

E5IOT

Internet of Things

Automatisering af hønsehus



5. semester 2020

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

Deltagere

Studienummer	Navn	Initialer
201808044	Lasse Greve Rasmussen	LGR

Undervisere:

Morten Opprud Jakobsen

Klaus Kolle

Resume

Denne rapport gennemgår hvordan der er arbejdet med en IoT løsning til automatisering af døren i et hønsehus.

Rapporten beskriver analyse, design, implementering og verifikation af projektet.

Indhold

Resume	1
Forord	4
Begreber og forkortelser	5
Indledning	6
Projektbeskrivelse	7
Løsning af krav fra projektbeskrivelse	7
Analyse	8
Krav	8
Platform	9
Servomotor	10
Phototransistor	10
Design	11
Tilstande	11
Feedbacksignaler	11
Aktivitetsdiagram	12
Styring efter solopgang og solnedgang	13
Energiforbrug	14
Sleep	16
Interface	18
Implementering	19
Hardwareopbygning	19
kode	20
Verifikation (LGR)	21
Test	22
skrivebordstest	22
Opsætning:	22
Test:	23
Vurdering	24
Field test	25
Vurdering	25
Resultater	26

Resultat af skrivebordstest	26
Resultat af field test	27
Opsamling af resultater	28
Konklusion	29
Fremtidigt arbejde	29
Referenceliste for samlet dokumentation.....	30
Appendix	30

Forord

Denne rapport er dokumentation for det selvvalgte projekt der er arbejdet med under faget 'E5IOT' der omhandler konceptet 'Internet Of Things'.

Hvem er jeg:

Lasse Greve Rasmussen

Uddannelsessted: Aarhus Universitet, Herning

Studieretning: Ingeniør - Elektronik

Semester: 5. semester

Undervisere: Morten Opprud Jakobsen, Klaus Kolle

Review foretaget af: Malthe Brauer-Nielsen

Afleveringsdato: 11. december 2020

Bedømmelsesdato: 21. december 2020

Begreber og forkortelser

Begreb	Definition
høsehus	Høsehuset består af selve høsehuset samt en tilhørende lille høsegård. Det er hele denne enhed der lukkes og åbnes for
Døren	Den dør der går ind til den lille høsegård der er i forbindelse med høsehuset
Particle	Firmaet bag udviklingsmiljøet
Argon	Udviklingsboard fra Particle
Particle Consol	Online consol til styring og overvågning af enheder fra Particle
IFTTT	Online service 'If This Then That' [ref05]
Thingsspeak	Online service til at logge data
Weather Underground	Vejrservice der bl.a. tilbyder information om solopgang og solnedgang
API	Application Programming Interface, grænseflade mellem software services

Indledning

Formålet med denne opgave er at undersøge, hvordan man kan lave en internetforbundet løsning der automatiserer døren i et hønsehus. Løsningen skal kunne åbne døren ved solopgang, og efter solnedgang skal den lukke døren igen.

Høns er meget pålidelige i at de altid søger ind i deres hus når det begynder at blive mørkt. Det er derfor ikke nødvendigt at kontrollere om hønsene er inde, når blot man sikrer at døren først lukkes efter at det er blevet mørkt.

Formålet med at holde hønsehuset lukket om natten, er at sikre at der ikke går skadedyr i hønsenes foder.

Systemet skal være baseret på data om dagens solopgang og solnedgang der ligger online, så enheden der styrer systemet skal være forbundet til internettet.

Projektbeskrivelse

Her er uddrag fra den stillede opgave.

Purpose

The purpose of this project is to apply certain theoretical topics, from the course, onto a specific project.

The project runs through the entire semester.

You must design an IoT artefact, composed of embedded hardware, embedded software, sensors, and actuators.

Your IoT artefact shall be capable of performing a meaningful task, better, than a similar device, that is not connected to the internet.

That could be anything from "a smart plant watering systems, that takes into account the weather forecast, sunshine hours, etc, " to a "streaming music device that plays music depending who is present" or perhaps an intelligent industrial sensor, or something completely different.

Requirements

Functional requirements

1 The device must be able to connect to the internet

1.1 Internet connection shall be via WIFI

1.2. The device should preferably be able to connect to AU's "AU Gadget network"

2 Your device must be able to read data from a connected sensor, local to the device

2.1 a sensor can be anything that quantifies a physical measure, into an electrical signal, such as temperature, light, humidity, presence, movement, magnetism, pollution, etc.

3 Your device must be able to control an actuator

3.1 An actuator can be anything that translates an electrical signal into a physical quantity, such as, motors, servos, valves, heaters, displays, lamps, etc.

4 Your device must be capable of using data from a web service, to augment "what it does", this could be weather data, traffic data, stock prices, twitter feeds, emails, rss-feeds or something different.

5 Your software and hardware design must be shared

5.1 You must create a public **github** account, and add relevant project files here

5.2 Hardware documentation, schematics, datasheets and pcb layouts are to be uploaded in pdf format

5.3 Software files are to be uploaded in raw source code format, e.g. **.C, CPP, .h, .py**, etc.

Technical requirements

1 The technical platform can be a suited embedded platform of your choice, e.g. the Particle Photon, an ESP8266, a raspberry pi, beagle bone black or similar.

1.1. The platform shall have Wifi connectivity

1.2. The platform shall have available digital or analog I/O to connect sensors and actuators

Løsning af krav fra projektbeskrivelse

I rapporten vil der blive gennemgået analyse, design og implementering. I Tabel 1 ses korte beskrivelser af hvordan kravene bliver løst.

	Funktionelle krav
1	En Argon er sat op til at forbinde til AU Gadget via WIFI
2	Lysniveau måles via phototransistor Der gives brugerinput via trykknapper
3	En servomotor kontrolleres via et pwm signal fra argon Der gives bruger feedback via en integreret RGB diode på argon
4	Der anvendes services fra 'Weather Underground' via IFTTT
5	Kode og dokumentation er løbende blevet opdateret på github [ref01]
	Tekniske krav
1	Projektet er implementeret på en Particle argon der kan forbindes via wifi. Den har både digitale og analoge I/O til rådighed

Tabel 1 - løsninger på krav fra projektbeskrivelse

Analyse

For at klarlægge hvad systemet skal kunne er der lavet krav der definerer dette. Kravene er kategoriseret og formuleret efter EARS metoden.

Krav

ufravigelige

- Systemet skal kontrollere en servomotor der kan åbne og lukke døren i et hønsehus.
- Systemet skal have informationer om solopgang og solnedgang via internettet.

Handling

- Når solen er stået op, skal døren åbne.
- Når solen er gået ned, skal døren lukke.

Tilstand

- Mens systemet ikke har nogen funktionalitet, skal systemet bruge så lidt strøm som muligt.

Uønsket adfærd

- Hvis der ikke er internetforbindelse, skal systemet styres ud fra lysmålinger målt med en phototransistor.

Valgfri

- Systemet kan indeholde styring af belysning i hønsehuset
- Systemet kan være drevet af et batteri der om dagen oplades af en solcelle

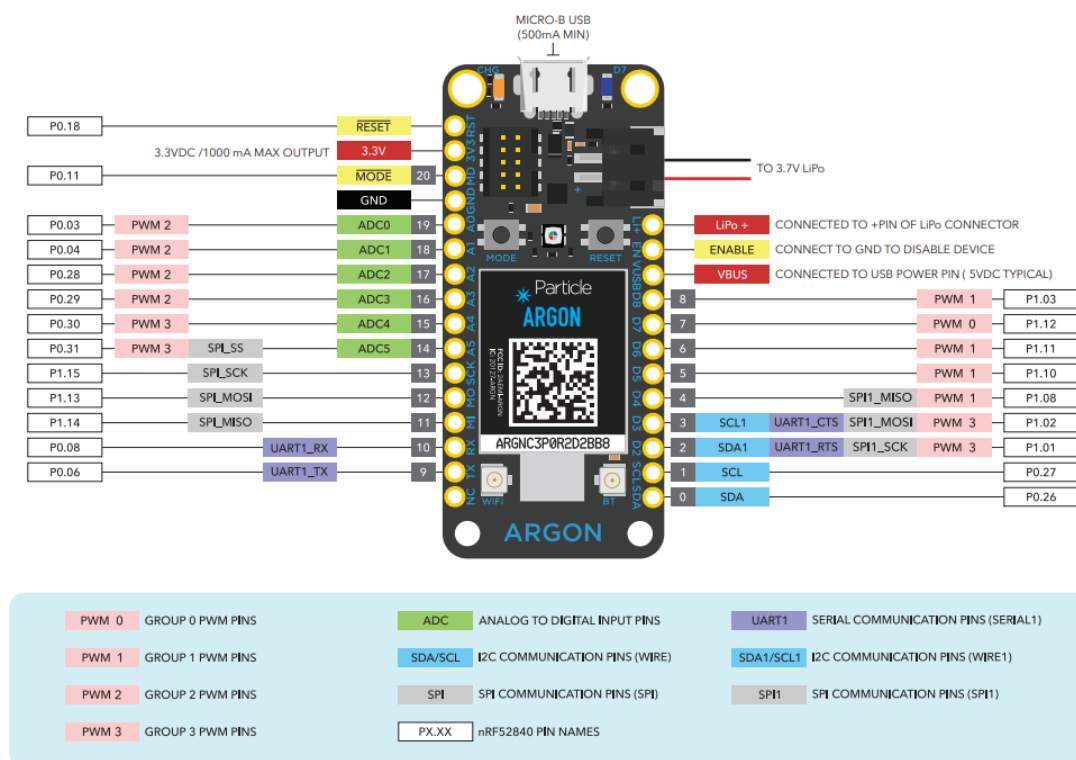
Platform

For at kunne løse opgaven er der følgende krav til den platform der skal anvendes:

- Minimum én ADC til phototransistor.
- Minimum én PWM udgang til at kontrollere servomotoren.
- Mulighed for internetforbindelse via WiFi.
- Systemet stiller ikke store krav indenfor proces og hukommelse som ressourcer.

I forbindelse med faget er der anvendt en Particle Argon, der er en god platform med mange muligheder til IOT løsninger. Det vurderes at denne platform også vil være god til projektet, da den overholder ovenstående krav til en platform.

I Figur 1 ses et overblik over en Particle Argon. Her ses at der er flere muligheder for både PWM og ADC, og boardet kan forbinde til internettet via WiFi. Particle Argon bygger på en 64MHz ARM Cortex-M4F 32-bit processor og har 1 MB flash, og 256 KB RAM [3]. Dette er rigeligt til de relativt simple opgaver der skal løses i dette projekt.



v1.0

Figur 1 - pinout for Particle Argon [ref03]

Servomotor

Da der er en servomotor af type HS755HB tilgængelig, undersøges det om denne vil kunne anvendes til projektet. Servomotoren er styret af et 50 Hz pwm signal, hvor længden af positiv puls bestemmer motorens position.

Udpluk fra datablad [ap03]:

- Forsyningsspænding: 4.8V - 6.0V
- Strømforbrug: 230mA til 1500mA (4.8V)
- Moment: omkring 10kg.cm svarende til ca. 1Nm
- Signal: PWM 1.5 ms = neutral
- Arbejder i et område på 180°

Ud fra disse data ser servomotoren ud til at være egnet til opgaven. Dørens konstruktion er meget let, og der skal derfor ikke mange kræfter til at åbne og lukke den.

Desuden betyder forsyningsspændingen at man vil kunne drive argon boardet og servomotoren fra samme forsyningskilde på 5 V.

Phototransistor

Til at måle lysstyrken anvendes en phototransistor der er tilgængelig i ellab. Til målingen anvendes en opstilling inspireret af eksempel fra particle [ref08].

Design

I dette afsnit vil de overordnede tanker om systemets design være beskrevet. Der vil blive beskrevet hvordan styringen er delt op i tilstande og hvordan der i den sammenhæng arbejdes med bruger feedback. Systemets processer og forbindelser bliver visualiseret. Desuden findes en metode til styring efter solopgang og solnedgang, og det undersøges hvordan systemets strømforbrug kan optimeres.

Tilstande

Systemet bygges op omkring tre tilstande;

- Normal: systemet afventer funktionskald i forbindelse med solopgang eller solnedgang, eller interrupt for manuel kontrol. Mens systemet er i normal, vil en tæller stå og tælle op. Hvis der går 20 timer uden kald for solopgang eller solnedgang, antages det at der er sket en fejl, og systemet går i 'sensor control'.
- Sensor control: dette er en sensorstyret tilstand, der fungerer som fejlhåndtering. Systemet måler via en phototransistor lysstyrke, og baseret på en midlet værdi vurderes det om døren skal åbnes eller lukkes. Dette fortsætter indtil der igen er blevet gennemført et online funktionskald, eller 'manual mode' aktiveres.
- Manual mode: Med to trykknapper kan man manuelt åbne eller lukke døren, hvis dette gøres, forbliver systemet i 'manual mode' indtil en tredje tryknap for 'manual mode off' aktiveres.

Feedbacksignaler

For at give brugeren feedback styres en RGB diode så forskellige farver viser hvilken tilstand der er aktivt.

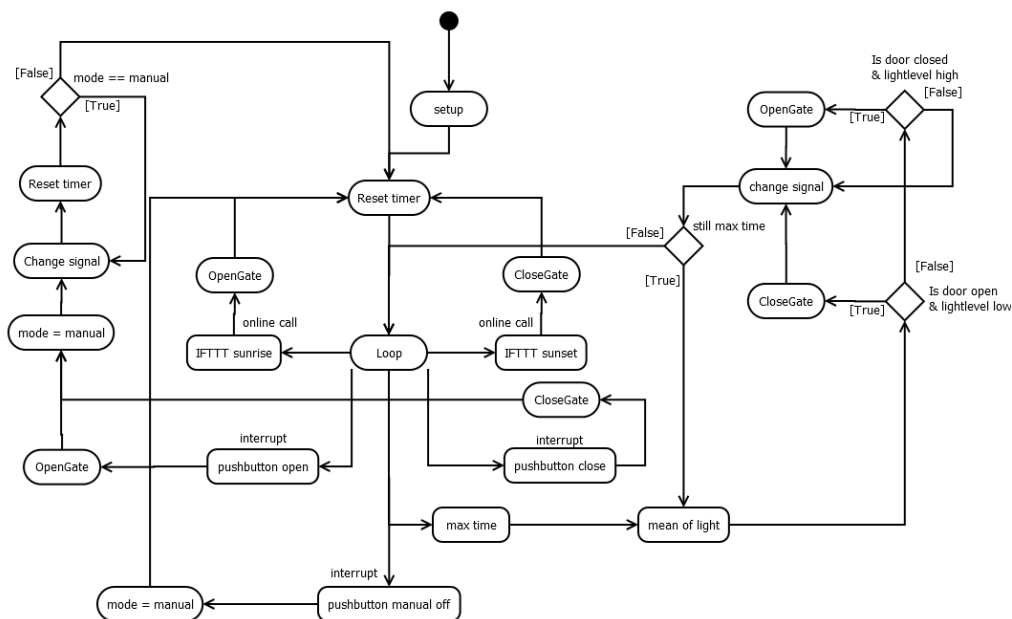
- Normal: dioden lyser grøn som det er vist på Figur 9.
- Sensor control: dioden lyser rød for at indikere at der er 'fejl' i systemet. Dioden lyser ikke konstant, da den vil være slukket under måling af lysniveau. Dette gøres for ikke at forstyrre målingen.
- Manual: dioden lyser blå.

Aktivitetsdiagram

For at klarlægge handlinger i systemet laves der et aktivitetsdiagram som ses i Figur 2. Aktivitetsdiagrammet består af en blanding af almindelige processer, online funktionskald, og interrupt styrede processer.

Systemet starter op og går i loop indtil en af følgende hændelser sker:

- Døren åbnes eller lukkes på baggrund af funktionskald fra IFTTT. Systemet går tilbage i loop.
- Døren åbnes eller lukkes via et interrupt fra en trykknop, systemet går i 'manual mode' og bliver der indtil tilstanden skifter igen. Når tilstanden skifter, går systemet tilbage i sit loop.
- Timeren når max time, og systemet går i tilstanden 'sensor mode'. Lysniveau måles og bliver midlet, og det vurderes om en handling er krævet. Systemet forbliver i denne tilstand til timeren nulstilles. Når timeren nulstilles går systemet tilbage i sit loop.



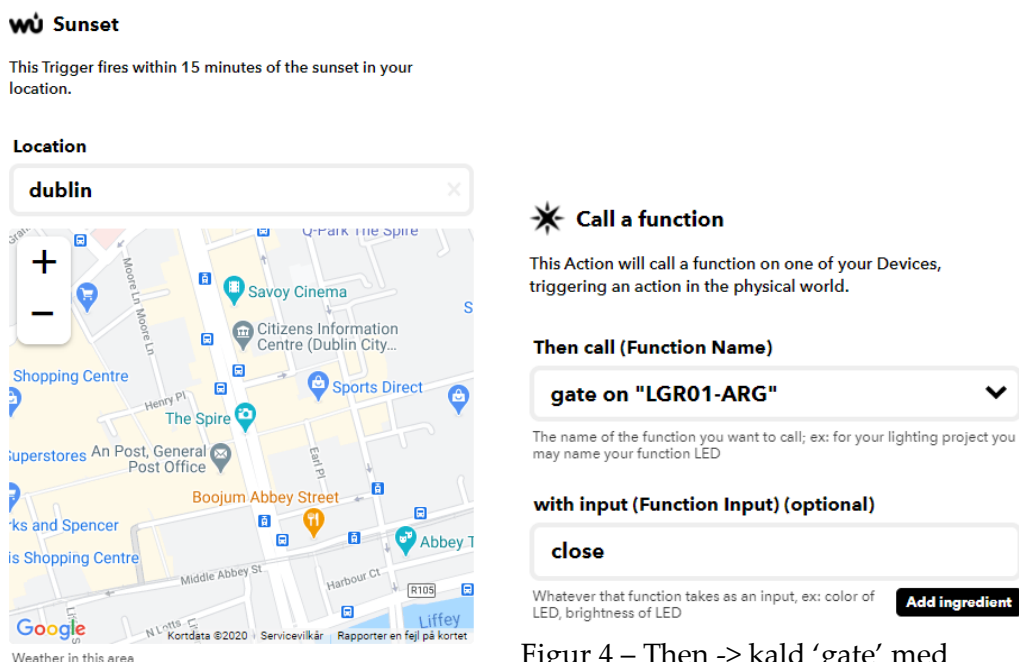
Figur 2 – aktivitetsdiagram

Styring efter solopgang og solnedgang

For at styre døren efter solopgang og solnedgang vælges det at anvende to services fra IFTTT.com. IFTTT er en online tjeneste der kombinerer mange forskellige services særligt til anvendelse for IOT-enheder. Services kan anvendes til applets, så man på baggrund af en hændelse kan få udført en anden.

Til at detektere solopgang og solnedgang anvendes services fra 'Weather Underground' der giver signal ved henholdsvis solopgang og solnedgang. For solnedgang sættes lokationen til Dublin, for at sikre at alle hønsene er gået ind inden døren lukker. Indstillinger for 'sunset' ses i Figur 3.

Particle har også en service der giver mulighed for at kalde en funktion på f.eks. en argon. Indstillinger for funktionskald ses i Figur 4.



WU Sunset

This Trigger fires within 15 minutes of the sunset in your location.

Location

dublin

Call a function

This Action will call a function on one of your Devices, triggering an action in the physical world.

Then call (Function Name)

gate on "LGR01-ARG"

The name of the function you want to call; ex: for your lighting project you may name your function LED

with input (Function Input) (optional)

close

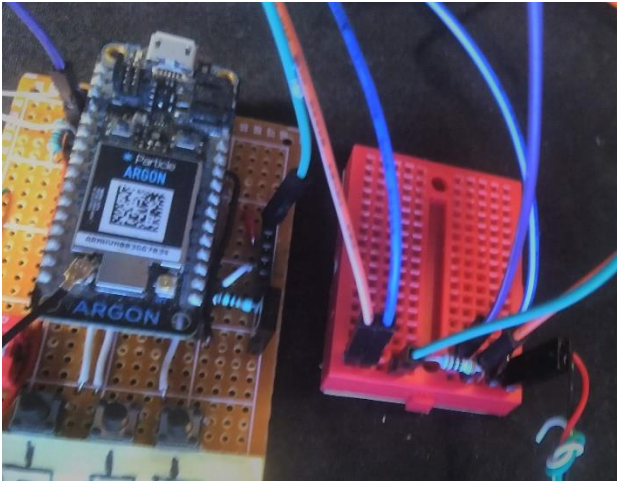
Whatever that function takes as an input, ex: color of LED, brightness of LED **Add ingredient**

Figur 3 – If -> sunset

Figur 4 – Then -> kald 'gate' med argumentet close

Energiforbrug

For at systemet potentielt skal kunne drives af et batteri, er det vigtigt at det har et lavt strømforbrug. Den enhed der har det største strømforbrug er servomotoren. Målinger foretaget med 1 ohms modstand i serie med 5V forsyning (Figur 5), viser at denne typisk et forbrug på op mod 550 mA.



Figur 5 – opstilling til måling af systemets samlede strømforbrug

Servomotoren vil fast have dette høje strømforbrug, idet den konstant vil forsøge at fastholde sin position. Den eneste måde at stoppe dette er at afbryde forsyningen. Der sættes derfor en transistor som forbindelse mellem servomotorens stelledning, og GND på boardet. Denne transistor aktiveres af en GPIO sat som digitalt output.

Der vælges en BD135 NPN transistor der bl.a. har følgende egenskaber:

- Den kan trække op til 1.5 A på collector benet.
- Den har en forstærkning på mindst 25.
- Base til Emitter spænding er op til 1 V

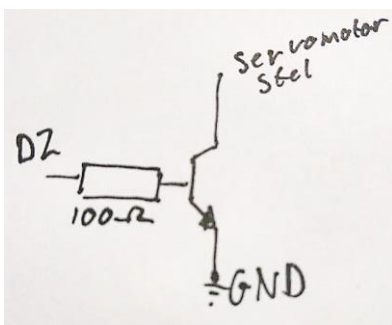
For at kunne trække en strøm på op til 550 mA med transistoren i mætning, skal base strømmen altså være mindst $\frac{550mA}{25} = 22 mA$.

Spændingsfald over basemodstanden vil mindst være $3.3V - 1V = 2.3 V$.

Dermed giver det en modstand på $\frac{2.3 V}{22mA} = 104.5 \Omega$.

Der anvendes en 100Ω modstand mellem D2 og basen på transistoren som det ses i Figur 6.

Se appendix [ap02] for datablad over transistor BD135.



Figur 6 – styring af stelforbindelse via transistor

Sleep

For at spare på strømmen undersøges det om enheden kan puttes i sleep, når den ikke har noget at foretage sig. Der er tre typer af sleep der skal vælges mellem.

De tre er 'Stop', Ultra Low Power' og Hibernate.

Kriterier for valg af sleep

- Der er ikke behov for ofte at vågne op, så forbrug ved opvågning har ikke den store betydning.
- Den største del af tiden har enheden ingen funktioner, så der ønskes en sleep tilstand med så lavt strømforbrug som muligt.
- Skal kunne styres af funktionskald gennem Wifi
- Skal kunne styres af gpio.
- Skal kunne styres af Real Time Clock(RTC).
- Der kan muligvis blive behov for at arbejde videre med variabler, så disse må ikke glemmes.

	Kan styres af GPIO	Kan styres af RTC	Kan styres af wifi	Forbrug ned til	Gemmer variabler
Stop	Ja	Ja	Ja	Ca. 400 μ A	Ja
Ultra Low Power	Ja	Ja	Ja	Ca. 80 μ A	Ja
Hibernate	Ja	Nej	Nej	Ca. 65 μ A	Nej

Tabel 2 – funktioner i forskellige typer af sleep [ref02]

Ud fra denne sammenligning virker det oplagt at vælge Ultra Low Power, der ligger langt under forbruget i 'stop', men som stadig har de funktioner der skal anvendes.

Ud fra tabellen på Figur 7 ses det dog at strømforbruget ikke nødvendigvis bliver meget lavere, når wifi skal være slået til som wakeup mulighed under sleep.

Parameter	Symbol	Min	Typ	Max	Unit
Operating Current (uC on, peripherals and radio disabled)	I_{idle}	3.1	3.52	3.58	mA
Operating Current (uC on, radio connected but idle)	$I_{wifi_cloud_idle}$	20.5	25.8	219	mA
Operating Current (uC on, radio connected and transmitting)	$I_{wifi_cloud_tx}$	20.1	31.7	261	mA
STOP mode sleep, GPIO wake-up	I_{stop_gpio}	350	396	459	uA
STOP mode sleep, analog wake-up	I_{stop_analog}	349	398	456	uA
STOP mode sleep, RTC wake-up	I_{stop_intrtc}	340	398	461	uA
STOP mode sleep, BLE wake-up, advertising	$I_{stop_ble_adv}$	340	442	3420	uA
STOP mode sleep, BLE wake-up, connected	$I_{stop_ble_conn}$	102	435	1970	uA
STOP mode sleep, serial wake-up	I_{stop_usart}	348	397	449	uA
STOP mode sleep, Wi-Fi wake-up	I_{stop_wifi}	15.3	22.2	110	mA
ULP mode sleep, GPIO wake-up	I_{ulp_gpio}		81.7	169	uA
ULP mode sleep, analog wake-up	I_{ulp_analog}		81.1	174	uA
ULP mode sleep, RTC wake-up	I_{ulp_intrtc}		80.7	168	uA
ULP mode sleep, BLE wake-up, advertising	$I_{ulp_ble_adv}$		141	3280	uA
ULP mode sleep, BLE wake-up, connected	$I_{ulp_ble_conn}$		138	1870	uA
ULP mode sleep, serial wake-up	I_{ulp_usart}	476	520	569	uA
ULP mode sleep, Wi-Fi wake-up	I_{ulp_wifi}	16.3	21.3	105	mA
HIBERNATE mode sleep, GPIO wake-up	I_{hib_gpio}		64.7	161	uA
HIBERNATE mode sleep, analog wake-up	I_{hib_analog}		65.0	159	uA
Power disabled (EN pin = LOW)	$I_{disable}$		20	30	uA

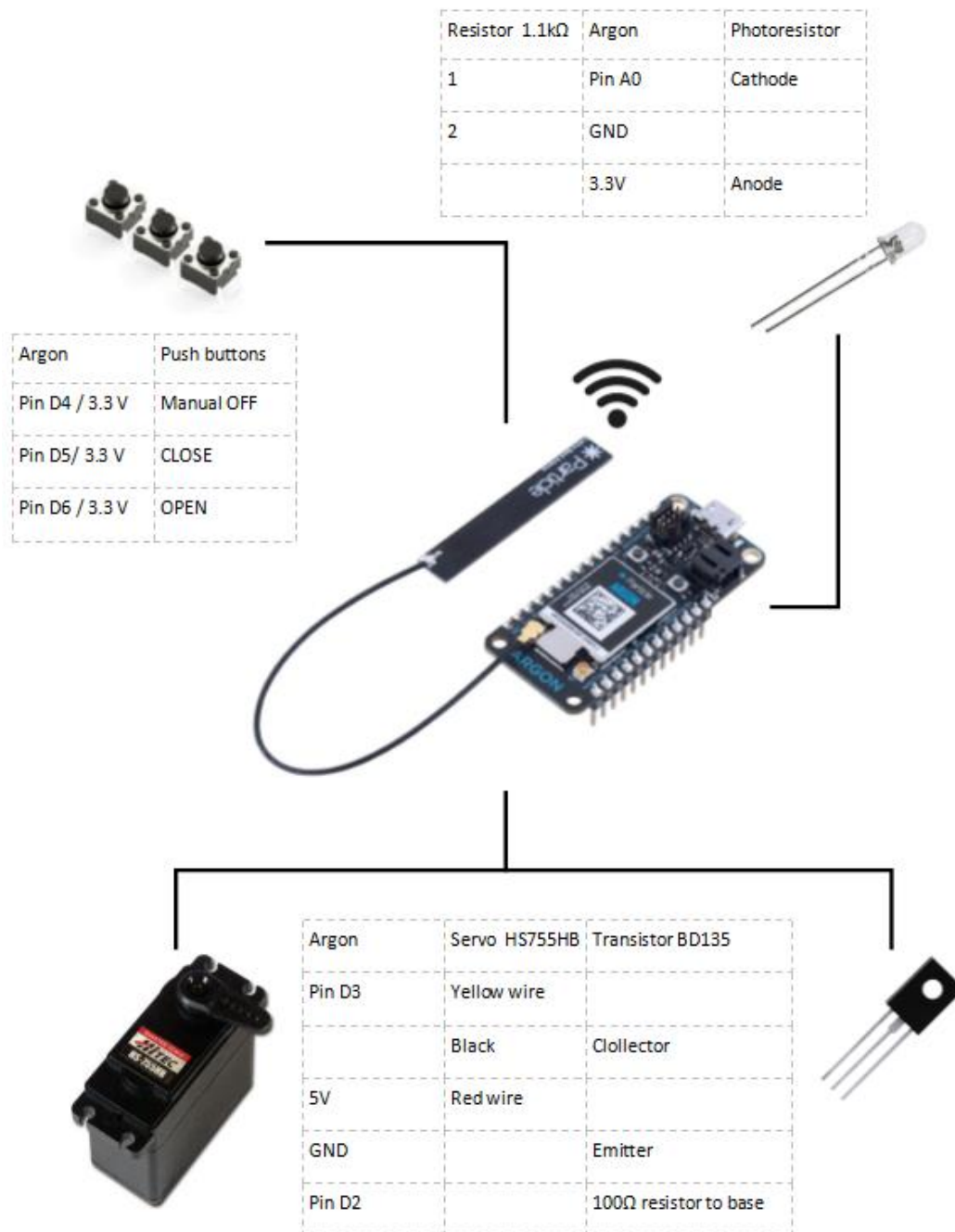
Figur 7 – tabel over strømforbrug fra datablad [ref03]

For at blive klogere på det reelle strømforbrug er 8 forskellige konfigurationer af sleep afprøvet. Resultatet af disse kan ses i bilag [ap01].

Konklusionen er at det ikke at der ikke kan findes en type af sleep som kan vækkes via de anvendte funktionskald fra IFTTT. Der er i øvrigt ikke den store strømbesparelse at hente når wifi fortsat skal være slået til. Der anvendes derfor ikke sleep i implementeringen.

Interface

Her ses det hvordan komponenterne i systemet skal forbindes.



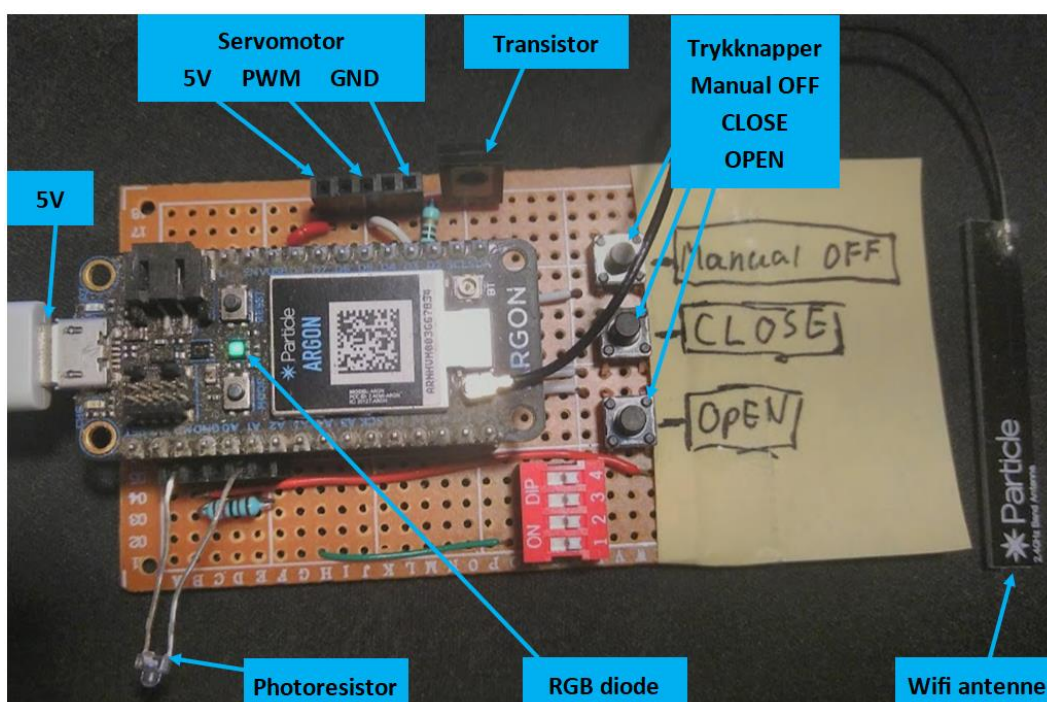
Figur 8 – Interface analyse

Implementering

I dette afsnit vises hvordan systemet er implementeret. Systemet er som udgangspunkt forsynet med 5 V gennem USB.

Hardwareopbygning

Kredsløbet er bygget op på et veroboard hvor forbindelser er loddet sammen på bagsiden. To rækker pinheaders er placeret, så man nemt kan montere en particle argon i midten. Phototransistoren er også monteret i pinheaders. Dette giver en mere fleksibel løsning så man f.eks. kan isætte ledninger, og føre dem ud til et andet sted hvor phototransistoren kan monteres.



Figur 9 – opbygning af kredsløb på veroboard

Veroboardet er placeret i et afskærmet rum inde i hønsehuset. Der er vinduer ud for rummet, så phototransistoren kan måle lysstyrken når den er monteret direkte på boardet. Opstillingen ses i Figur 10.



Figur 10 – afskærmet rum til enheden, dør med vinduer for lys

Servomotoren er forbundet via stænger til døren, og kan på den måde drive døren op eller i. Forbindelsen ses i Figur 11



Figur 11 – forbindelse mellem dør og servomotor



Figur 12 – servomotor og styreenheds placeing i forhold til hinanden

kode

Koden er skrevet i C++ og den kan findes på github [ref01].

Verifikation (LGR)

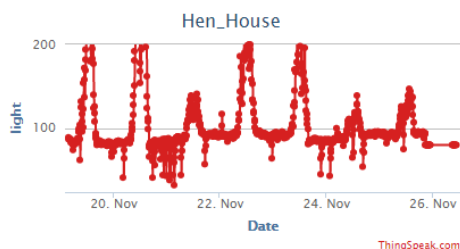
Under indkøring og test er particles online consol anvendt [ref04]. Herfra er det muligt at lave online funktionskald tilsvarende dem der kommer fra IFTTT. Desuden kan man overvåge events og udlæse de variabler der er gjort tilgængelige. Particles online consol giver et godt billede af hvad der sker lige nu. Der har dog også været behov for at kunne overvåge systemet over flere døgn i forbindelse af indstilling af lysniveau for aften, nat, morgen og dag. I den forbindelse har der også været anvendt thingsspeak.com. Via et webhook sendes et post til thingsspeak hver gang der bliver published en værdi for 'position' eller 'lightlevel' fra argon. Opsætning af et webhook kan ses i Figur 13. På thingsspeak er der oprettet to grafer der løbende plotter lysniveau og dørens position [ref06] [ref07], der ses to eksempler på dette i Figur 14 og Figur 14. Thingsspeak har givet et rigtig godt overblik til både tilpasning og test.

```

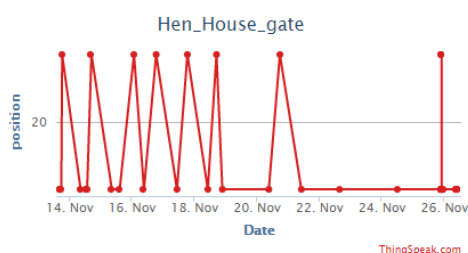
1 {
2   "event": "position",
3   "url": "https://api.thingsspeak.com/update",
4   "requestType": "POST",
5   "noDefaults": true,
6   "rejectUnauthorized": true,
7   "form": {
8     "api_key": "FTV0HGS9WJJPYWH1"
9   },
10  "query": {
11    "field1": "{{PARTICLE_EVENT_VALUE}}"
12  }
13 }

```

Figur 13 - webhook for position



Figur 14 – graf fra thingsspeak der viser lysniveauet over flere døgn



Figur 15 – eksempel på at døren har fungeret i en periode, men at noget derefter er gået galt

Test

Der er to tests af systemet. Den ene er en grundig test ved 'skrivebord' hvor specifikke handlinger og kriterie er beskrevet. Den anden er en field test hvor man installerer systemet, og over flere dage overvåger om døren åbner og lukker som ønsket. Opdelingen laves fordi det er tidskrævende at udføre reelle test af systemet da der kun er to hændelser om dagen, og et fejlscenarie kræver 20 timers ventetid.

skrivebordstest

Denne test udføres som en samlet test af systemet. Her er testen beskrevet punktvis. Testen udføres ved 'skrivebord' og hastigheden af fejlhåndtering sættes op.

Betingelser skal være opfyldt for at testen er bestået.

Forklaring: positiv pwm puls på ca. 14% betyder at døren lukkes, og ca. 2% betyder at døren åbnes.

Opsætning:

- koden hentes fra github [ref01].
- Koden ændres, så konstanten 'MAXWAIT' sættes til 60000. Dette vil sige at systemet allerede går i sensor control efter 1 minut.
- Et scope med mindst to kanaler tilsluttes. En kanal til 5V forsyning og stel til servomotoren. Den anden kanal til pulssignalet til servomotoren og GND på boardet.
- Hvis ikke systemet er tilsluttet servomotoren er det nødvendigt at sætte en 100k Ω modstand ind mellem 5V og GND fra servomotoren. Dette gøres for at trække spændingen på stelledningen op til 5 V, hvis ikke der er forbindelse til stel.

Test:

Testen skal følges kronologisk for at give det rigtige resultat

Handling	Kriterie
Systemet tilsluttes, og flashes med den ændrede kode.	Efter at enheden er genstartet skal RGB dioden lyse svagt grøn.
Vent ét minut	Efter et minut skifter RGB dioden til rød
Forsøg ny at skærme for lyset til phototransistoren ved at holde fingrene omkring den.	Indenfor 10 sekunder skal pwm signalet gå mod en puls på ca. 2%.
Lys i stedet på phototransistoren.	Indenfor 10 sekunder skal pwm signalets puls gå mod ca. 14%.
Vent yderligere 10 sekunder.	Se at spændingen til servomotoren er faldet til 0V.
Fortsæt med at holde phototransistoren oplyst. Via particles online consol sendes 'open' til funktionen 'gate'	Pwm signalet skal gå mod en puls på ca. 2%, og efter 10 sekunder skal forsyningen til servomotoren være slukket. RGB dioden skal være blevet grøn igen.
Efter 30 sekunder sendes 'close' igen online, og efter yderligere 30 sekunder sendes 'open' igen. Dette getages én gang.	I de to minutter skal RGB dioden forblive grøn.
Trykknappen 'Close' aktiveres.	Det kontrolleres at pwm falder til ca. 2%, og at RGB dioden bliver blå, efter 10 sekunder skal forsyningen til servomotoren være faldet til 0V.
Trykknappen 'Open' aktiveres.	Det kontrolleres at pwm stiger til ca. 14%, og at RGB dioden bliver blå, efter 10 sekunder skal forsyningen til servomotoren være faldet til 0V.
Trykknappen 'manual off' aktiveres.	RGB dioden bliver igen grøn

Vurdering

Til testen er der anvendt funktionskald fra particle consol i stedet for IFTTT, da disse kun forekommer to gange om dagen. Det betyder at disse kald ikke er omfattet af testen. Det er derfor ikke påvist at hændelserne forekommer ved solopgang og solnedgang ved denne test.

Det der verificeres, er;

- At systemet kan modtage online systemkald når ikke den er i 'manual mode'.
- At systemet kan måle og styres efter et lysniveau hvis ikke det for online systemkald.
- At systemet kan styres manuelt.

Field test

Systemet installeres i hønsehuset. Det kontrolleres at man kan åbne og lukke døren ved hjælp af trykknapperne 'open' og 'close'.

Vurdering

Sidst på aftenen kontrolleres det at døren er lukket, og om morgenen efter solopgang kontrolleres det at døren er åben. RGB dioden skal under testen lyse grønt. Ved hjælp af thingsspeak kan man se præcise tidspunkter for hvornår døren åbnes og lukkes. Man kan også sikre at systemet ikke går i 'sensor mode' ved at checke at der ikke kommer målinger af lysniveau.

Det der verificeres, er;

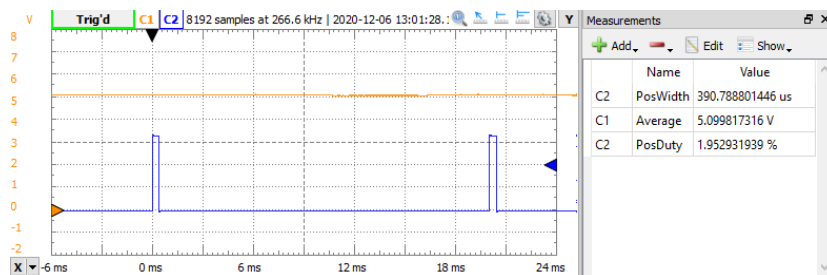
- At lågen kan åbnes og lukkes af systemet
- At handlingerne bliver udført på baggrund af online systemkald ved solopgang og solnedgang.

Resultater

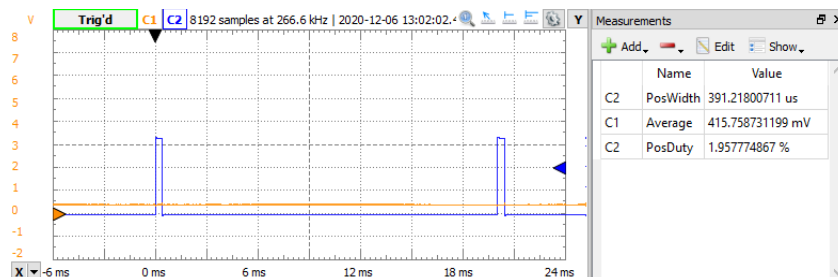
Har vurderes resultaterne af de to udførte tests.

Resultat af skrivebordstest

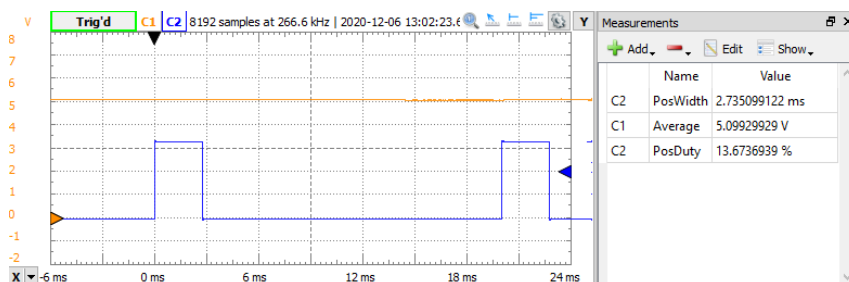
Første test er udført ved 'skrivebord' med scope på Analog Discovery. Testen er gået godt, og alle betingelser er opfyldt. Uddrag fra test ses i Figur 16, Figur 17, Figur 18 og Figur 19.



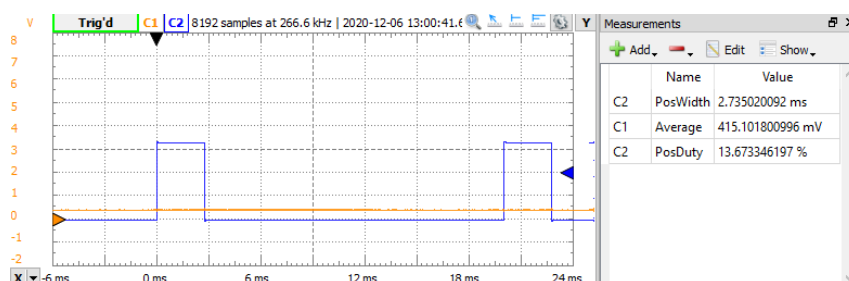
Figur 16 – efter online funktionskald 'open' bliver PosDuty kortere



Figur 17 – Efter fem sekunder er forsyningsspændingen faldet



Figur 18 – efter online funktionskald 'close' bliver PosDuty længere



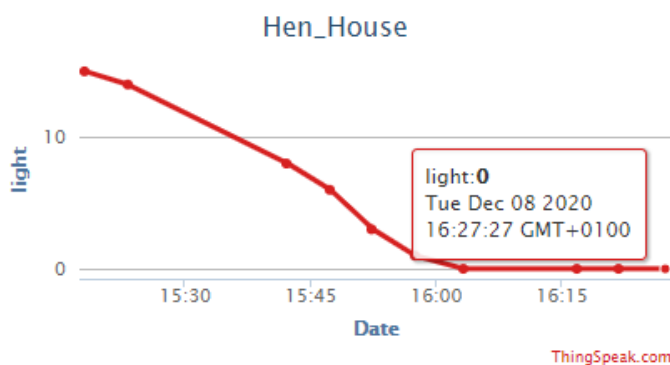
Figur 19 – Efter fem sekunder er forsyningsspændingen faldet

Resultat af field test

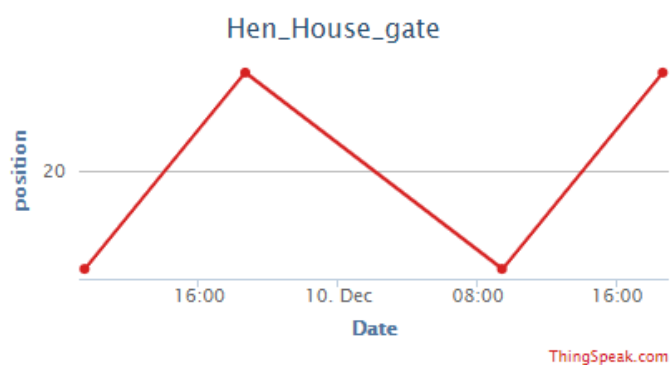
Testen er afviklet fra d. 8. december klokken 16:27 (før solnedgang) til d. 10. december efter solnedgang. Det er observeret at lågen har været åben og lukket som ønsket. Den grønne diode har været tændt under hele testen.

På thingspeak er det kontrolleret at de seneste målinger har været inden testens start som det ses i Figur 20. Der har altså ikke været målinger under testen, hvilket betyder at systemet ikke har været i 'sensor mode'.

Det kontrolleres også at tidspunkterne for at åbne og lukke lågen er passende i forhold til solopgang og solnedgang. Af grafen vist i Figur 21 ses det at lågen er gået til position 5 (åben) hver morgen klokken 9:25, og til position 35 (lukket) hver aften klokken 18.38.



Figur 20 - sidste lysmåling d. 8. december klokken 16.27 da systemet sættes op med rigtig konfiguration



Figur 21 - position 5 = lågen er blevet åbnet, position 35 = lågen er blevet lukket

Opsamling af resultater

Ud fra resultaterne vurderes det hvilke krav der er verificeret ved hver af de to test. Et overblik ses i Tabel 3.

Krav	Verificeret ved test		Note
	skrivebord	Field	
Systemet skal kontrollere en servomotor der kan åbne og lukke døren i et hønsehus.		Ja	
Systemet skal have informationer om solopgang og solnedgang via internettet.		Ja	
Når solen er stået op, skal døren åbne.		Ja	
Når solen er gået ned, skal døren lukke.		Ja	
Mens systemet ikke har nogen funktionalitet, skal systemet bruge så lidt strøm som muligt.			Det burde være muligt at optimere strømforbruget yderligere, så dette krav er ikke løst
Hvis der ikke er internetforbindelse, skal systemet styres ud fra lysmålinger målt med en phototransistor.	Ja		
Systemet kan indeholde styring af belysning i hønsehuset			Ikke implementeret
Systemet kan være drevet af et batteri der om dagen oplades af en solcelle			Ikke implementeret

Tabel 3 - vurdering af resultater i forhold til krav

Konklusion

Der er gennem projektet fundet en løsning der løser de krav der var stillet til opgaven.

Der er til opgaven anvendt en Particle argon, som har vist sig at være meget fleksibel at arbejde med til cloudløsninger. Der er gennem projektet anvendt online services til at udføre online funktionskald, logge data, og overvåge variabler og events. De anvendte services har i høj grad været forberedt til enheder fra Particle. Jeg ville bestemt overveje Particle argon til min næste internetforbundne 'dime'.

Der er arbejdet med optimering af strømforbrug. Bortset fra at en høj standbystrøm til servomotoren er fjernet, så har det været svært at optimere på strømforbruget. Så længe enheden skal være forbundet til internettet via wifi, vil der være et forbrug på mindst 30mA – 40mA. Dette kan være en udfordring hvis man ønsker at lave en løsning der er forsynet fra et batteri. Hvis man i perioder kan slukke for wifi kan man det mindske strømforbruget flere hundrede gange.

Fremtidigt arbejde

Som løsning på det høje strømforbrug med aktiv wifi under standby, kunne man overveje at lægge initiativet til internetkommunikation hos enheden selv. På den måde kan wifi slukkes, når enheden ikke har behov for kommunikere.

Dette kunne gøres på to måder. En metode kunne være at anvende en API service som kan kaldes fra enheden. Dette kunne være en API der returnerer et JSON objekt der bl.a. indeholder oplysninger om solopgang og solnedgang. Et eksempel på en egnet API kan findes ved [ref09].

En anden metode er at få en service til, ved solopgang og solnedgang, at publishe til en MQTT-broker. For hver time eller lignende kunne enheden så vågne og subscribe hos MQTT-broderen, og derefter udføre påkrævede handlinger hvis der er noget til den.

Desuden kan der arbejdes videre med implementering af batteri og opladning ved hjælp af solcelle, samt styring af lys i hønsehuset.

Referenceliste for samlet dokumentation

Id	Beskri- velse	Reference
[ref01]	Github til projektet	https://github.com/spejderlasse/IOT
[ref02]	Particle reference manual	https://docs.particle.io/reference/device-os/firmware/argon/
[ref03]	Particle datablad	https://docs.particle.io/datasheets/wi-fi/argon-datasheet/
[ref04]	Particle online consol	https://console.particle.io/devices
[ref05]	If This Then That	https://ifttt.com
[ref06]	Things Speak, gate status	https://thingspeak.com/channels/1194001
[ref07]	Things Speak, lightlevel	https://thingspeak.com/channels/1192992
[ref08]	Eksempel med phototransistor	https://docs.particle.io/tutorials/hardware-projects/hardware-examples/argon/
[ref09]	Eksempel på api	http://api.weatherapi.com/v1/forecast.json?key=20f7c5e9b4b64ad19c174250201611&q=copenhagen&unixdt

Appendix

Appendix findes på github [ref01]

Id	Filnavn	Beskrivelse
[ap01]	ap01 Målinger af strømforbrug.pdf	Dokumentation for målinger af strømforbrug
[ap02]	ap02 BD135.pdf	Datablad for transistor BD135
[ap03]	ap03 Datablad_HS755HB.pdf	Datablad på servomotor HS755HB
[ap04]	ap04 argon-pinout-v1.0.pdf	Oversigt over pins på argon board
[ap05]	ap05 bpw17n.pdf	Datablad på phototransistor