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

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

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.

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

Indhold

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

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
hønsehus	Hønsehuset består af selvehønsehuset samt en tilhørende lille hønsegård. Det er hele denne enhed der lukkes og åbnes for
Døren	Den dør der gåt ind til den lille hønsegård der er i forbindelse med hønsehuset
Argon	Udviklingsboard fra Particle
IFTTT	Online service 'If This Then That' [ref05]
Thingsspeak	Online service til at logge data

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 sikrer 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.

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 photoresistor.

Valgfri

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

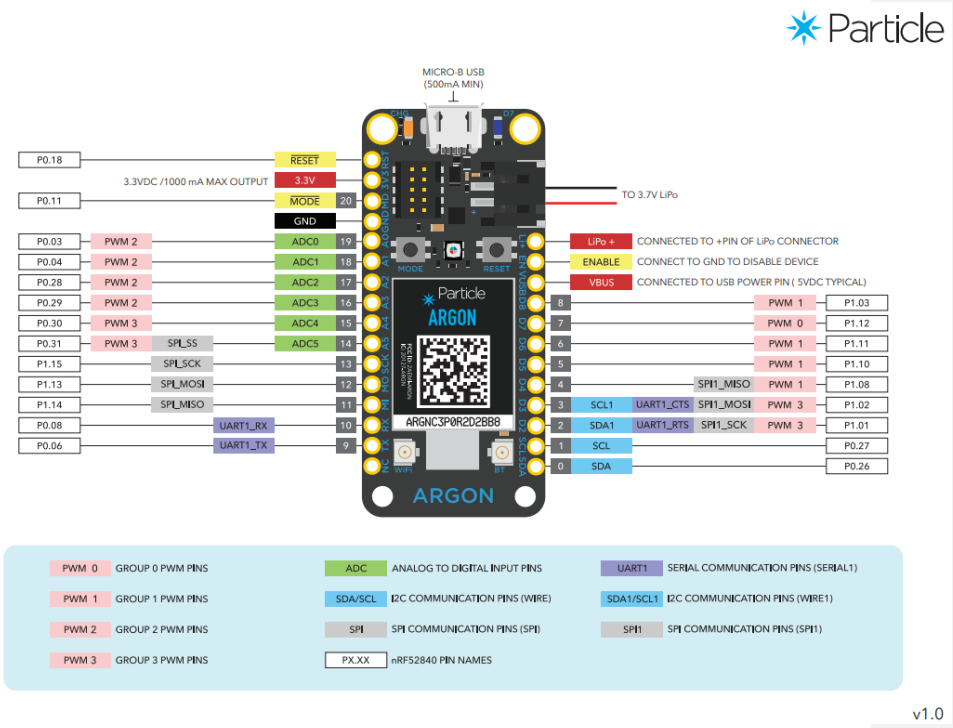
Platform

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

- Minimum én ADC til photoresistor.
- 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 frobindelse 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.



Figur 1 - pinout for Paricle 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 = neural
- 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.

Kommenterede [LGR1]: Der kunne godt være et afsnit om photoresistoren

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 fungere som fejlhåndtering. Systemet måler via en photoresistor 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 trykknap for 'manual mode off' aktiveres.

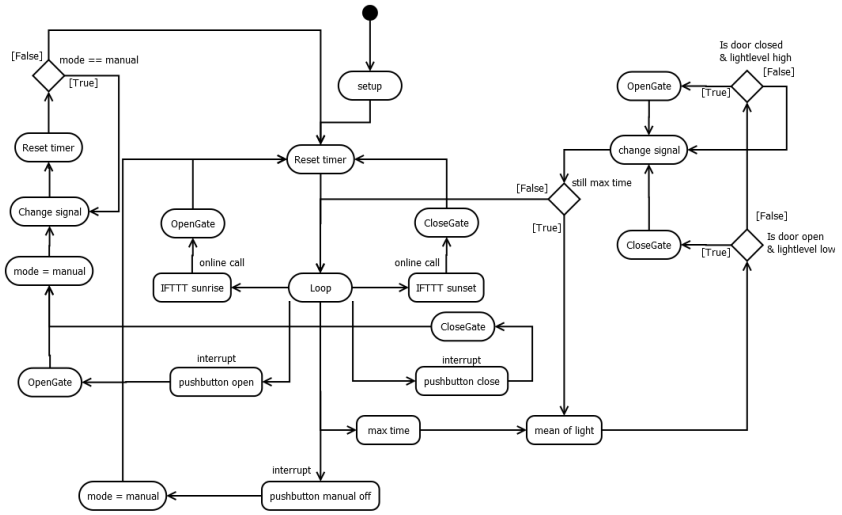
Feedbacksignaler

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. Aktivitetsdiagrammet består af en blanding af almindelige processer, og interrupt styrede processer.



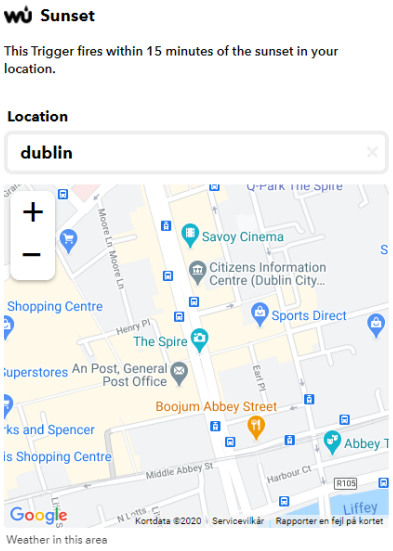
Figur 2 - aktivitetsdiagram

Styring efter solopgang og solnedgang

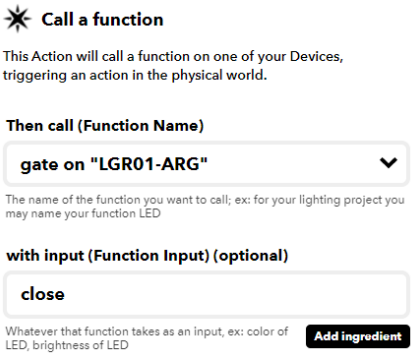
For at styre døren efter solopgang og solnedgang vælges det at anvende to services fra IFTTT.com. IFTTT er en online tjeneste der kombinerer mange forskellige services særligt til anvendelse for IOT-enheder. Services kan anvendes til applets, så man på baggrund af en hændelse kan få udført en anden.

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.



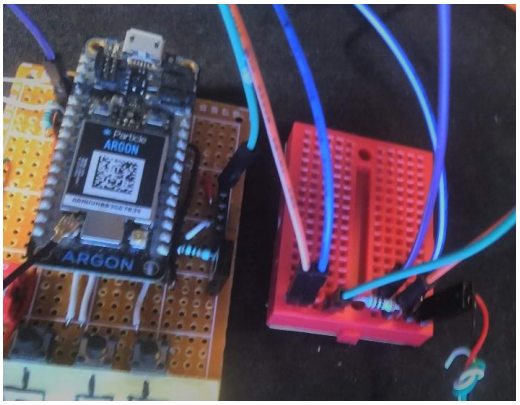
Figur 3 - If -> sunset



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

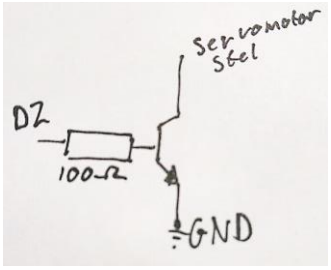
For at kunne trække en strøm på op til 550 mA med transistoren i mætning, skal base strømmen altså være mindst $\frac{550mA}{25} = 22\ mA$.

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 µA	Ja
Ultra Low Power	Ja	Ja	Ja	Ca. 80 µA	Ja
Hibernate	Ja	Nej	Nej	Ca. 65 µA	Nej

Tabel 1 – funktioner i forskellige typer~~modes~~ 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

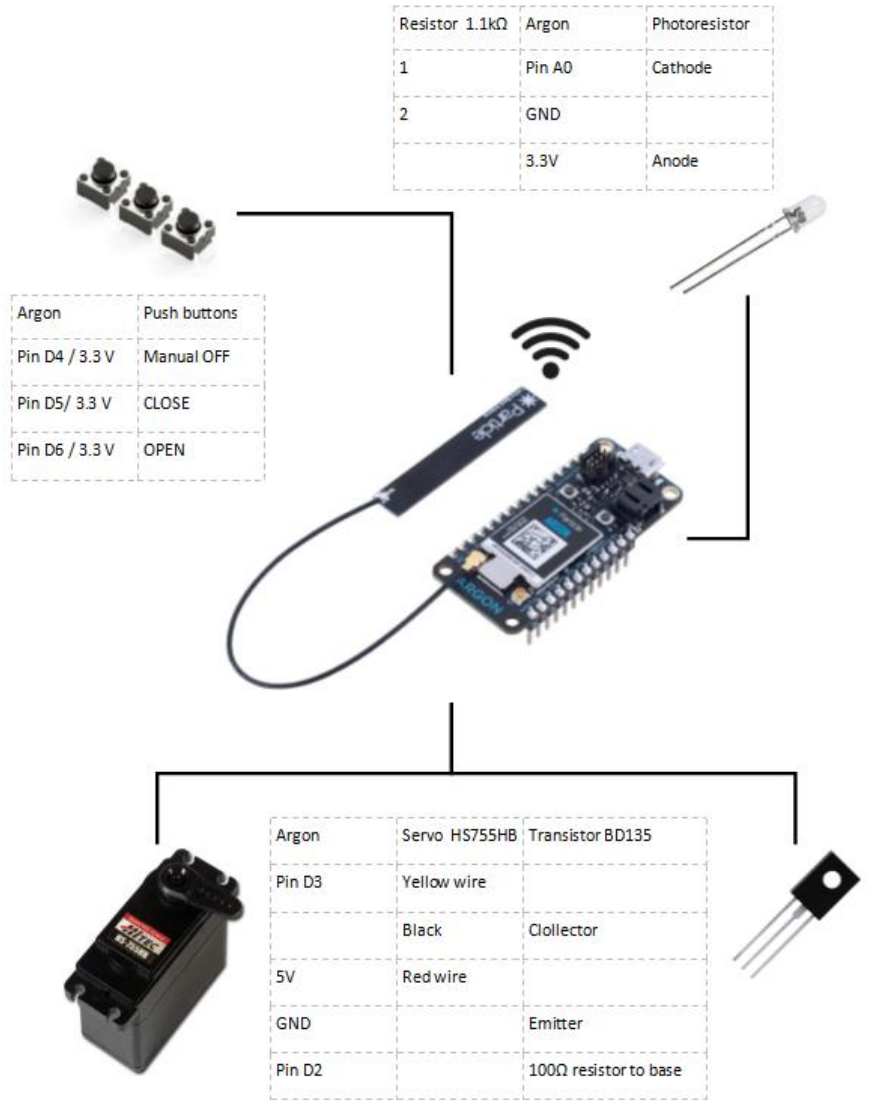
Figur 7 - tabel over strømforbrug fra datablad [ref03]

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.



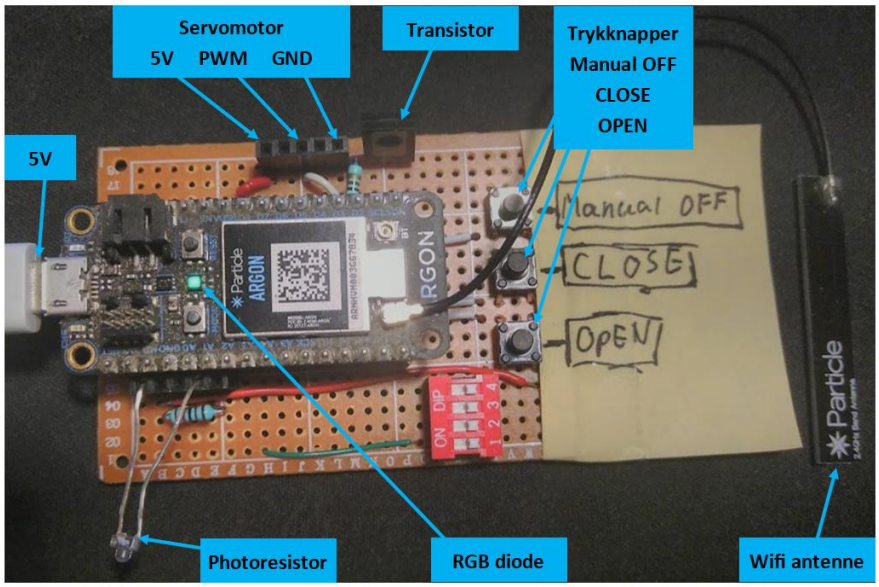
Figur 8 - Interface analyse

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. Photoresistoren 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 photoresistoren 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 photoresistoren 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

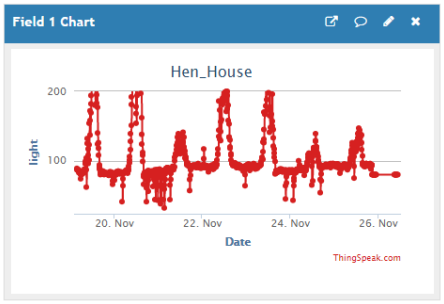
Figur 12 - servomotor og styreenheds placeing i forhold til hinanden

kode

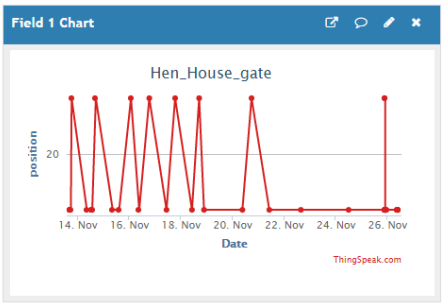
Koden er skrevet i C++ og den kan findes på github [ref01].

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. Her er der oprettet to grafer der løbende plotter lysniveau og dørens position [ref06] [ref07], der ses to eksempler på dette i Figur 13 og Figur 13. Thingsspeak har givet et rigtig godt overblik til både tilpasning og test.



Figur 13 - graf fra thingsspeak der viser lysniveauet over flere døgn



Figur 14 - eksempel på at døren har fungeret i en periode, men at noget derefter er gået galt

Test

Der er udført en samlet test af systemet. Her er testen beskrevet punktvist. Testen udføres ved 'skrivebord' og hastigheden i koden sættes op, da det er tidskrævende at udføre reelle test fordi der kun er to hændelser om dagen.

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

- 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 kan det være nødvendigt at sætte en 100kΩ modstand ind mellem 5V og GND til servomotoren.
- 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 photoresistoren ved at holde fingrene omkring den.

*indenfor 10 sekunder skal pwm signalet gå mod en puls på ca. 2%.

- Lys i stedet på photoresistoren.

*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 photoresistoren 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.

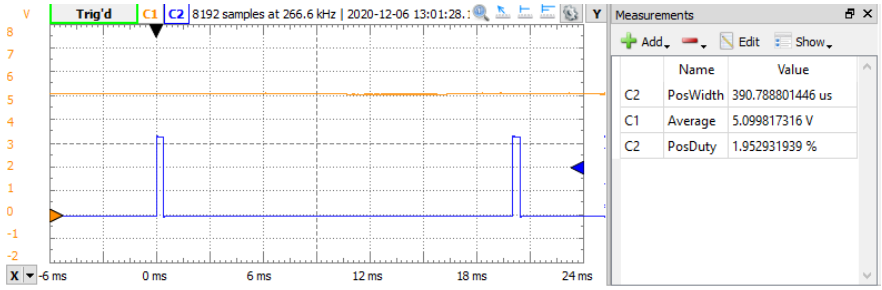
*RGB dioden bliver igen grøn

Vurdering af test:

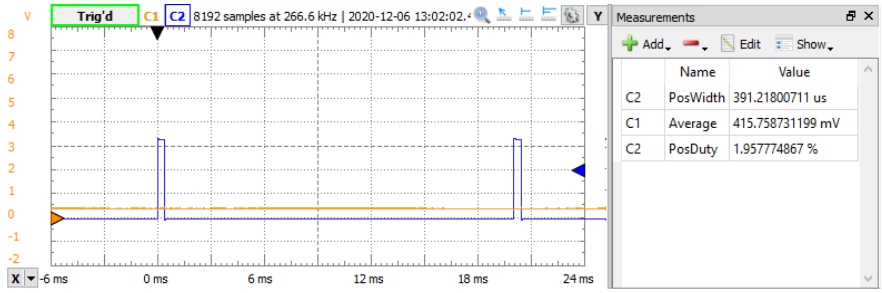
Hvis disse betingelser markeret med * er opfyldt, er testen godkendt

Resultat

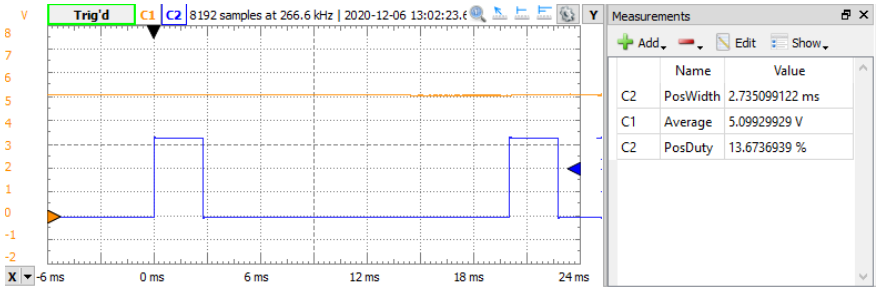
Testen som er udført ved 'skrivebord' med scope på Analog Discovery. Testen er gået godt, og alle betingelser er opfyldt. Uddrag fra test ses i Figur 15, Figur 16, Figur 17 og Figur 18.



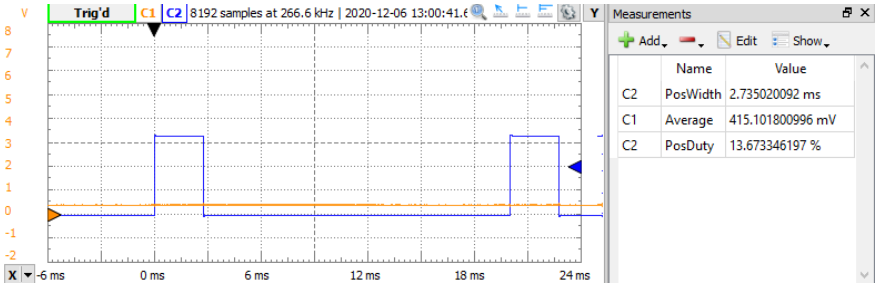
Figur 15 - efter online funktionskald 'open' bliver PosDuty kortere



Figur 16 - Efter fem sekunder er forsyningsspændingen faldet



Figur 17 - efter online funktionskald 'close' bliver PosDuty længere



Figur 18 - Efter fem sekunder er forsyningsspændingen faldet

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 'dims'.

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.

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 ikke enheden vil kommunikere. Dette kunne gøres hvis man i stedet for et online funktionskald, anvendte et API kald fra enheden selv.

Referenceliste for samlet dokumentation

Id	Beskrivelse	Reference
[ref01]	Github til projektet	https://github.com/spejderlasse/IOT
[ref02]	Paticle 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

Appindix findes på github

[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