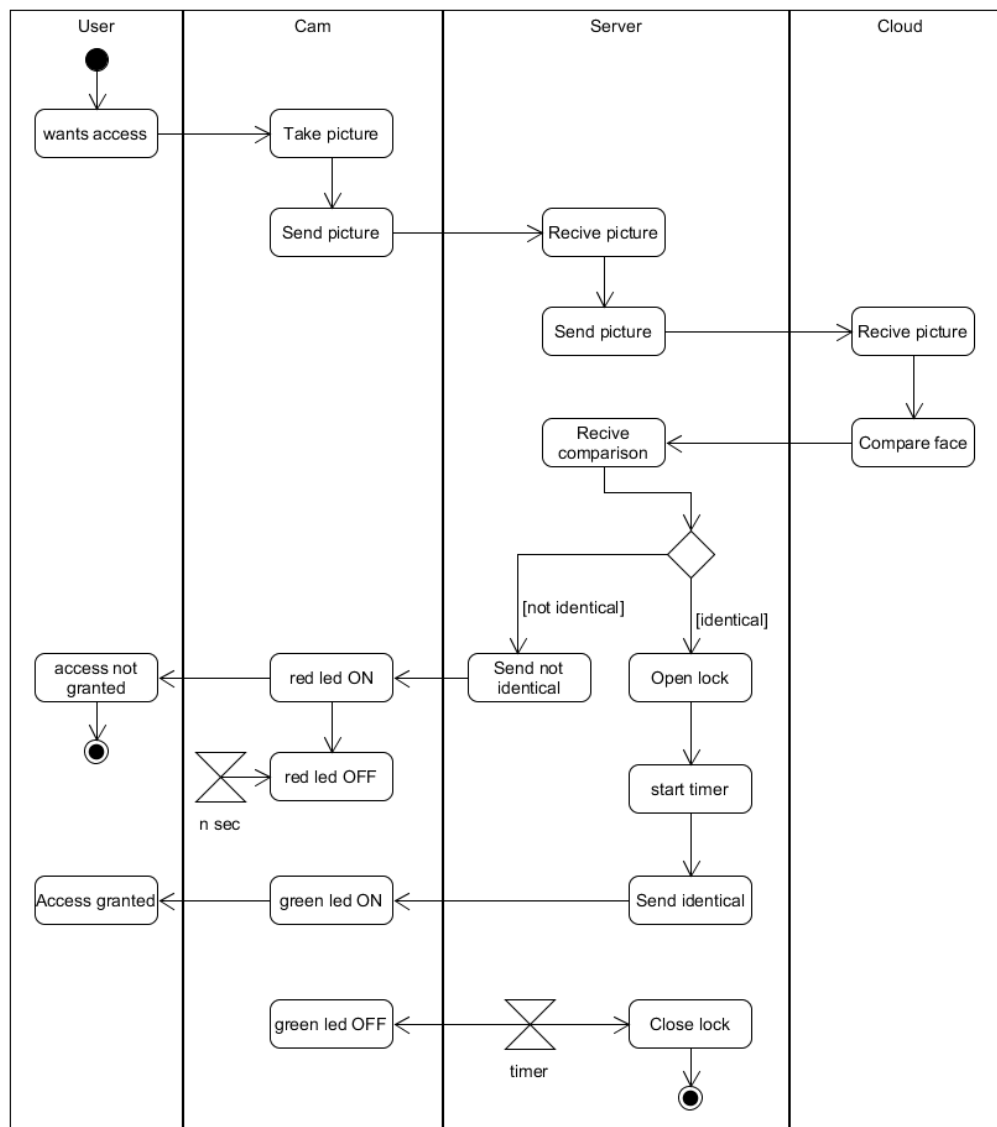
- Kameramodulet skal drives med 5v vha. batteri.
- Mens lås er låst op, skal systemet ikke låse før objekt er lukket.

At kameramodulet skal drives af et 5v batteri, er ikke vigtigt for prototypen, det er dog vigtigt at notere sig, der er andre krav, som sikrer at system-to-be gøres strømbesparende og derfor vil det være muligt at drive system-to-be med et batteri over en længere periode. Kravet er fravalgt pga. prisen for et lille batteri med mange mAh er høj og tilføjer intet til prototypen, som ikke kan simuleres.

Låsen som benyttes i system-to-be er simuleret med en simpel servomotor, derfor vil der ikke blive brugt ressourcer på at implementere yderligere logik vedrørende låsen.

Analyse

For at danne overblik over system-to-be, er der lavet et aktivitetsdiagram ud fra kravene til systemet. Diagrammet viser flow af systemet, de forskellige moduler og giver en indledende ide, om arkitekturen af systemet og dets funktionalitet.



Figur 2 - Aktivitets diagram

Low power overvejelser

For low power design er der to typer forbrug, dynamisk og statisk strømforbrug. Til at sikre at system-to-be kan gøres så strømbesparende som muligt, dannes der overblik over de forskellige teknikker og muligheder der er.

Static power consumption

Static power consumption er det energi der bruges, når systemet ikke aktivt foretager sig noget. Dette vil typisk være en bekymring for enheder, hvor produktet størstedelen af tiden er i sleep-tilstand.

For dette projekt vil eksempler på statisk strømforbrug være:

- Når kameramodulet er i sleep-tilstand vil PIR-sensoren stadig være forsynet
- Tab i PIR-sensor
- Forbrug af kameramodul når dette er i sleep-tilstand
- Self-discharge på batteriet

Dynamic Power consumption

Dynamic power consumption er det energi der bruges, når systemet kører. Det kan være alt fra tab i komponenter til funktionalitet af, at produktet afvikles.

For dette projekt vil eksempler på dynamisk strømforbrug være:

- Når bevægelse detekteres af PIR-sensor
- Når et billede tages
- Når det undersøges om der er en person på billedet
- Når et billede sendes til hovedmodulet

Ting udviklere kan kontrollere

- GPIO
 - Input-pins forbruger mindre energi når input spændingen er tæt på forsyningsspænding (+/-)
 - Interne pullups i stedet for eksterne, så strømvejen til stel kan deaktiveres f.eks. efter detektering af tryk på en knap
- Sleep
 - Perifære enheder kan sættes i sleep og ubrugte kan deaktiveres
 - Selve microprocessoren kan sættes i sleep
- Printlayout
 - Minimere impedans i strømveje
 - Minimere impedans i highspeed switching strømveje

- Minimere lækstrøm
- Minimere dutycycles
- Komponenter
 - Vælg komponenter med så lavt tab som muligt
 - Lav lækstrøm på kondensator
 - Batterier med lav self-discharge
 - Outputs har lav impedans, inputs har høj impedans
 - Gerne så høje impedanser som muligt – husk dog at dette kan give uønskede virkninger, som f.eks. I2C hastighed nedsættes eller overfølsomhed overfor støj
- Kommunikations protokol

Valg af kommunikationstype

Et vigtigt element hvor der kan spares strøm, er på kommunikation. Forskellige kommunikationstyper har forskellige karakteristisk, disse kan ses i Tabel 1.

	Datarate (teoretisk)	Strømforbrug	Rækkevidde	Latency
BT	1-3Mbit/s	~1W	~100m	~100mS
BLE	<2Mbit/s	<0.5W	<100m	6mS
WiFi	<2.4Gbps	2-20W (avg 6W)	<50m	2-6mS

Tabel 1 - Vigtige parametre for trådløse teknologier

Tabellen viser tydeligt at under optimale forhold ville BLE være det gode valg for kommunikationen imellem moduler for dette projekt. Det har en acceptabel datarate, et meget lavt energiforbrug og acceptabel latency. Det er vigtigt også at blive opmærksom på at WiFi er et ufravigeligt krav, da der skal kunne kommunikeres med internettet. BLE er kun en løsning til at kommunikere mellem moduler i IOT-løsningen.

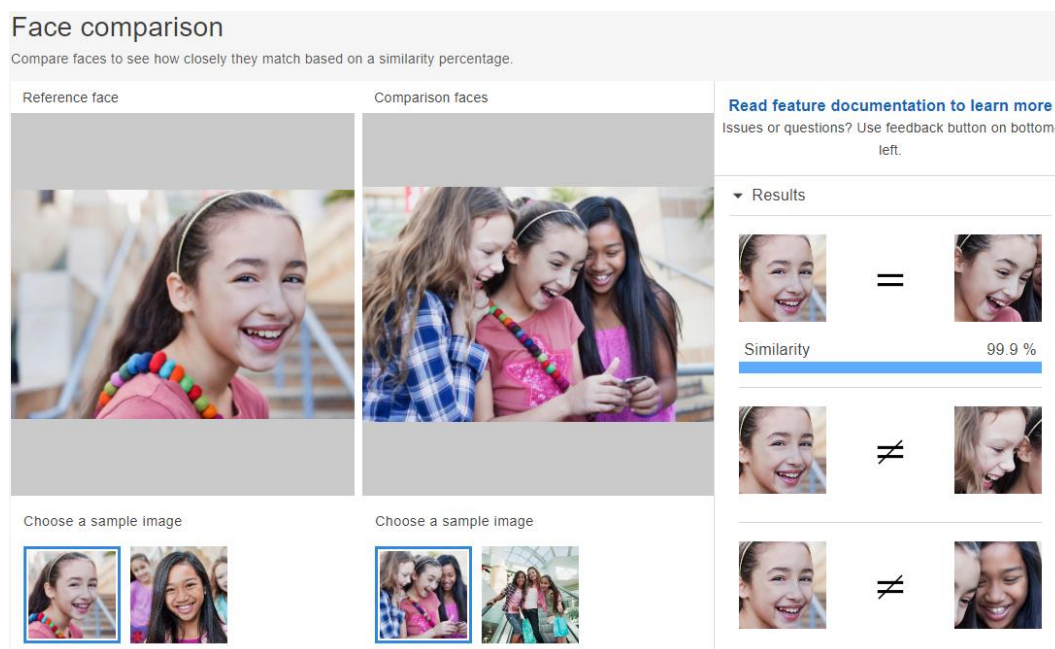
Hvorfor en webservice

Ansigtsgenkendelse er typisk implementeret med et deep learning neural network. Disse neural networks kræver typisk kraftig hardware for at være så hurtige som de er. Disse neurale netværk kræver derfor også en masse energi for at arbejde. Dette driver prisen i vejret, hvis ikke der benyttes en online service.

Alternativt kunne der benyttes et smart edge device, hvor selve modellen er trænet af en online service, men hvor sammenligningen imellem input og modellen foregår lokalt. Der er efterhånden muligt at få devices med hardware specifikt designet til neurale netværk direkte på boardet, men det er stadig ret ny teknologi. At kunne foretage edge computing er en stor fordel, især fordi det vha. dette giver mulighed for at reducere mængden af data der skal transmitteres til backend.

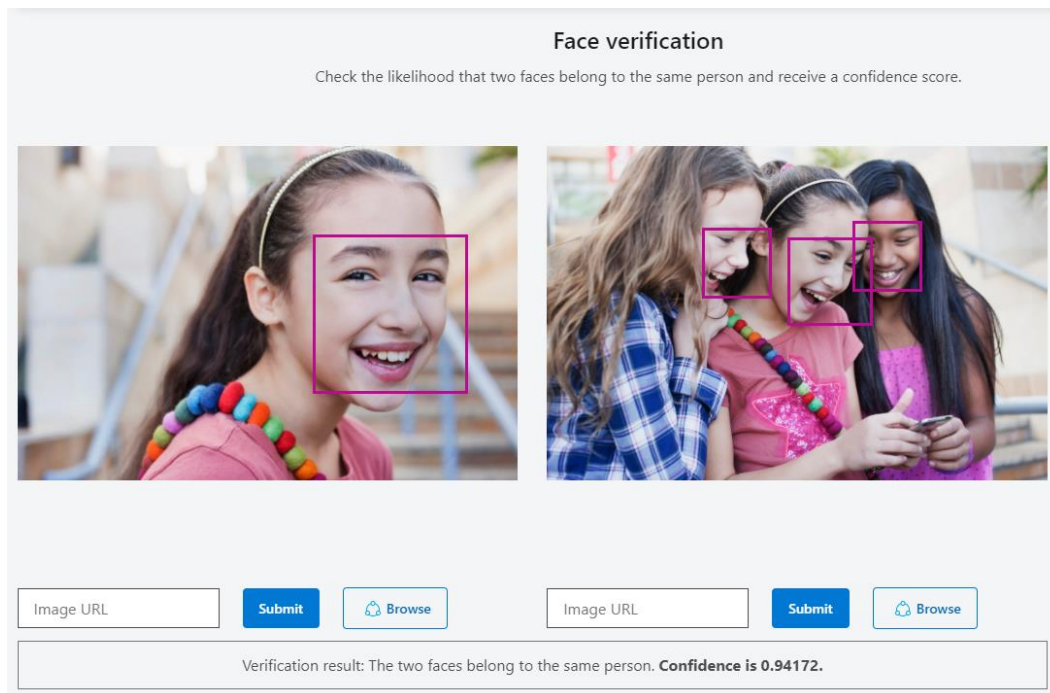
Web service

Til dette projekt er det besluttet at en webservice skal benyttes til ansigtsgenkendelse. Teamet fandt 2 mulige kandidater; Microsoft Azure Facial Recognition og Amazon Web Services Rekognition.



Figur 3 - AWS Rekognition eksempel (<https://eu-west-1.console.aws.amazon.com/rekognition/home?region=eu-west-1#/face-comparison>)

AWS har servicen Rekognition gratis tilgængelig til nem afprøvning. Ansigtsgenkendelse er meget god og meget hurtigt udført. På hjemmesiden er det nemt at sammenligne to ansigter, det er dog langt fra nemt at få AWS Rekognition til at arbejde sammen med vores system.



Figur 4 - Face Verification Test af Microsoft Azure (<https://azure.microsoft.com/en-us/services/cognitive-services/face/#demo>)

Microsoft hoster også en demo af deres ansigtsgenkendelse. Samplebillederne fra AWS er genbrugt til denne test, for at kunne sammenligne de to services. Tilsyneladende ser AWS ud til at være mere præcis, da AWS scorer 99,9% confidence og Azure scorer 94.1% confidence. Dog er Microsoft Azure's service nemmere at implementere sammen med Particle platformen.

For at teamet kan fokusere på at implementere så meget funktionalitet som muligt på den korte tid der er til rådighed, vælges Azure som cloudservice.

Edge computing

Ved at inkorporere artificial intelligence eller machine learning, er det muligt at processere meget data på små MCU'er. Dette fænomen kaldes for edge computing. I projektet vil der i forsøget på at gøre kameramodulet mere strømbesparende blive brugt et neuralt netværk til at detektere ansigter. Uden et neuralt netværk til at udføre denne opgave ville processen tage for lang tid og gøre brugeroplevelsen dårlig. Det er derfor også vigtigt at der kan findes en model, som er effektiv både i form af ansigtsdetektering og hastighed.

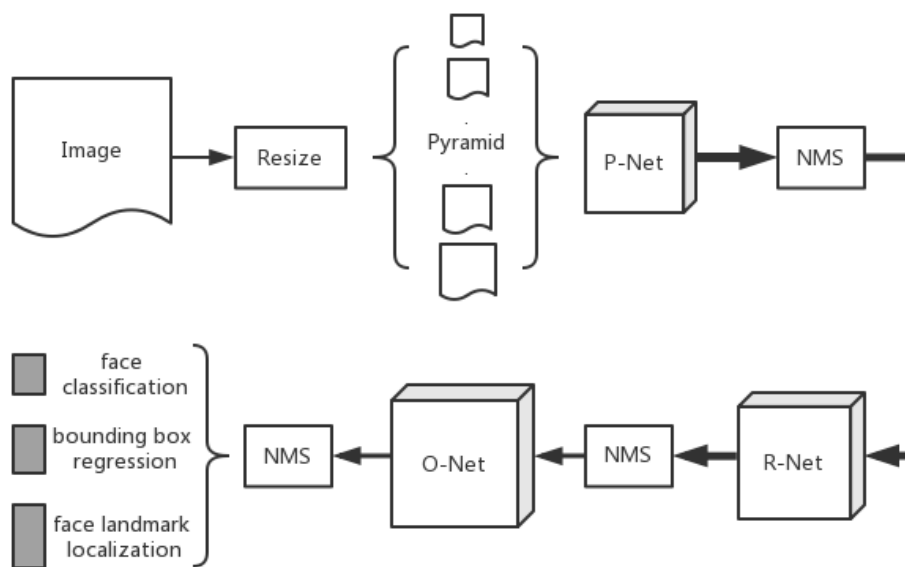
Ved at udføre en tidlig processing af billederne, kan der spares strøm ved ikke at sende billeder, hvor der ikke kan identificeres ansigter. I [05] kan det tydeligt ses at WiFi bruger betydelige mængder af strøm på at overføre data. Udover at der kan spares strøm ved ikke at sende billeder, der ikke kan detekteres ansigter på, spares der også på dataforbindelsen, samt forespørgsler til cloud API'er.

I denne IOT-løsning benyttes der en cloud løsning til at foretage ansigtsgenkendelse, løsningen koster penge per kald, hvilket i mange scenarier er en virkelighed for hostede cloud løsninger. Derfor kan det også være en ide at nedbringe antallet af kald til cloud API'er.

Modellen som vil blive brugt, er en MTMN [12] model og er baseret på multi-task cascaded convolutional netværk. Modellen er ikke en teamet selv har trænet, men en model der er specifikt udviklet til at detektere ansigter af espressif. Modellen er baseret på tre dele:

- Proposal Network (P-Net) – foreslå kandidatområderne og sender disse til R-Net.
- Refine Network (R-net) – Screener kandidatområderne fra P-Net.
- Output Network (O-Net) – håndterer slutresultater. i.e. confidence koefficient.

Opbygning af modellen kan ses i Figur 5. modellen har mulighed for konfigurerings hvilket giver større anvendelsesmuligheder, samt gør det muligt at opsætte modellen specifikt til system-to-be.



Figur 5 - MTMN model

Valg af hardware

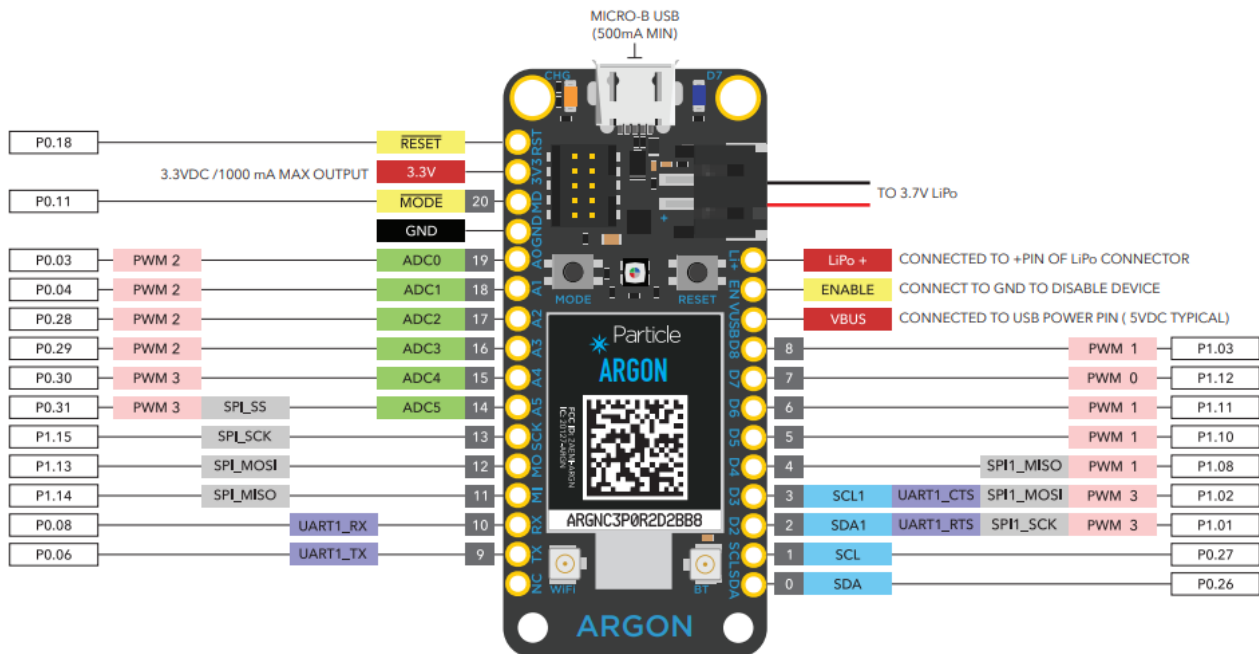
Til at realisere system-to-be kræves der hardware, som kan opfylde [Kravspecifikation]. Det valgte hardware har udover kravene til system-to-be, også skulle være nemt at arbejde -og være hurtigt at prototype med. Uden disse kvaliteter vil en færdig prototype ikke være mulig indenfor tidsgrænsen.

Particle Argon

Til at kommunikere med internettet og styre låsen, er der valgt at benytte en Particle Argon [06]. En af hovedårsagerne til at Argon er valgt til at være gateway til internettet, er med en Argon medfølger et cloud-miljø [07]. Cloud-miljøet har en bred palette af funktioner til at hjælpe ved udvikling af IOT-enheder. De betydende faktorer for valget er:

- Hurtig at opsætte og dermed god til prototype udvikling
- Platformen er blevet benyttet i undervisningen i kurset E5IOT
- Nemt at oprette webhooks
- Low power
- Mange I/O
- BLE + WiFi

I Figur 6 ses et overblik over en Argon. Processeringskraften på en Argon består af en ESP32-D0WD, der benyttes som co-processor og en nRF52840 SoC som hovedprocessor.



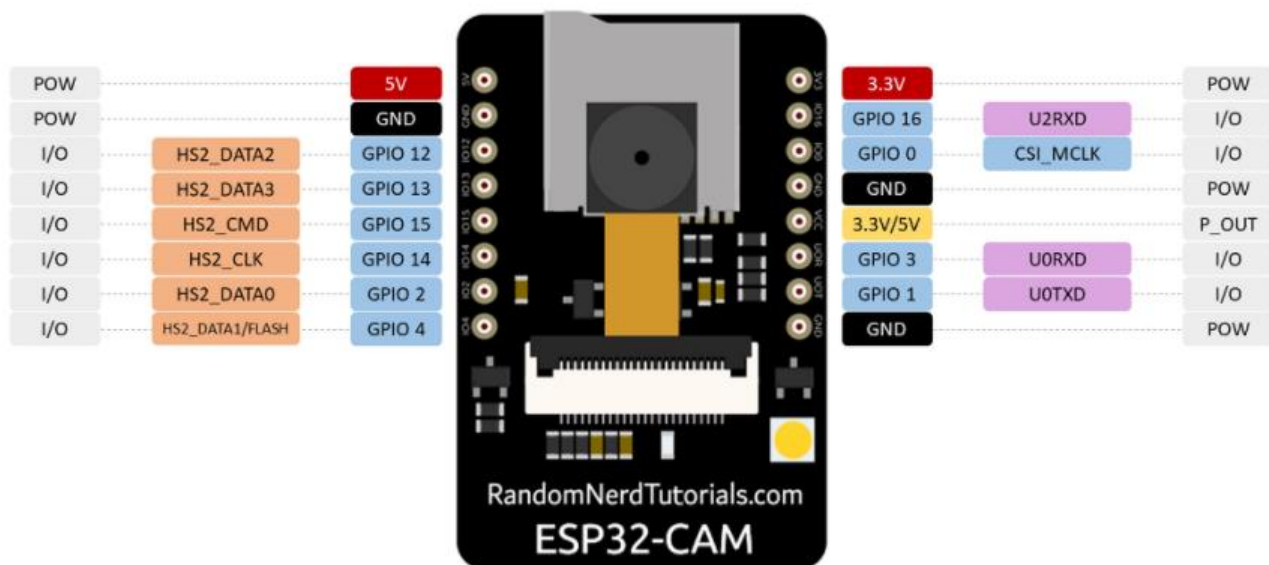
Figur 6 – Argon pinout (<https://docs.particle.io/assets/images/argon/argon-pinout-v1.0.pdf>)

ESP32-CAM

Til at foretage tidlig billedprocessering og tage billeder af personer, der ønsker adgang. Er der forsøgt at finde et board, som understøtter et kamera og samtidigt har en relativ kraftig processor, da billede analyse er tungt. Desuden har det også været vigtigt at enheden har mulighed for at gå i sleep-tilstand og være low power da ideen er at enheden skal drives af et batteri. Valget er derfor endt på en ESP32-CAM [08]. Faktorer for valget er:

- Næsten samme arkitektur som Argon – hurtig at opsætte og derfor god til prototype udvikling
- Low power
- Kraftig processor
- En komplet pakke med det, teamet havde brug for
 - Indbygget kamera, flash, WiFi/BLE og antenne
- I/O til RGB og Sensor input
- pris

Figur 7 viser oversigt over ESP32-CAM.

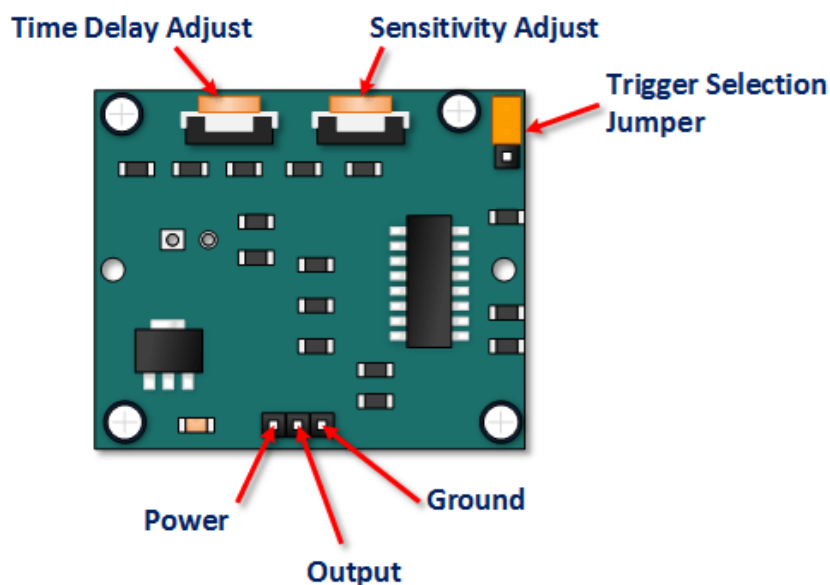


Figur 7 - ESP-CAM pinout (<https://randomnerdtutorials.com/esp32-cam-ai-thinker-pinout/>)

HC-SR501

Til at aktivere kameramodulet er der blevet valgt en færdig løsning bestående af en PIR-sensor, hvor der er mulighed for at justere sensitivitet og delay. En PIR-sensor er et oplagt valg da den også virker når det er mørkt. Sensoren er en HC-SR501 [09].

- Enkelt modul, med mulighed for justering af sensitivitet og time delay
- Nemt at tilslutte og benytte
- Lav standby strøm

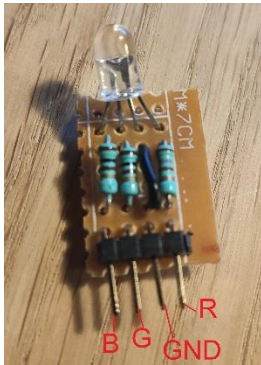


Figur 8 - HC-SR501 pinout (<https://ardustore.dk/produkt/motion-pir-module-adjust-hc-sr501>)

RGB-LED

Til at signalere brugeren omkring hvilket stadie system-to-be er i, er en RGB-diode et oplagt valg da den ikke fylder meget. Der er derfor blevet lavet et lille board med en RGB-diode og modstande, se Figur 9.

- Mulighed for mange farver, enkelt diode



Figur 9 - RGB-diode

RC-differentiator

Da ESP32-CAM får signal til at opvågne fra HC-SR501, har det vist sig at være nødvendigt at reducere tiden GPIO-benet får input. Hvis input holdes høj, eller lav, for længe sker der det, at ESP32-CAM booter op mere end én gang, hvilket resulterer i ekstra strømforbrug. For at imødekomme nødvendigheden af, at input signalet er kort, vil der blive brugt et RC-differentiator kredsløb.

- ESP32-CAM kræver kortvarig gpio signal ved op vækning fra sleep-tilstand, RC-differentiatoren sikre der ikke udføres "double" boot, når HC-SR501 sender signal

Alle specifikationer kan findes i bilag [31].

Estimering af energiforbrug

Ifølge kravspecifikationen er det et ønske, at enheden kan holde strøm i 30 dage. Derfor er der lavet estimering af enhedens energiforbrug for at finde en omtrentlig kapacitet for batteriet.

Udregningerne for et estimeret energiforbrug er lavet ud fra producenternes oplyste energiforbrug af de benyttede komponenter.

Energiforbruget udregnes ud fra at der er gennemsnitligt 6 aktiveringer i døgnet, resten af tiden er enhederne i dvale.

Dette vil give teamet en idé om, hvor meget der kræves af batteriet for at opnå 30 dages levetid.

Tidsforbruget i diverse tilstande er fundet vha. eksperimenter med hardwaren hvilket gav et groft gennemsnit for hvor lang tid en proces har kørt.

Udregningerne kan ses i bilag [05].

Task	Consumption	Duration	Duty cycle	Total power
Sleep	6.005 mA	~86306s	99%	~518267.53 mC
Capture image	310 mA	1.2s	< 1%	372 mC
Facial detection + boot up	180 mA	18s	< 1%	3240 mC
PIR-On	65 mA	15s	<1%	975 mC
Send image	260 mA	0.024s	<1%	6.24 mC
Receive response	85 mA	60s	<1%	5100 mC
Total (1 day)				527960,77mC
Total (30 days)				1583807.2 mC
Battery Capacity			1583807.2/3600	4399.67 mAh

Tabel 2 - Udregningsoversigt af batteriforbrugs-estimering

Varigheden af hver task i Tabel 2 er opgjort for et helt døgn.

En kapacitet på 4400 mAh kan opnås med et 21700 Li-ION batteri på 5000 mAh. Dette kræver dog en stepup DC-DC-converter fra 3.7V til 5V, hvilket også vil give et let øget energiforbrug. Dog kan man komme op på 85-95% effektivitet på disse convertere.

Protokoller

Ved kommunikation mellem moduler og eksterne services, benyttes der forskellige protokoller og beskedformater. System-to-be benytter sig af forskellige protokoller, der hver bruges til forskellige formål. Protokollerne sikrer at data kan sendes og genskabes igen på modtager siden.

TLS

For at sende et http kald til Azure skal kaldet krypteres, dette gøres med Transport Layer Security (TLS) protokollen. TLS-protokollen sørger for forbindelsen imellem to enheder er krypteret, så udefrakommende ikke kan lytte med.

TCP / IP

Til at sende data mellem hoved- og kamera-modul benyttes TCP/IP protokollen, der benyttes TCP frem fra UDP til at sikre modtagelsen af pakker.

TCP / IP header			
Version: 4 bytes	IHL: 4 bytes	TOS: 8 bytes	Total length: 16 bytes
Identification: 16 bytes	Flags: 3 bytes	Fragment offset: 13 bytes	
TTL: 8 bytes	Protocol: 8 bytes	Header checksum: 16 bytes	
Source IP address: 32 bytes			
Destination IP address: 32 bytes			
Options and padding: 32 bytes			
Data (MTU 1500 bytes)			

http protokol

For at verificere et ansigt skal hovedmodulet lave et http kald til Azure cloud. http kaldet skal følge standarden for http kald, derudover skal det overholde Azures specifikationer. Et http kald er bygget op af en request URL, header og en body. For at verificere et ansigt skal der laves 2 http kald til Azure, et detect- og verify kald.

HTTP header							
METHOD	sp	URL	URI	sp	Version	cr	lf
Header field name	:	value				cr	lf
.....							
Header field name	:	Value			Cr	lf	cr lf
http body							

Figur 10 - http header format

Specifik opbygning af http request og respons til Azure kan finde i bilag [10].

BLE protokol

Til at sende data mellem hoved -og kameramodul er der undersøgt BT/BLE protokollen, da strømforbruget er en del mindre end wifi [Tabel 1].

BLE er et oplagt valg med det lave strømforbrug, lave latency og tilfredsstillende hastighed. Det har dog vist at Argon har en relativ lav hastighed over BLE som kan ses på Figur 11. Udover den lave hastighed, er den maksimale størrelse af en pakke 244 bytes. Hvis der skal sendes mere end 244 bytes vil der skulle afsendes adskillige pakker.

Characteristic Size	Maximum Data Transfer Rate
20 bytes	222 bytes/sec.
100 bytes	1100 bytes/sec.
200 bytes	2210 bytes/sec.
236 bytes	2186 bytes/sec.
237 bytes	1753 bytes/sec.
244 bytes	1793 bytes/sec.

Figur 11 - Udklip fra Particle Argons specifikation (<https://docs.particle.io/tutorials/device-os/bluetooth-le/>)

For at udregne tiden det vil tage og transmittere, bruges et billede 320x240 med en bit dybde på otte, som eksempel. Da bit dybden er otte, vil en pixel være lig én byte.

$$320 \text{ bredde} * 240 \text{ højde} = 76800 \text{ bytes}$$

Ligning 1 - størrelse af billede

$$\frac{76800 \text{ bytes}}{1793 \frac{\text{bytes}}{\text{s}}} = 43 \text{ s.}$$

Ligning 2 - tid for transfer af billede

43 sekunder for at transmittere et billede er ikke brugbart, dette vil med stor sandsynlighed også bruge mere strøm, end at sende det over wifi eller bluetooth classic.

Bluetooth classic er dog ikke en mulighed da Particles API ikke har implementeret BT protokollen [11]. Derfor falder valget på WiFi som kommunikationsmediet mellem moduler for dette projekt.

Beskedformater

Data der sendes mellem moduler og eksterne services, skal formateres således datastrukturen er ens hver gang. Ved data indkapsling bruges der forskellige beskedformater alt efter hvilken kommunikationsprotokol der benyttes og data der sendes, samt hvor dataene skal sendes.

Den specifikke opbygning af http-request og http-respons, samt beskedformatet imellem hovedmodul og kameramodul, kan findes i bilag [10].

Interface analyse

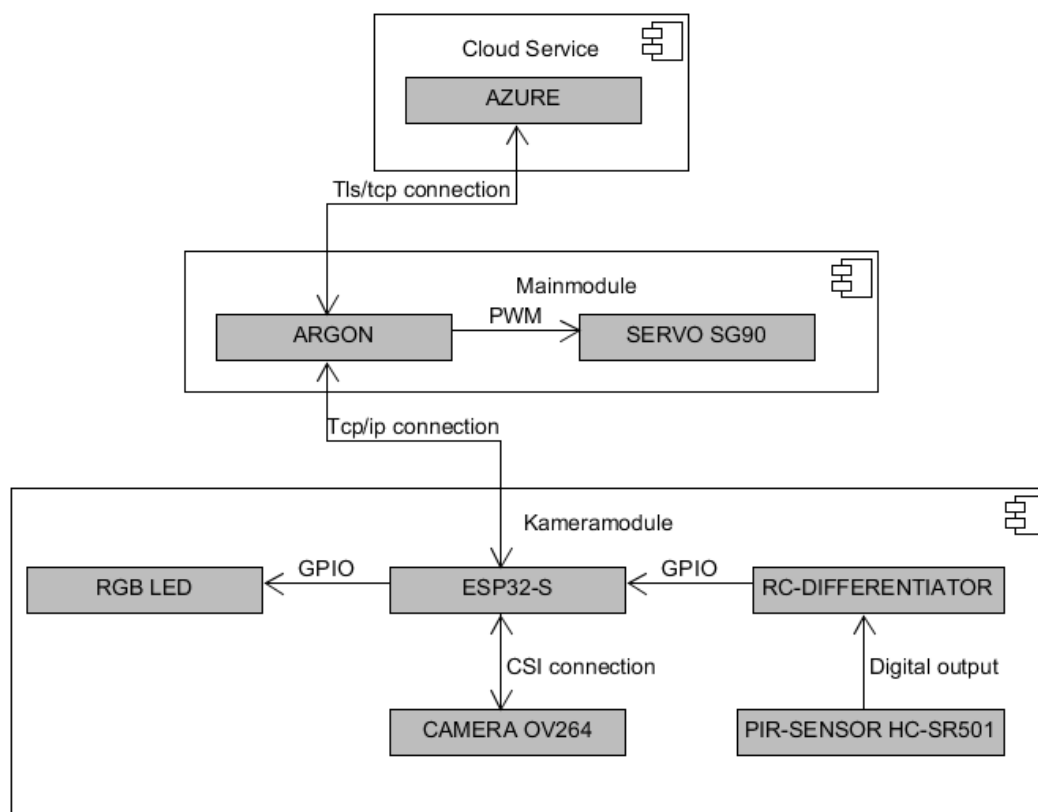
For at gøre implementeringen af alle moduler lettere, identificeres alle interfaces imellem moduler. Dette sikrer at moduler kan interagere med hinanden effektivt, da der er et fælles grundlag for kommunikationen imellem moduler.

Enheder	Kommunikationsparametre
HC-SR501 → RC-differentiator	Signal type: Digital HIGH/LOW Voltage: 3.3V One-shot modul
RC-differentiator → ESP32-CAM	Signal type: Digital HIGH/LOW Voltage: 3.3V Pin on ESP32-CAM: GPIO13
ESP32-CAM → HC-SR501	Signal type: Supply voltage Voltage: 5V Pin on ESP32-CAM: P_OUT 5v
ESP32-CAM → Status LED	Signal type: Digital HIGH/LOW Voltage: 3.3V Pin on ESP32-CAM: GPIO12, GPIO14, GPIO15
ESP32-CAM → Argon	Signal type: WiFi 802.11 b/g/n Protocol: TCP/IP Data packet: binary stream
Argon → ESP32-CAM	Signal type: WiFi 802.11 b/g/n Protocol: TCP/IP Data packet: Change state on status LED
Argon → Azure	Signal type: Internet Protocol: TLS Data Packet: binary stream Requires: API-key
Azure → Argon	Signal type: Internet Data Packet: HTTP response

Tabel 3 - Interface analyse

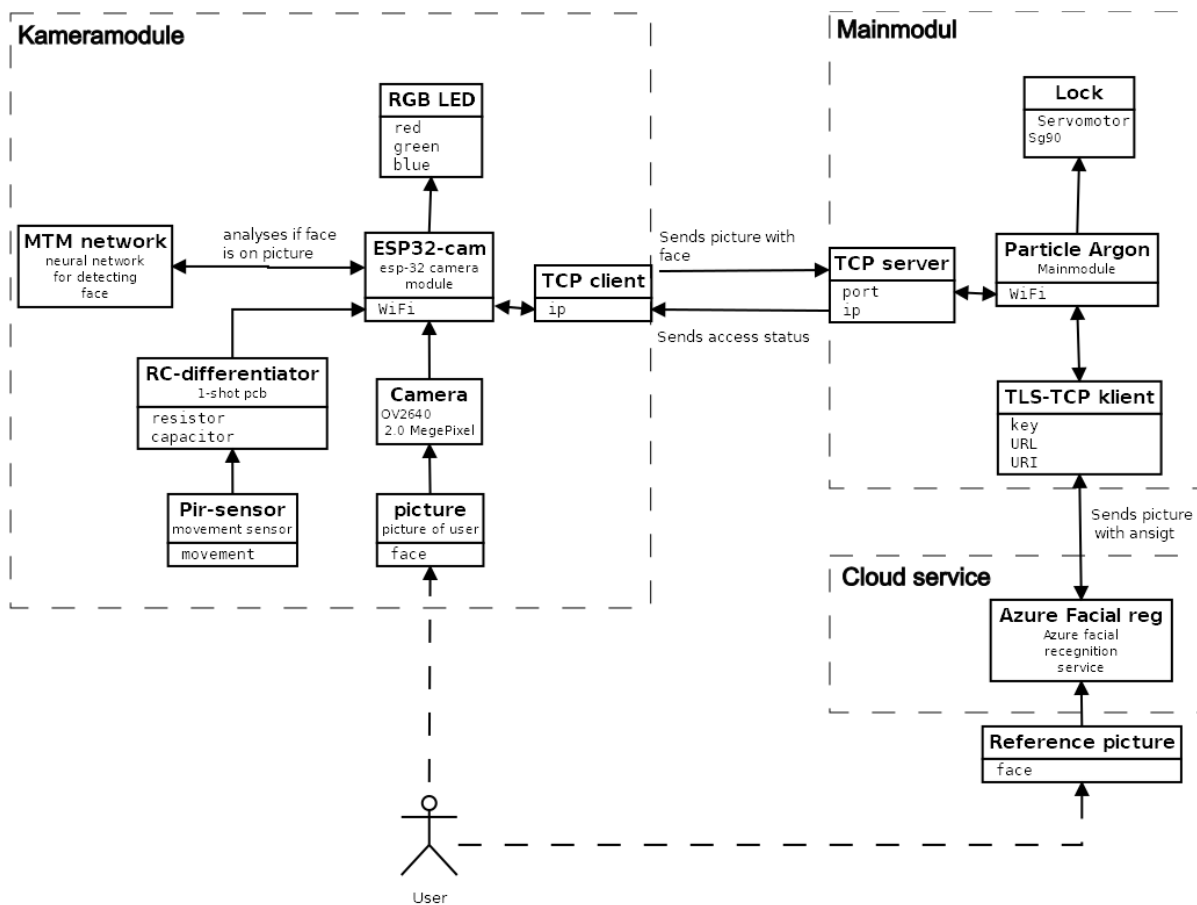
Arkitektur

Der skal udvikles et distribueret system bestående af et kameramodul, hovedmodul og en cloudservice. Kameramodulet står for at tage et brugbart billede af brugeren, hvorefter dette sendes til hovedmodulet. Kameramodulet står også for at visualisere adgang med en RGB-led. Hovedmodulet står for at sende det modtagne billede til cloudservicen og åbne låsen, afhængig af cloudservicens svar. Cloudservicen sammenligner det modtagne billede med et referencebillede, hvorefter der sendes et respons til hovedmodulet, om personen på billedet er magen til personen på referencebilledet.



Figur 12 - diagram over komponenter

Pakkes komponenterne yderligere ud fra Figur 12, giver det en domænemodel over system-to-be. Domæne modellen giver et overblik over adfærd og data.

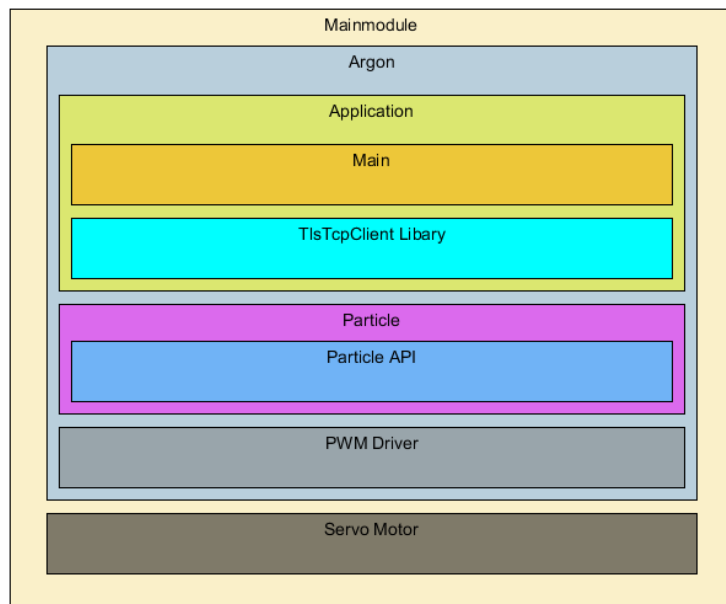


Figur 13 - Domain model

Da systems adfærd og data er beskrevet, kan systemets funktionalitet kortlægges og designes.

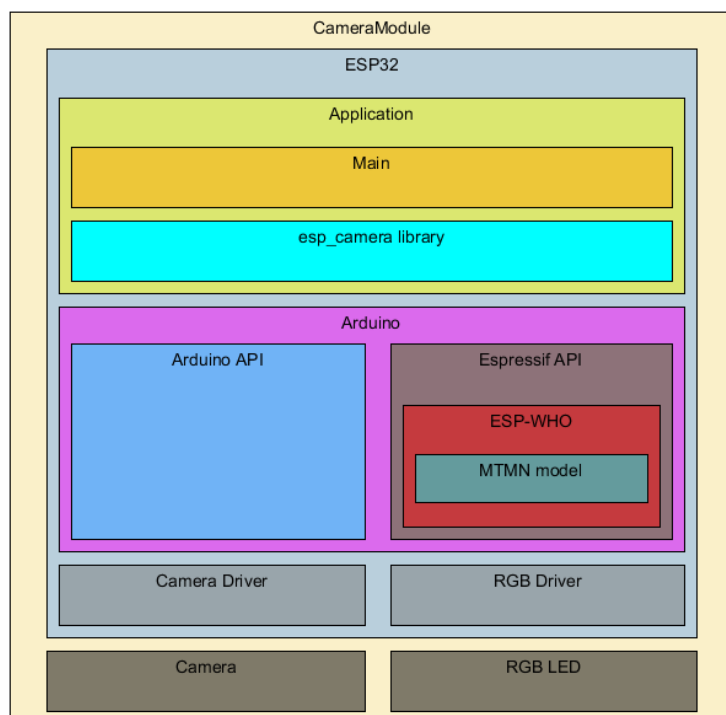
Til at skabe et overblik over arkitekturen på de forskellige moduler, er der udarbejdet to arkitektur-tegninger der illustrer opbygningen for hhv. hovedmodul og kameramodul.

Generel arkitektur hovedmodul



Figur 14 - Arkitektur diagram Hovedmodul

Generel arkitektur kameramodul



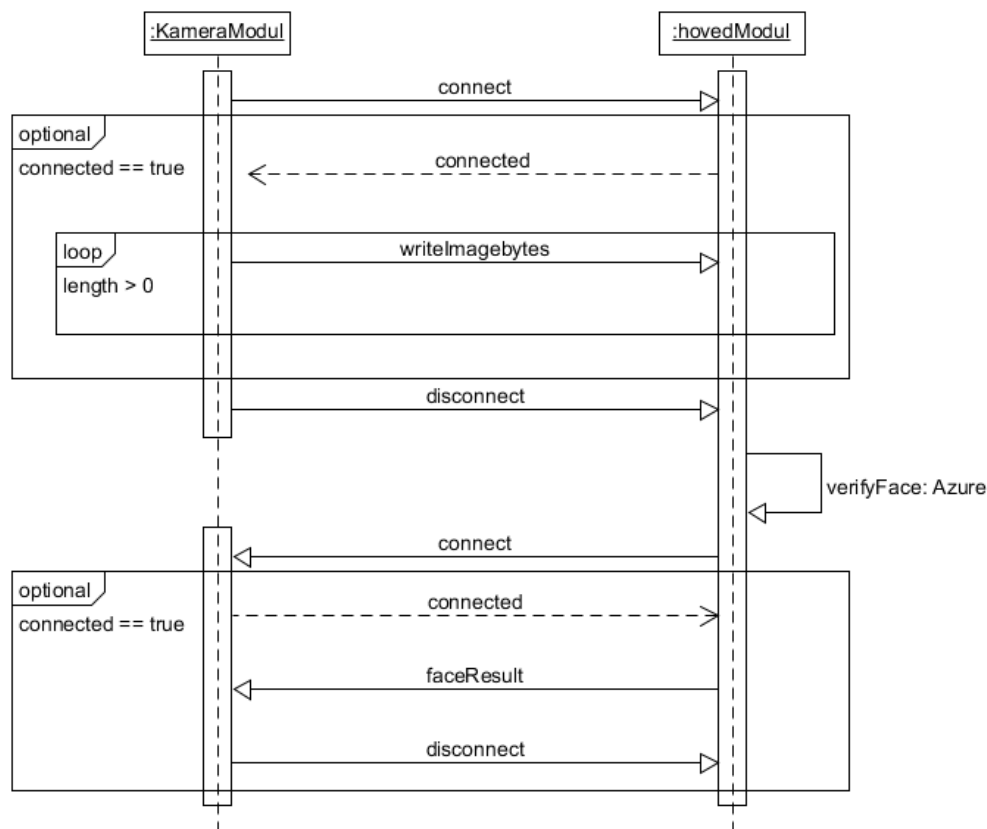
Figur 15 - Arkitektur diagram Kameramodul

Design

I følgende afsnit vil designet af moduler, klasser og funktioner beskrives. Designet vil blive beskrevet i sådan en grad at det er muligt at implementere modulerne efterfølgende.

Kommunikation mellem Moduler

Kommunikationen mellem modulerne sikre, at der kan sendes billeder fra kameramodul til hovedmodul hvor der vil blive foretaget ansigtsgenkendelse. Når hovedmodulet har foretaget en ansigtsgenkendelse, skal kameramodulet have besked omkring resultatet for a kunne notificere brugeren. Figur 16 viser hvordan en kommunikation mellem modulerne ser ud.



Figur 16 - netværks kommunikation

Kameramodul

Kameramodulets funktion er at tage billeder af personer som ønsker adgang, billederne sendes til hovedmodul, hvor der foretages yderligere behandling af billederne i form af ansigtsgenkendelse. Modulet er baseret på et ESP32 board [08].

Kameramodul er trådløst forbundet til samme netværk, som hovedmodul, hvilket gør det muligt at placere de to moduler hver for sig. Dette kunne være på modsatte sider af en dør, der ønskes adgangskontrol til vha. ansigtsgenkendelse. At kameramodul kan placeres uafhængigt af hovedmodul, stiller krav til forsyningen, hvilket også er årsagen til ønsket om en løsning med batteriforsyning.

At kameramodul er tiltænkt til at skulle forsynes fra et batteri, stiller krav til at designet er strømbesparende. For at tilfredsstille kravet om at være strømbesparende, benyttes der, hvor det er muligt, sleep-funktioner.

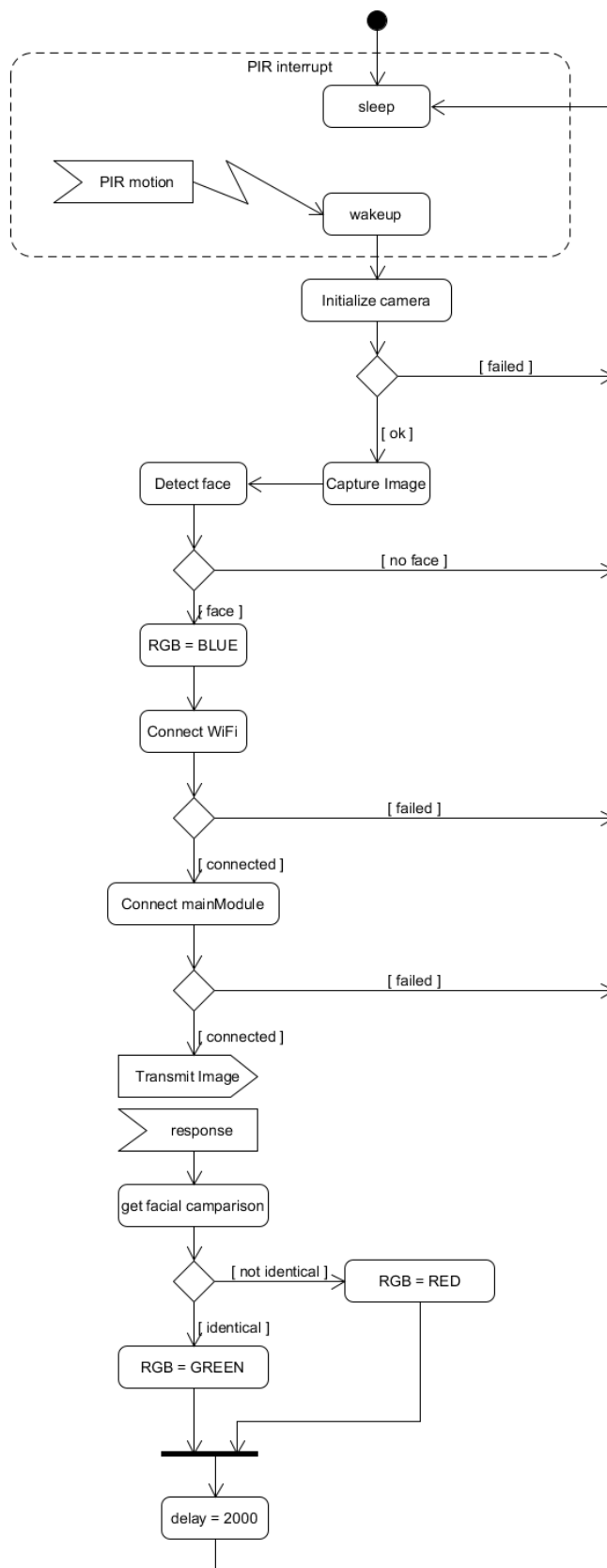
Til at vække systemet, så der kan tages et billede, bruges en PIR-sensor [Valg af hardware]. PIR-sensoren aktiveres når der opfanges bevægelse foran kameramodul.

For at reducere netværkskommunikation vil der på kameramodul blive benyttet hvad der kaldes edge computing.

Dataprocesseringen består i at frasortere billeder uden ansigter. Dette sker vha. et neuralt netværk [12] til at foretage ansigtsdetektering.

Til at signalere til brugeren omkring status på ansigtsgenkendelsen benyttes der lys. Kameramodul har en RGB-LED, som bruges til at signalere forskellige stadier af ansigtsgenkendelsen.

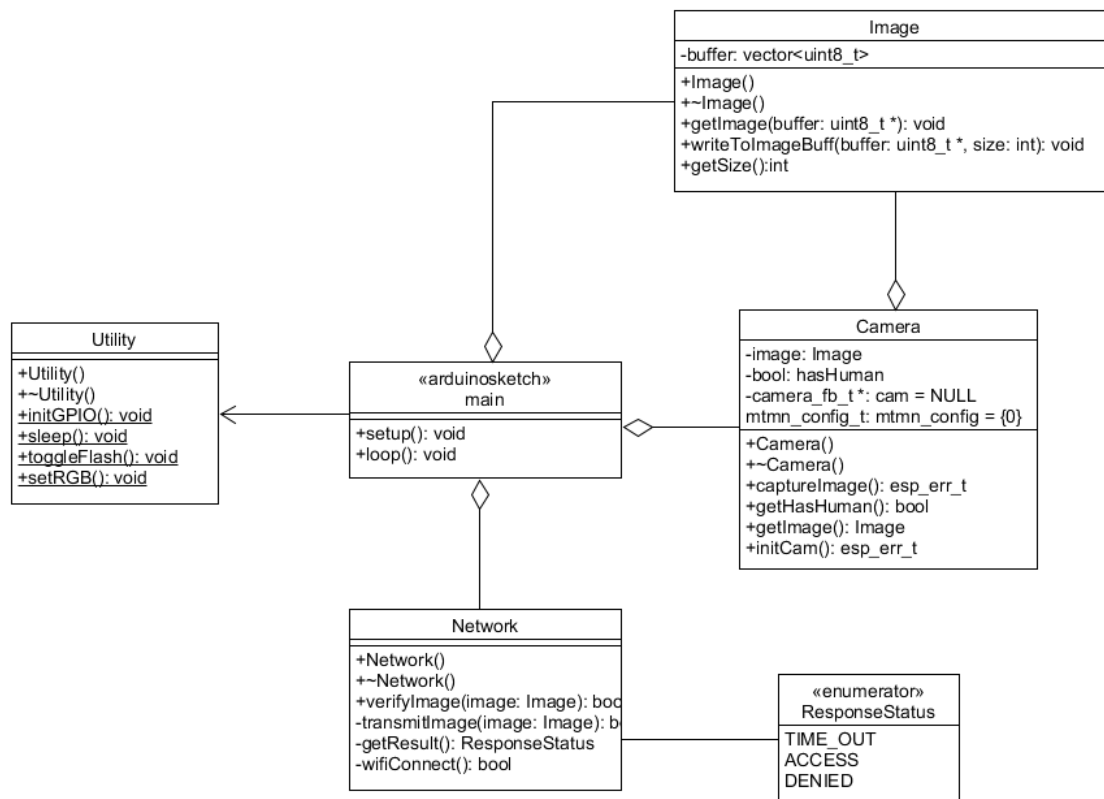
Til at beskrive kameramodul og forstå flowet er der udarbejdet et aktivitetsdiagram, se Figur 17 - Aktivitetsdiagram Kameramodul. Diagrammet viser flowet fra modul aktiveres til et ansigt enten er afvist eller godkendt.



Figur 17 - Aktivitetsdiagram Kameramodul

Til at illustrere sekvenserne involveret når kamera modulet vågner, er der lavet et sekvensdiagram, se [33]. Diagrammet giver et detaljeret overblik over kameramodulet. Det er vigtigt at notere sig hvordan modulet altid har fokus på at gå i sleep-tilstand når der ikke er noget at udføre.

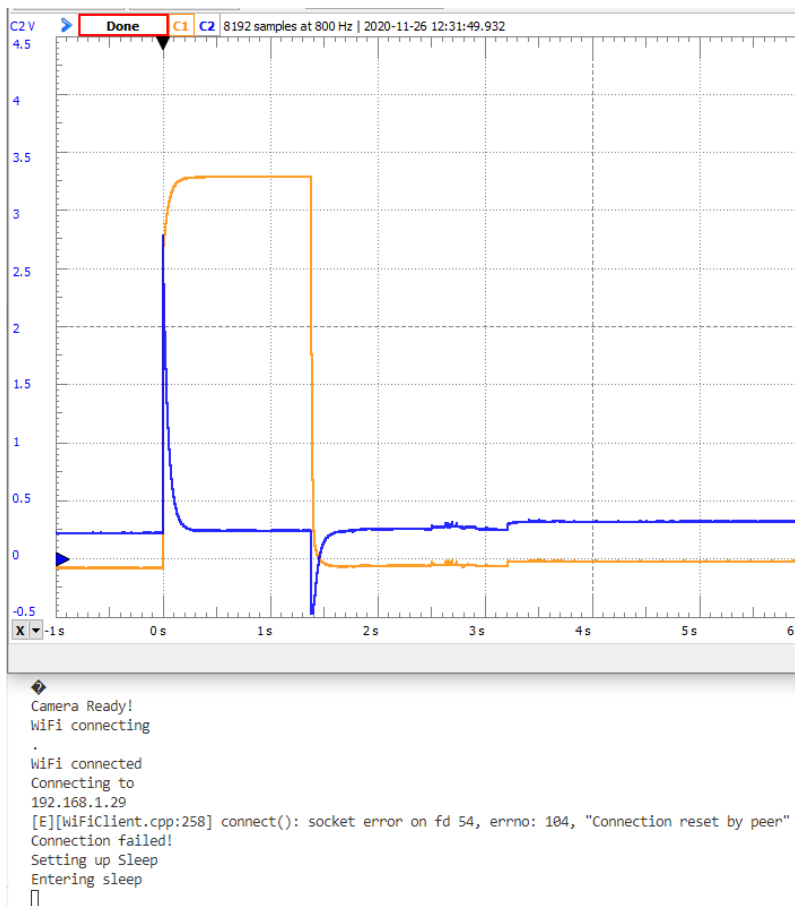
Ud fra tidligere analyse er der designet et klassediagram, sammen med denne og sekvensdiagrammet er rammerne opsat for hvordan kameramodulet kan implementeres.



Figur 18 - klassediagram Kameramodul

RC-differentiator

For at imødekomme at kameramodulet vågner fra sleep to gange, pga. PIR-sensorens lange output, er der designet en RC-differentiator, for at undgå for lange pulser på ESP32-CAMs GPIO pins. PIR-sensoren udsender et højt signal ved aktivitet, der er længere end handlingen der udføres af kameramodulet. Det giver en ekstra aktivering af kameraet. Ved at indsætte et RC-filter i signalvejen, overføres der kun spænding i en spike, når der kommer et niveau-skifte. Dette er blevet simuleret og resultatet samt schematic kan ses i bilag [32].



Figur 19 - Måling af RC-differentiator. Orange = PIR-output, Blå GPIO-input

I Figur 19 ses en måling efter RC-differentiatoren er blevet implementeret. Det ses, at kameramodulet ikke længere kommer ud af sleep 2 gange.

Udregninger for dette kan ses i bilag [13].

RGB-board

For at undgå at trække for meget strøm igennem LED'erne eller ud af GPIO-portene undersøges disses maximale værdier.

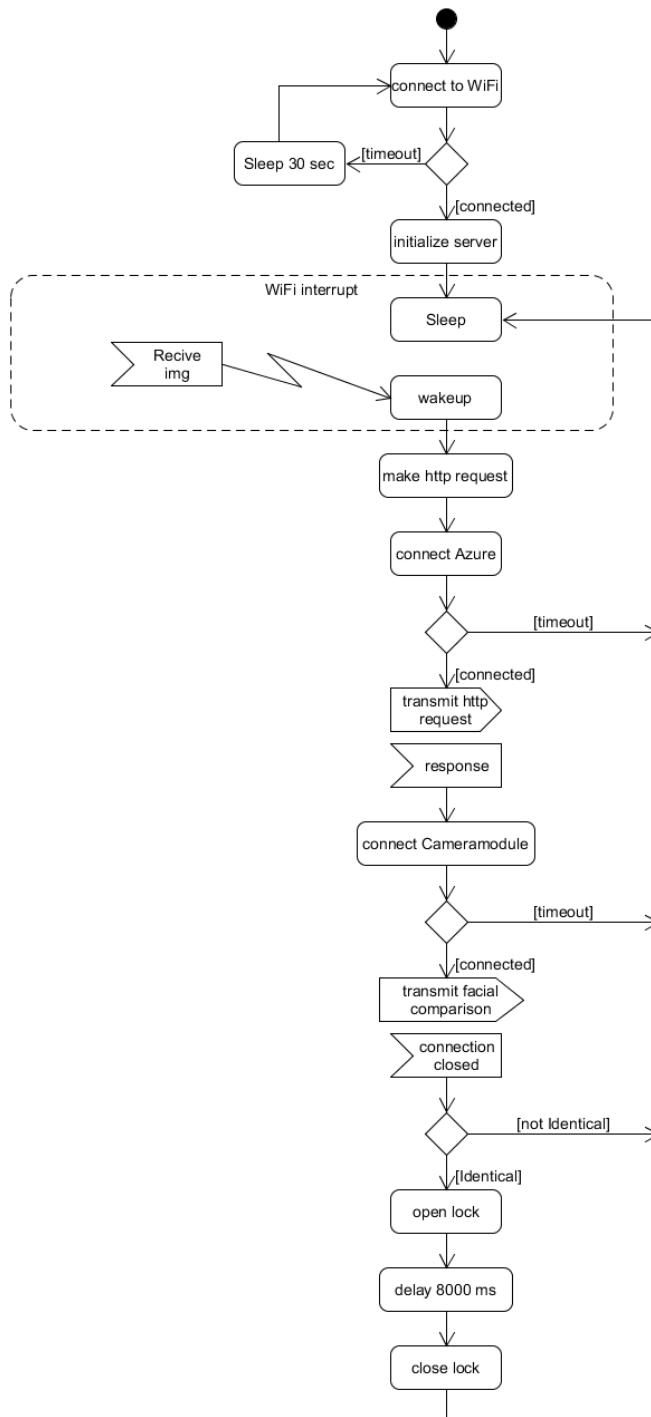
	Max current	Unit
RGB-LED	25	mA
ESP32-CAM GPIO output	40	mA

Figur 20 - Absolutte maksimum værdier for LED og ESP32-CAM

ESP32-CAM GPIO-output-voltage er 3.3V. Ud fra dette er formodstandens minimumsværdi fundet til at være 52Ω [13]. For at spare mere strøm er der brugt 100Ω til grøn og blå, og 180Ω til rød. Dette giver en tilsyneladende ensartet lysstyrke og trækker mindre strøm end hvis man brugte minimumsværdien.

Hovedmodul

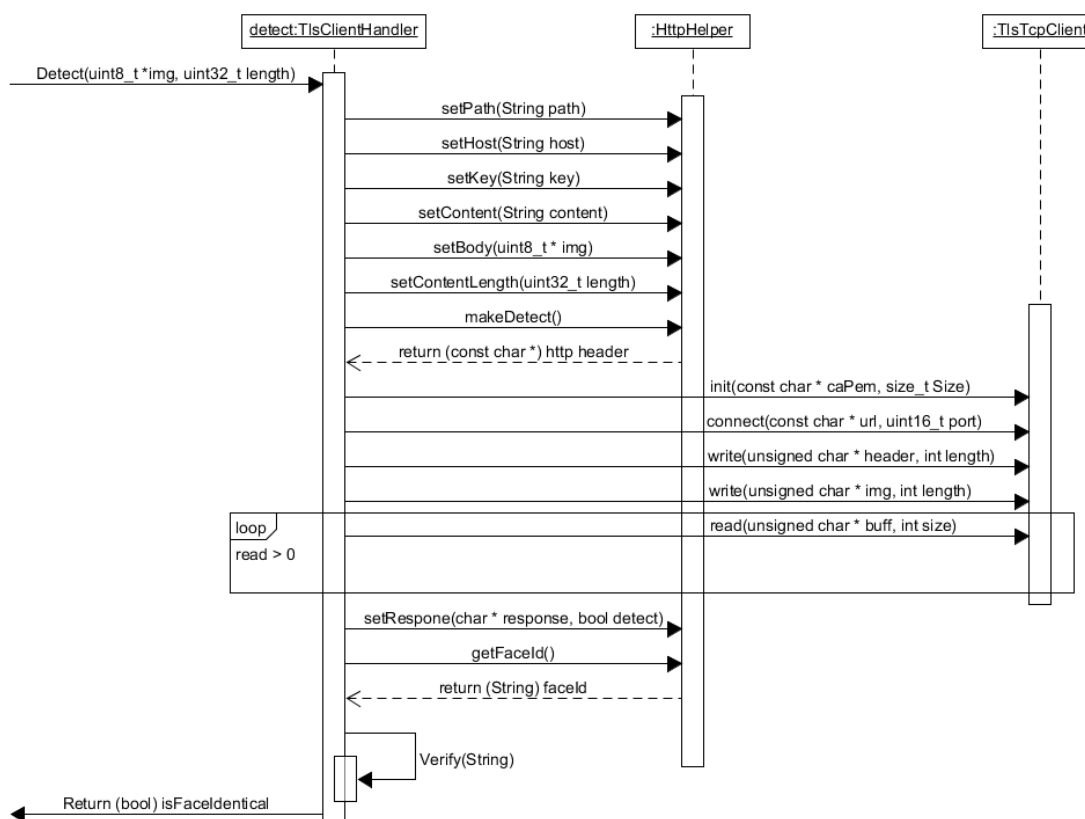
Hovedmodulets primære opgave er at låse op for brugeren, hvis brugeren er i systemet. Derudover fungerer hovedmodulet som gateway imellem kameramodulet og Azure facial recognition service. På Figur 21 kan flowet af systemet ses.



Figur 21 - aktivitetsdiagram over Hovedmodul

Hovedmodulet skal modtage et billede fra kameramodulet, dette gøres relativ nemt med Particles API, hvor der oprettes en Server [14] og Client klasse [15].

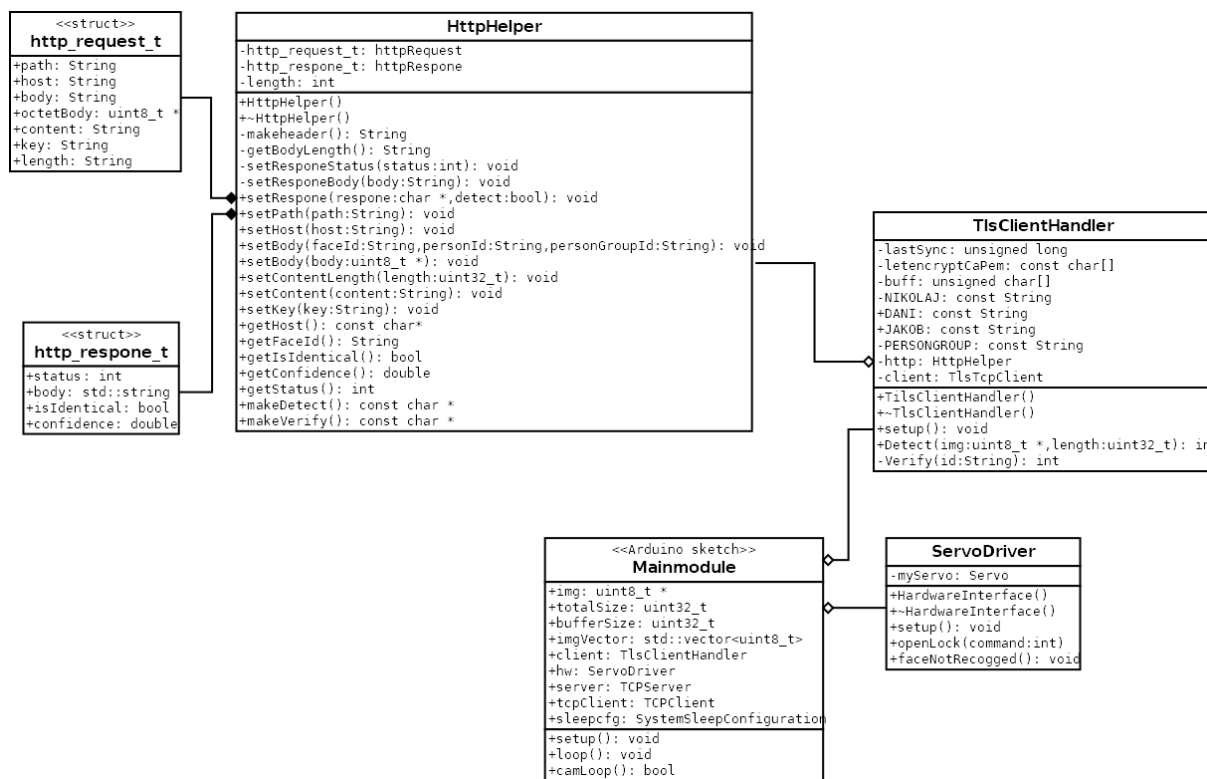
Herefter skal der forberedes til at sende data til Azure, hvor der skal laves et http kald. Det er muligt at lave http kald vha. Particles cloud platform, men denne understøtter ikke binary octet streams. Derfor er der designet en HttpHelper klasse, der står for at parse http kald og svar. Forbindelsen til Azure oprettes via det eksterne bibliotek TlsTcpClient [16] der er bygget op omkring mbedTLS [17] og Particles TCPClient. For at samle tls klienten med klassen HttpHelper er der designet en TlsClientHandler, der står for håndteringen af HttpHelper og kommunikationen imellem Azure.



Figur 22 - sekvens diagram over detect

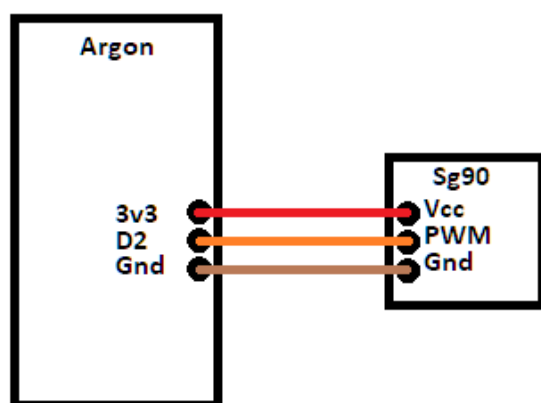
På Figur 22 kan det ses hvordan TlsClientHandler interagerer med de 2 klasser. For at sammenligne et ansigt skal Azure først detektere et ansigt, hvorefter der returneres et id. Dette id kan efterfølgende sendes til sammenligning med en person, allerede gemt i Azures database. Dette gøres med kaldet Verify, Verify kaldet har samme opbygning som Detect og er derfor ikke inkluderet her. nærmere beskrivelse af Verify kan findes i bilag [18].

Ud fra tidligere analyse er der udarbejdet et klassediagram.



Figur 23 - klassediagram

Til at simulere en lås bliver der brugt en SG90 servomotor. Dette er en lille hobby servomotor der max bruger 650 mA hvis den er blokeret [19]. Da Argon kan levere op til 1 A [20], er det muligt at forsyne motoren herfra. På Figur 24 kan det ses hvordan det er koblet sammen.



Figur 24 - SG90 hookup guide

Et mere dybdegående designafsnit af hovedmodulet kan findes under bilag [18].

Implementering

Følgende afsnit viser hvordan projektets hardware dele er implementeret. Derudover bliver tekniske dele for softwareimplementeringen kort gennemgået.

Udviklingsplatform

Til valg af udviklingsplatform har der været fokus på at kunne prototype hurtigt, der samtidigt kan benyttes sammen med de MCU der er valgt til at realisere system-to-be. Til udvikling på Argon benyttes der Particles egen platform [21], og til at udvikle til ESP32-CAM benyttes Arduino [22].

IDE & Versionskontrol

Som IDE er det valgt at bruge Visual Studio Code [23], dette skyldes at der findes gode extensions der gør det muligt både at kode til Particle og Arduino.

De benyttede extensions er:

- Particle Workbench [24]
 - Particle
- PlatformIO [25]
 - Arduino

Til versionsstyring er der brugt GitHub.

Sprog & compiler

Udviklingssproget er C++, da dette er understøttet på begge platforme og det giver bedre muligheder at arbejde objektorienteret. Compileren der benyttes, er GCC C++11.

Firmware

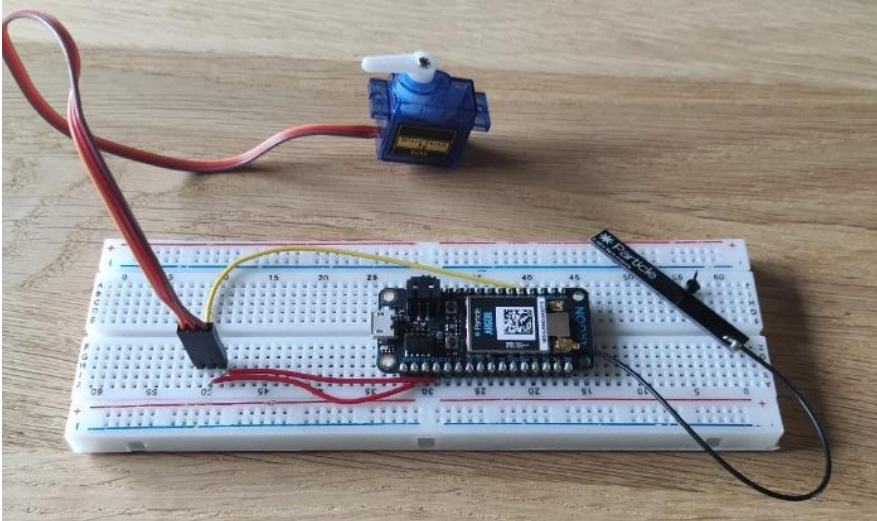
Firmware der kører på de forskellige enheder kan ses i Tabel 4.

Modul	Version	Placering
Hovedmodul	1.0	https://github.com/danimolsen1988/wifa_lock/tree/master/Sourcecode/mainModule
Kameramodul	1.0	https://github.com/danimolsen1988/wifa_lock/tree/master/Sourcecode/201203-133003-esp32cam

Tabel 4 - Firmware

Hovedmodul

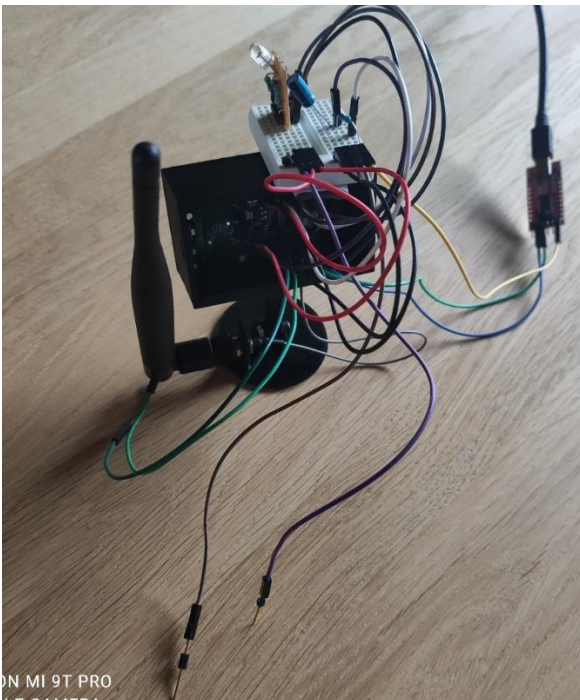
Hovedmodulet består af en Argon, antenne og servomotor. Antennen var en del af Argon pakken, dertil er det kun servomotoren der er tilsluttet.



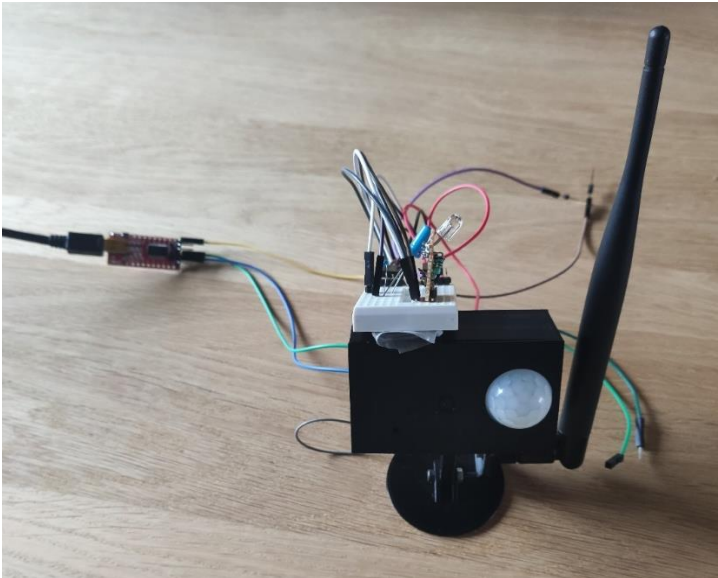
Figur 25 - hovedmodul HW

Kameramodul

Kameramodulet består af ESP32-CAM, antenne, RC-differentiator, RGB-Led board og en HC-SR501 PIR-sensor. Dertil er der 3d printet et hus til kameramodulet. I selve huset sidder ESP32-CAM og PIR-sensoren.



Figur 26 Kameramodul HW - bagfra



Figur 27 Kameramodul HW- forfra

Verifikation

Der er skrevet funktionelle tests til de vigtigste krav, de funktionelle test verificere op til flere krav ad gangen. Dette skyldes at der ikke er udført unit-tests, men visuelle bekræftelser af funktionalitet. Derfor har nogle krav været afhængige af at andre var implementeret.

Test	Status	Kommentar
Test [1]	Passed	Udført.
Test [2]	Passed	Udført.
Test [3]	Passed	Udført.
Test [4]	Passed	Testen er delvist en succes, Ansigts detekteringen er ikke perfekt og kræver gode forhold for at detektere et ansigt.
Test [5]	Passed	Udført, med lav fejlmargen, der tillades en mindre tolerance i forhold til afstanden.
Test [6]	Passed	Udført.
Test [7]	Passed	Udmåling af energiforbrug

Tabel 5 - testresultater

Alle udførte tests kan findes i bilag [26]

Ikke udførte krav

De krav der ikke er udført tests på, kan ses nedenfor i Tabel 6, samt årsagen for at der ikke er skrevet og udført test.

Tabel 6 - Ikke udførte krav

Krav	Årsag
Kameramodulet skal optage i høj nok opløsning til at foretage ansigtsgenkendelse og objektgenkendelse	Kravet kan tjekkes ved undersøge databladet for kameraet [27] og sammenligne med kravne fra ESP-WHO [28] og Azure [29]. Kameraet opfylder minimumskravene på hhv. 320x240 og 36x36.
Systemet skal automatisk oprette forbindelse mellem kameramodulet og hovedmodulet	Der oprettes altid ny forbindelse mellem moduler for at sende billeder, eksisterende tests [26], verificerer allerede at moduler kan kommunikere sammen.
Mens kameramodulet har afsendt et billede til hovedmodulet, skal kameramodulet vente i max 10 sekunder på verifikation.	Er ikke et key requirement, og der er derfor ikke skrevet en test.
Hvis kameramodulet vækkes af andet end et menneske skal billedet ikke sendes til hovedmodulet	Er delvist verificeret gennem anden test, og har derfor ikke egen test.
Hvis hovedmodulet ikke kan forbinde til WiFi/bluetooth skal modulet gå i error-tilstand, blink med LED	Particle API håndterer allerede fejlfinding vha. LEDS [30] og en af disse er manglende WiFi forbindelse

Diskussion af resultater

Det er lykkedes at opnå 30 dages batterilevetid, hvis der bruges 18650/21700 Li-ION batterier på minimum 4500 mAh. 6 måneder ville være mere belejligt for brugere. Dog har teamet gjort, hvad der har været muligt, for at mindske energiforbruget indenfor opgavens rammer.

Teamet er tilfredse med at det vil være muligt at drive kameramodulet fra et batteri over 30 dage. Der er blevet estimeret en strømforbrug på 4438 mAh baseret på teoretiske udregninger. Som er blevet påvist ved at udføre strømprofilering, som har vist et reelt forbrug på 4381 mAh.

Et positivt overraskende element er latenstiden fra et billede tages til der er givet adgang, der tager cirka 10-15 sekunder. Det var forventet at kommunikationen til cloudservicen ville tage væsentligt længere tid. Ligeledes var det forventet, at det ville tage længere tid at vække modulet fra sleep – dette var heller ikke tilfældet.

Mængden af tilgængelige services på internettet rettet mod IoT overraskede også teamet. Der er utroligt mange services, som tæller bla. Google, Amazon, Microsoft, IBM, Cisco og PTC. Mange af disse services handler om at gøre IoT mere tilgængeligt og gøre håndtering af IoT-løsninger nemmere.

Det neurale netværk teamet har implementeret er ikke helt så godt som det kunne være. For at der detekteres ansigter kræver det gunstige forhold for kameraet, således billedet bliver meget skarpt. Det kunne have været et ønske at ansigtsdetekteringen var mere konsistent end hvad der er oplevet.

Bedre kamera og en mere kraftig platform kunne være løsninger. Der findes nyere platforme med netop kerner specifikt til machine learning og AI opgaver, hvilket nok ville have givet et resultat tættere på forventningerne.

Konklusion

Det er lykkedes at producere en prototype, der fungerer som forventet, men hvor der stadig er plads til forbedringer. Teoretiske emner fra kurset er blevet anvendt i tilvejebringelsen af IoT-artefaktet. Der er udover underviste emner blevet undersøgt og anvendt nye emner i forløbet.

Opgaven var at få lavet et IoT-artefakt, som forbedrede et ikke internetforbundet produkt og samtidigt gav det yderligere værdi for brugeren.

Ved at omfavne IoT-principperne er det lykkedes at få et hverdagsprodukt gjort intelligent og "connected", hvor det samtidigt har bidraget positivt til brugen af produktet.

Alle deltagere har fået læring ud af dette projekt. Specielt indenfor strømbesparelse, at arbejde med kommunikation imellem devices – både lokalt og via internettet – og dertilhørende protokoller.

Typisk har IoT-devices det tilfælles, at de kører på batterier. Derfor er teamet godt tilfredse med, at det har været muligt at overholde kravet omkring 30 dages levetid på én opladning.

Teamet er ikke helt overbeviste omkring neurale network på små enheder – endnu! Det er på vej, men lige nu er det ikke på et funktionelt niveau.

Processen for dette projekt er alle medlemmer af teamet enige om har været tilfredsstillende. Samarbejdet har fungeret og vi har nået de mål vi satte os for.

Fremtidigt arbejde

Det er lykkedes at opnå det forventede, dertil er der gjort overvejelser om de ting der kunne forbedres og tilføjes i fremtiden.

1. Detektering af bevægelse kunne håndteres bedre

PIR-sensoren kan ikke indstilles præcist som teamet gerne så den indstillet. Hvis der var tid, kunne teamet have designet egen sensor, der kørte præcist som ønsket

2. Detektering af ansigter kunne være behandlet over en serie af billeder, for at gøre det nemmere for brugeren at få adgang

Hvis ansigtsgenkendelsen blev foretaget over en serie af billeder, fremfor et enkelt billede, ville der være en større sandsynlighed for, at brugeren ville komme ind i første forsøg.

3. Ansigts sammenligning kunne være behandlet lokalt

Ved at køre data op på internettet, er der mulighed for at flere services kan være utilgængelige, hvilket vil kunne betyde udfald af produktets funktion. Endda udfald, som teamet ikke kan gøre noget ved.

4. Håndtering af reference billeder

Dette kunne gøres vha. en app på brugeres smartphones eller direkte fra kameramodulet. Dette ville gøre brugeroplevelsen markant bedre. Det kunne også være en god idé at koble en app på systemet, så brugere kunne give adgang til personer via en smartphone.

Referencer

Id	Reference
[01]	esp-face/face_detection at master · espressif/esp-face (github.com)
[02]	Wi-Fi & Bluetooth MCUs and AIoT Solutions I Espressif Systems
[03]	https://blackboard.au.dk
[04]	http://eudp.dk/index.php/EARS_Requirement_Capture
[05]	"Strømodregninger&målinger_bilag.pdf" i bilag
[06]	https://docs.particle.io/argon/
[07]	https://console.particle.io/devices
[08]	https://www.espressif.com/en/products/socs/esp32
[09]	https://www.mpja.com/download/31227sc.pdf
[10]	"Protokoller&beskedformater.pdf" i bilag
[11]	https://docs.particle.io/tutorials/device-os/bluetooth-le/
[12]	https://github.com/espressif/esp-face/tree/master/face_detection
[13]	"RC_og_Formodstand_udregninger.pdf" i bilag
[14]	https://docs.particle.io/reference/device-os/firmware/argon/#tcpserver
[15]	https://docs.particle.io/reference/device-os/firmware/argon/#tcpclient
[16]	https://github.com/hirotakaster/TlsTcpClient
[17]	https://tls.mbed.org/
[18]	"Hovedmodul_bilag.pdf" i bilag
[19]	https://opencircuit.shop/Product/TowerPro-SG90-9G-micro-servo-motor-180
[20]	https://docs.particle.io/reference/hardware/pin-info/?m=details&dev=Argon&show=all&sort=name
[21]	https://www.particle.io
[22]	https://www.arduino.cc/
[23]	https://code.visualstudio.com/
[24]	https://www.particle.io/workbench/
[25]	https://platformio.org/
[26]	"Test&Verifikation_bilag.pdf" i bilag
[27]	"OV2640_datasheet.pdf" i bilag
[28]	https://github.com/espressif/esp-who
[29]	https://docs.microsoft.com/en-us/rest/api/cognitiveservices/face/face/detectwithurl
[30]	https://docs.particle.io/tutorials/device-os/led/argon/
[31]	"HardwareSpecifikationer.pdf" i bilag
[32]	"Schematic_Simulering_RC-differentiator.pdf" i bilag
[33]	"kameramodul_bilag.pdf" i bilag
[34]	https://github.com/danimolsen1988/wifa_lock