



IoT Plant Irrigation

Gruppemedlemmer	Studie nr.
Mevlüt Kahraman	201810387
Tommy Kjær Jensen	201806167

Underviser
Morten Opprud Jakobsen
Klaus Kolle

Aarhus Universitet, School of Engineering

Elektronik, Herning

21.11.2020

Indhold

1	Introduktion.....	3
1.1	Projektbeskrivelse	3
2	Krav analyse.....	3
3	Systemdesign.....	4
3.1	Valg af komponenter	5
3.2	Interface analyse	6
3.2.1	Interface: Fugtsensor til Particle Argon.....	6
3.2.2	Interface: Particle Argon til Vandingsstyring	10
3.2.3	Interface: Particle Argon til/fra Vejrtjeneste	13
4	Power Profiling	15
5	Implementering	16
5.1	Software	16
5.2	Hardware	18
6	Test / Verifikation	21
6.1.1	Test 1: Tør jord	22
6.1.2	Test 2: Let fugtig jord (lessWet).....	22
6.1.3	Resultat af Test / verifikation	22
7	Konklusion og perspektivering	23
8	Referencer	24

1 Introduktion

Der skal laves et IOT-projekt, hvor der tages udgangspunkt i en Particle Argon. Argonen skal ved hjælp af data fra internettet, udføre en opgave, hvor der også gøres brug af en sensor og en aktuator, bedre end en tilsvarende løsning uden adgang til webdata.

1.1 Projektbeskrivelse

Sensor: Fugtighedsmåler

Aktuator: Magnetventil, relay

Webdata: Lokal vejrudsigt

Projektet er en avanceret plantevander, der tager højde for det kommende vejr. Ud fra vejrudsigten skal plantevanderens beslutte om der skal vandes eller om det kan vente. Hvis der er regn i sigte vil den ikke vande, da det vil være spildte ressourcer, derimod hvis der stadig er for tørt, efter det har regnet, og der ikke er udsigt til mere regn, vil den vande i det omfang der er nødvendigt, bestemt af inputtet fra sensoren.

2 Krav analyse

I denne afsnit ses de overordnede krav til systemet. Kravene beskriver hvad systemet skal kunne for at opfylde behovet. I denne analyse er der brugt *EARS Requirement Analysis* [8], som deler kravene op i 6 forskellige tilstande, såsom er *altid gældende*, *event drevet*, *tilstands-drevet*, *debugging-features*, *optionel*, *uønsket opførsel*.

Altid gældende

- Der skal foretages en fugtighedsmåling.
- Lokal vejrudsigt skal tjekkes igennem Wi-fi internettet.

Event drevet

- Hvis fugtighedsniveauet er for lavt, kontrolleres vejrudsigten.
- Hvis fugtighedsniveauet er for lavt, og der er udsigt til regn indenfor 2 timer, afventes regn.
- Hvis fugtighedsniveauet er for lavt, og der ikke er udsigt til regn indenfor 2 timer, vandes der.
- Hvis det ønskede fugtighedsniveau opnås, indstilles vanding.

Tilstands-drevet

- Når ventilen er i aktiv tilstand, skal der opsamles fugtighedsdata

Optionel

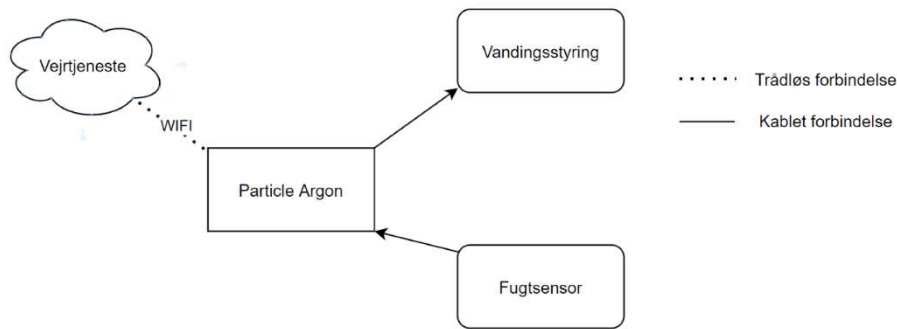
- Kontrol af dagtemperatur, i forhold til ikke at vande mens solen står på, og vandet blot vil fordampe (koge planten).
- Kontrol af nattemperatur, i forhold til ikke at vande mens der er frostvejr, og vandet blot vil lave plantens rødder til en isterning.

Uønsket opførsel

- Hvis net-forbindelsen forsvinder fra enheden, skal den prøve at få kontakt igen.

3 Systemdesign

Det overordnede blokdiagram, der også viser grænsefladerne, samt data flow.



Figur 1 Blokdiagram over systemet

Pilene angiver signalernes flow, dvs. fugtsensoren modtager ikke nogen data fra Argon, men sender kun data, over en kablet forbindelse.

Vandingsstyringen styres af Argon via en kablet forbindelse. Vandingsstyringen giver ikke noget feedback, altså er dette også en envejskommunikations.

Vejrtjenestedataene bestilles via en webhook, hvor der sendes en forespørgsel, der hvis tjenesten og wifi er oppe og køre, medfører at der kommer et svar indeholdende de vejrdato der er bedt om. Forbindelsen her er trådløs.

Der skal træffes en beslutning om hvornår og hvor meget der skal vandes, det gøres ud fra beslutningsmatrixen herunder.

Tabel 1 Beslutningsmatrix

Fugtmåling:	Tør		LessWet		MoreWet		Wet	
Vejrudsigt:	Regn	Tørt	Regn	Tørt	Regn	Tørt	Regn	Tørt
Vanding til thresholds	lessWet	lessWet	No	moreWet	No	No	No	No

Grænseværdierne er:

lessWet > 2,6 V eller 3250

moreWet > 1,62V eller 2025

wet > 1,3 V eller 1625

De er lavet ud fra et estimat, og der er som sådan ikke noget videnskabeligt grundlag for det.

3.1 Valg af komponenter

I denne sektion har man opstillet udvalgte komponenter, hvor der er beskrevet komponenter og deres funktioner. Der skal i produktet anvendes en fugtsensor til at give mulighed for at måle fugtighedsniveau direkte fra jorden omkring planten. I øvrigt spiller fugtsensoren en central rolle i projektet, da vandstyringsinterface skal kontrolleres ud fra de værdier, der er modtaget fra jorden igennem denne sensor. Desuden skal der anvendes et relæ, til at agere styring for vandforsyningen. I den forbindelse skal der også anvendes en aktuator, der tænder og slukker for vandet. Som kontrolplatform anvendes der IoT enhed Particle Argon. Komponenterne er valgt primært fra dem man har til rådighed derhjemme fra de tidligere semestre og hvad der ellers er fundet interessant og billigt.

Tabel 2 Oversigt over de udvalgte komponenter

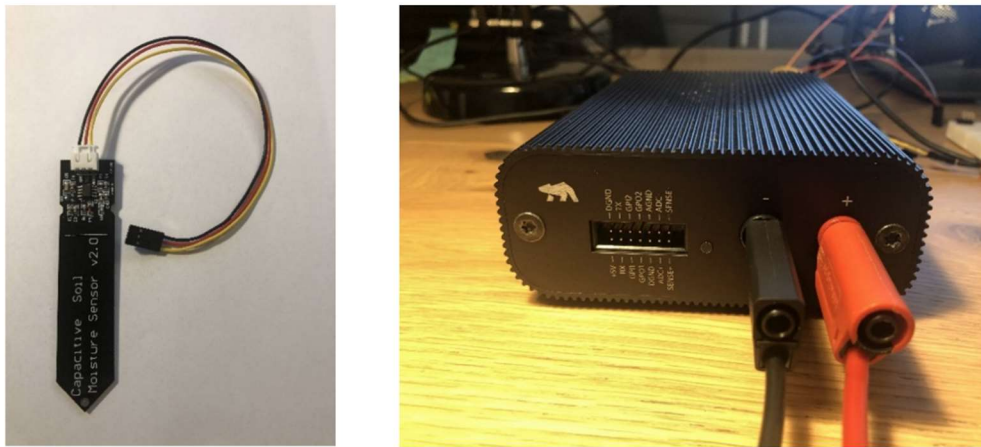
Valgt komponent	Model nr.	Funktionen af komponent	Forhandler
Fugtsensor	HW-390 Capasitive Moisture Sensor v2	Målingen af fugtigheden i jorden	minielektro.dk
Relæ	HLS8L-DC3V-S-C	Tænde og slukke forbindelsen til magnetventil	www.seeedstudio.com
Water pump	Submersible Water Pump 240L/H	Pumpe vandet fra vandbeholderen til planten	arduinotech.dk
IoT Device	Particle Argon	Mikrocontroller (Technical Platform)	let-elektronik.dk

3.2 Interface analyse

3.2.1 Interface: Fugtsensor til Particle Argon

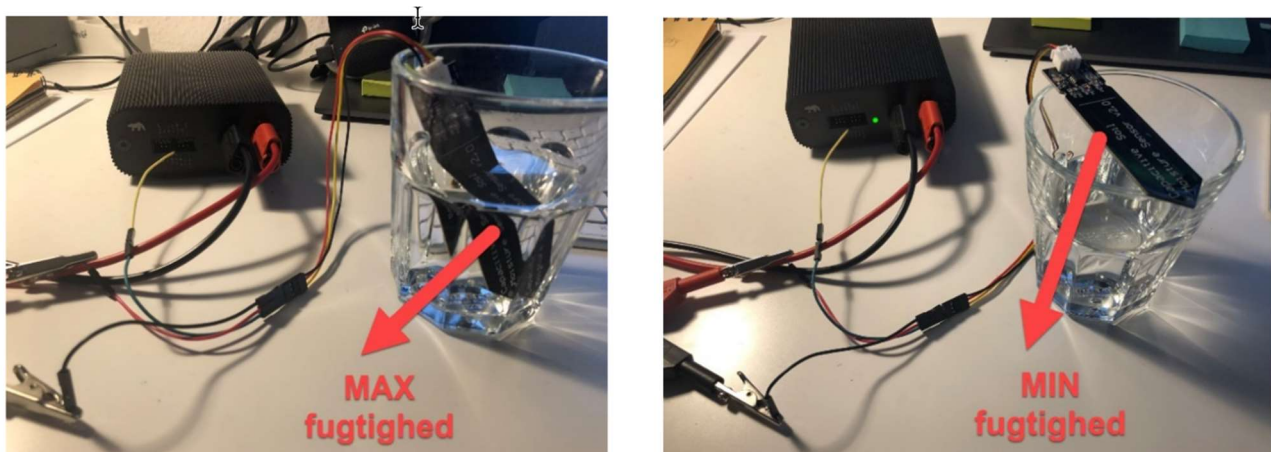
Fugtsensoren skal have en forsyning fra Argon, da den med fordel kan have mellem 3,3V-5,5V forsyning. Hovedsageligt skal alle målinger foretages fra jorden omkring planten med fugtsensoren, og der skal være interaktion mellem Argon og fugtsensoren jævnligt.

Herunder ses der forskellige test af fugtsensoren i forbindelse med power profiling og fugtniveauet. Der er anvendt en power profiler, som er produceret af *Otii* [15]. Formålet er at kunne observere og beregne hvor stort strømforbruget er, når fugtsensoren er i aktiv tilstand.



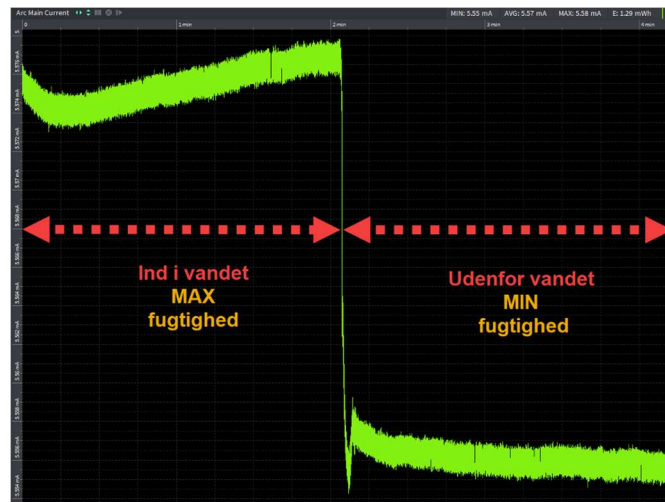
Figur 2 Fugtsensor og Otii Power Profiler

Herunder ses testopstillingen af fugtsensoren og Otii, til at foretage strømforbrugsmålinger. For at gøre det lidt praktisk, er der brugt et glas vand til at måle højeste og laveste fugtniveauet under aktiv tilstand. Fugtsensoren er forsynet af 3.3V fra Otii under testen.



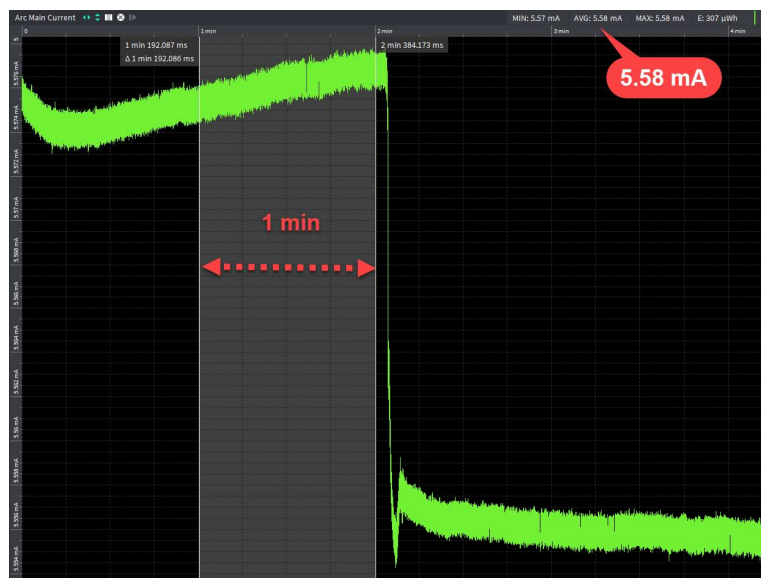
Figur 3 Test opstillingen med Power Profiler og fugtsensor

Herunder ses der et skærbillede, som er taget under målingen fra Otii [15] power profileren, hvor der ses de to forskellige tilstande, som fugtsensoren er testet i.

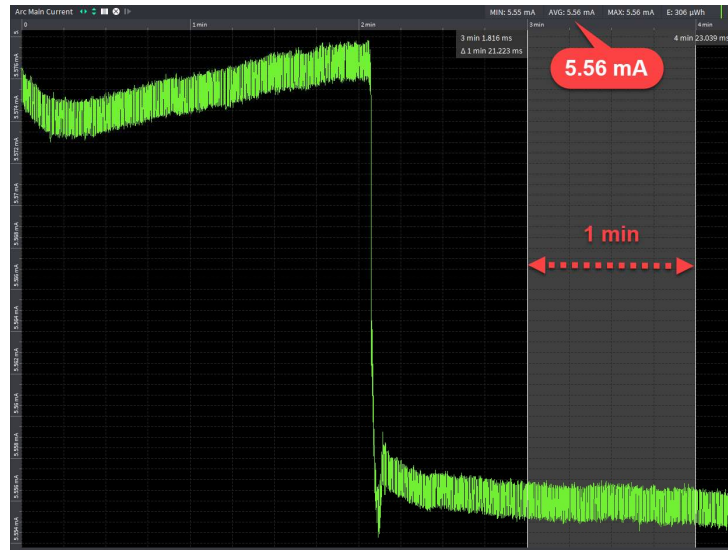


Figur 4 Skærbillede af power profiling af fugtsensoren

I løbet af et minut er der brugt 5.58 mA strøm, når sensoren ligger i vandet. Til gengæld er der brugt 5.56 mA, når sensoren er taget op af vandet. Der er næsten ikke forskel mellem de to tilstande mht. strømforbrug.



Figur 5 Strømforbrug i gennemsnit i 1 min - fugtsensoren ind i vandet



Figur 6 Strømforbrug i gennemsnit i 1 min - fugtsensoren udenfor vandet

Derefter er der foretaget en test med *Particle Argon*, for at aflæse spændingsværdier i *Teraterm*. I den forbindelse er der skrevet et testprogram, der konverterer målingerne til Volt, og printer dem ud på den serielle forbindelse, for at kunne se fugtsensorens reaktioner overfor fugten.

```
int MoistSense = A0; //Moistensor / ADC 0
float analogvalue; // Here we are declaring the integer variable analogvalue,
float bitres = 0.0008; // 100 microVolt
int bits = 4096; // The number of bits
float val = 0;
```

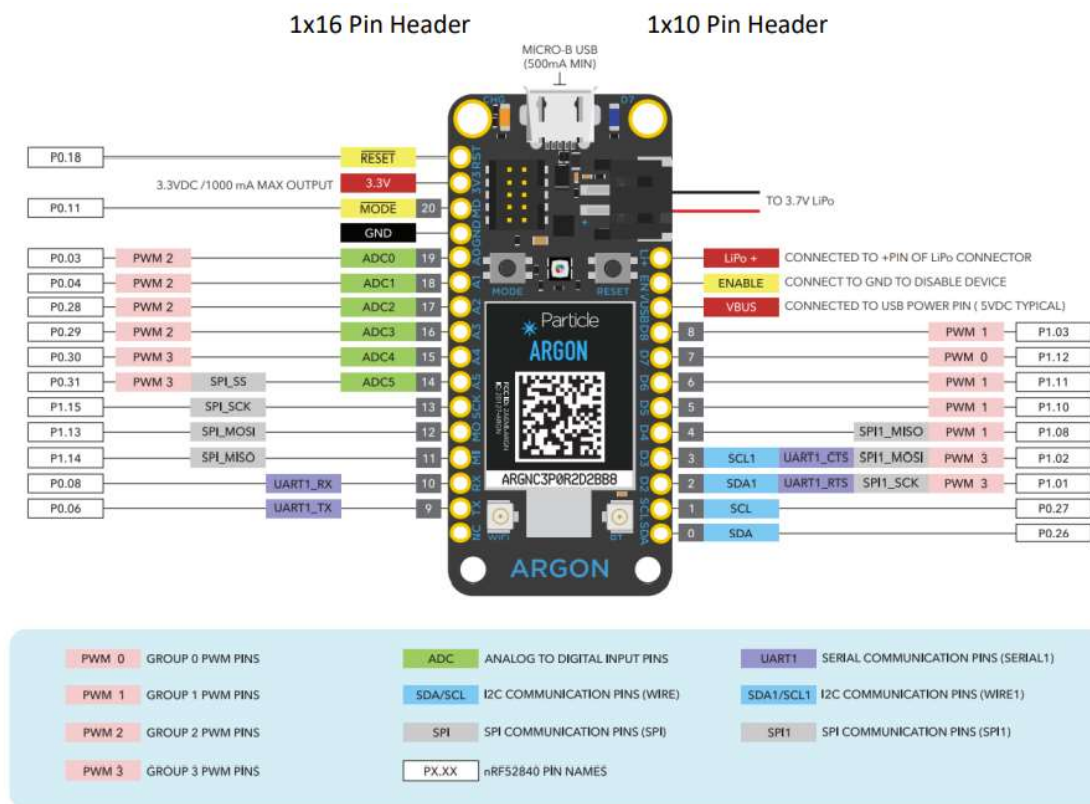
Figur 7 Et udsnit af testkoden

```
void loop() {

    //connect sensor to Analog 0
    analogvalue = analogRead(MoistSense);
    analogvalue = analogvalue * bitres;
    Serial.println(analogvalue);
    delay(100);
}
```

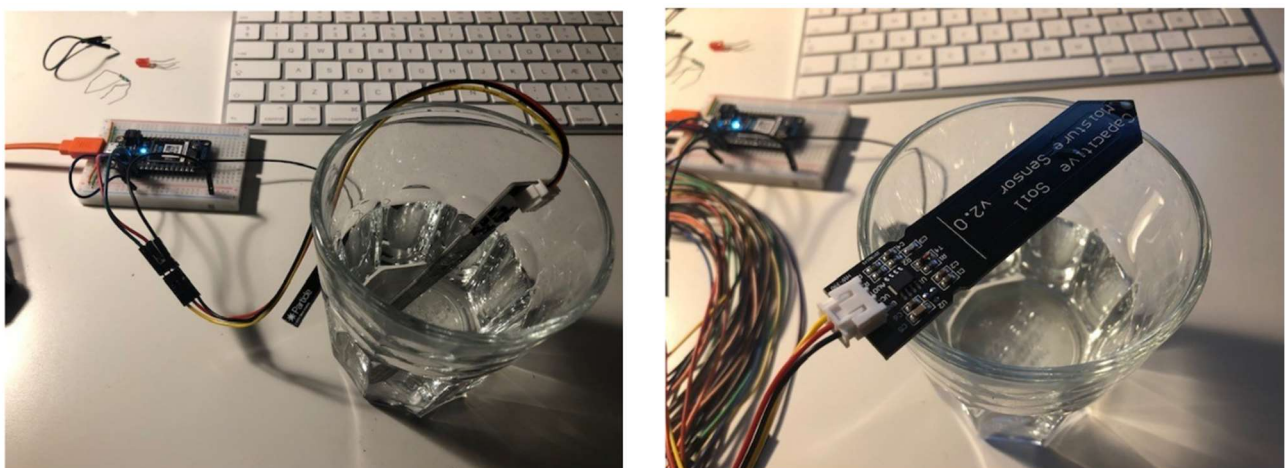
Figur 8 Et udsnit af testkoden

Herunder ses Argon header pinout [14]. Der er brugt pin A0 for at aflæse ADC værdier og A1 er brugt til at tænde slukke relæet.



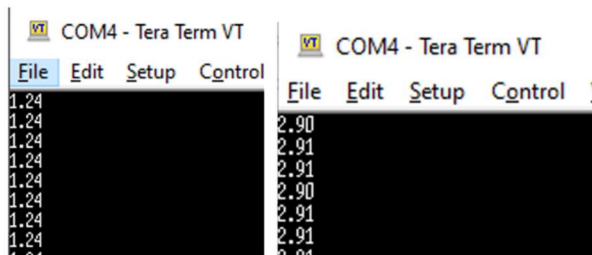
Figur 9 Argon header oversigt

Herunder ses testopstillingen på breadboard med fugtsensoren og Particle Argon [14]. Fugtsensoren er anvendt i to forskellige tilstande. I vandet, og udenfor vandet, for at observere tærskelværdier fra fugtsensoren. Det vil sige maksimale og minimale værdier i forbindelse med fugtniveauet.



Figur 10 Testopstillingen af fugtsensoren med Argon - i vandet og udenfor vandet

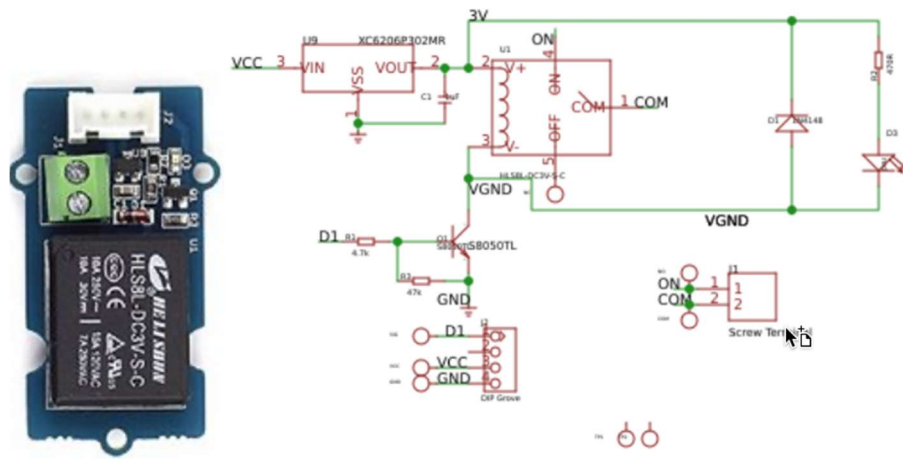
Herunder ses der to forskellige spændingsværdier, der er blevet aflæst ved hjælp af Argon og Tera Term.



Figur 11 Aflæsning af analog input fra Fugtsensor ved Tera Term

3.2.2 Interface: Particle Argon til Vandingsstyring

Vandingsstyringen består af et relæ og en aktuator(vandpumpe). Da vandpumpen [11] kræver 12 V forsyning, er der brugt et relæ til at tænde, og slukke for forsyningen, når der kommer et GPIO-signal fra *Particle Argon* til relæet [6]. Herunder ses diagrammet over relæmodulet.

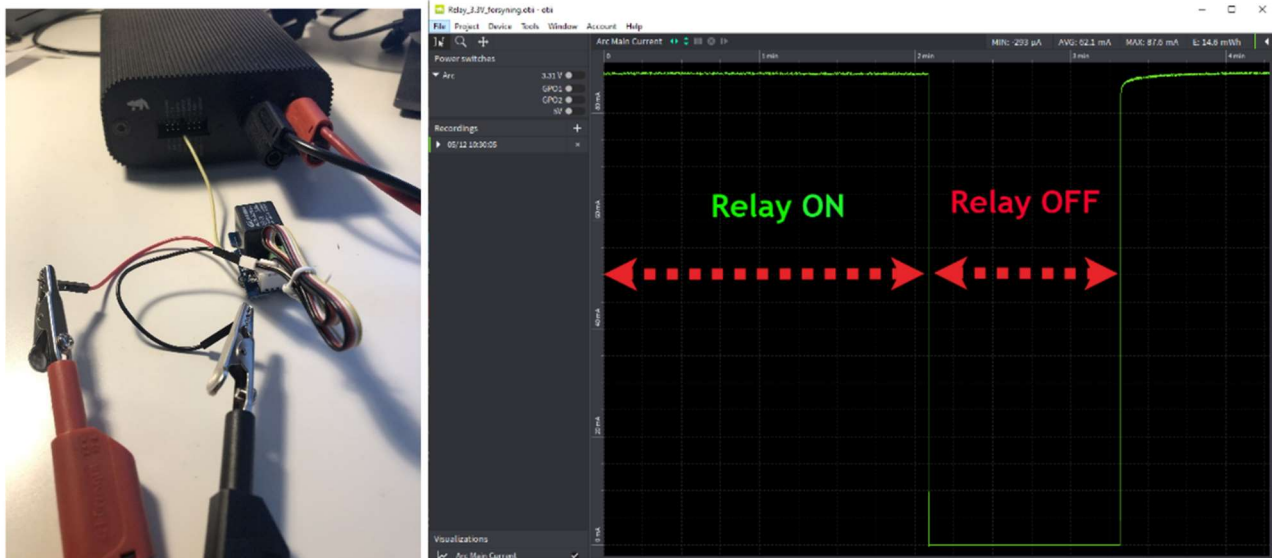


Figur 12 Relæmodulet og dens Schematic

Tabel 3 Pin connections af Relæmodulet

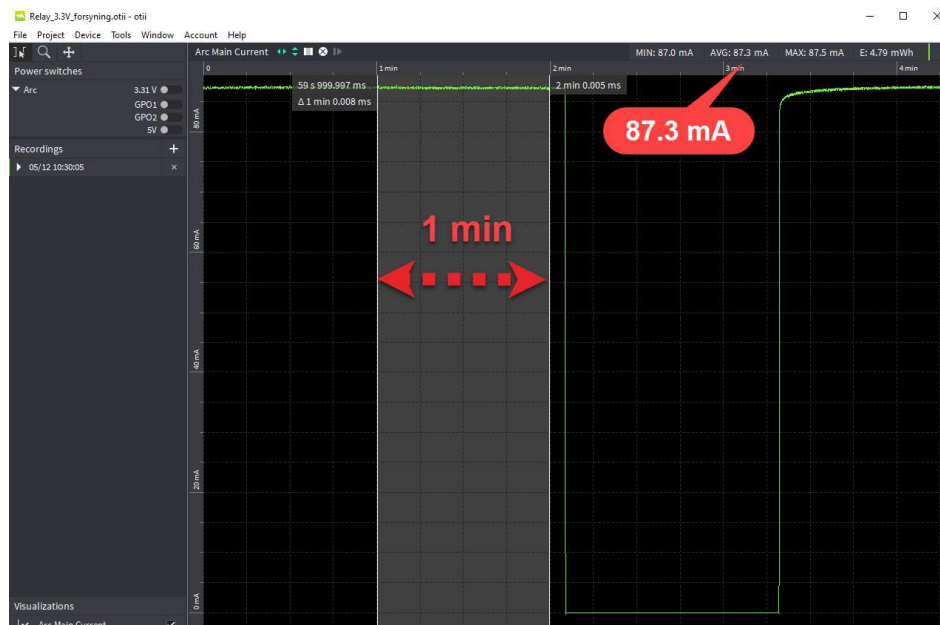
Name	Pin	Beskrivelse
ON	J1 - 1	Kontakt
COM	J1 - 2	Kontakt
GND	J2 - 1	Ground
VCC	J2 - 2	Forsyning
D1	J2 - 4	Signal

Herunder ses der testopstillingen af power profileringen og resultatet, hvor der ses at relæet er i aktiv og inaktiv tilstand. Under testen er der brugt GPO1 pin fra *Otii* [15] header til at tænde og slukke relæet og i øvrigt forsynet med 3.3V fra *Main Voltage*.



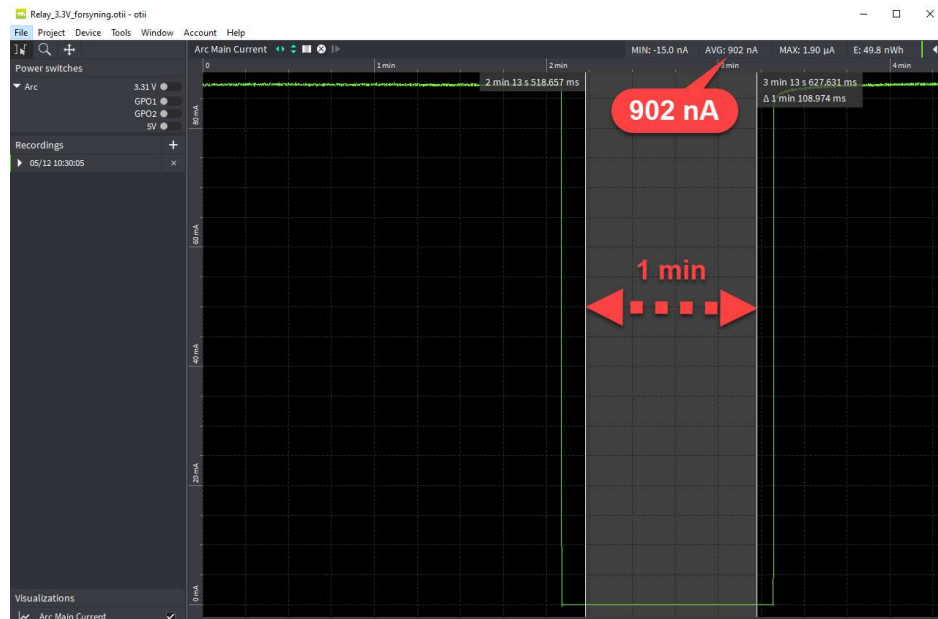
Figur 13 Testopstillingen med *Otii* power profiler og relæmodulets on/off tilstand

Der er målt 87.3 mA strømforbrug i løbet af et minut under aktiv tilstand. Se figuren herunder.



Figur 14 Strømforbruget af relæet i aktiv tilstand i løbet af 1 min

Til gengæld er der målt 902 nA under inaktiv tilstand. Se figuren herunder.



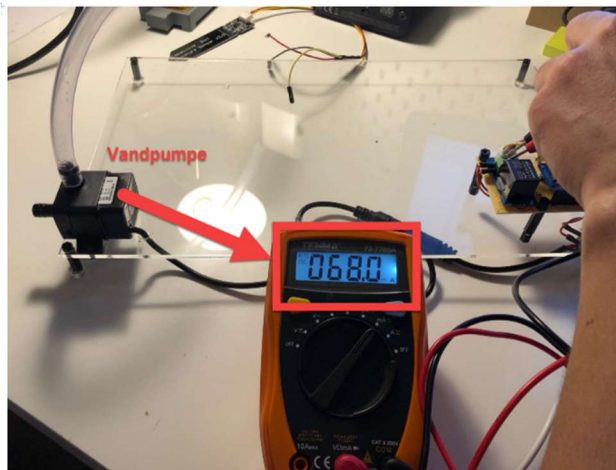
Figur 15 Strømforsbruget af relæet i inaktiv tilstand i løbet af 1 min

Vi har foretaget en ændring mht. magnetventil. Da magnetventilen kræver et højere tryk, end en flaske med bunden i vejret kan præstere, for at åbne ordentligt, og der ikke umiddelbart kunne findes på en læsning til det problem, er der taget en beslutning om at finde en anden måde, at udføre vandingen på. Der er købt en tilsvarende aktuator, som også kræver 12 V DC.



Figur 16 Vandpumpe

Herunder vises vandpumpen og dens strømmålingen, når den er forsynet med 12V. Der er målt 68,0 mA under aktiv tilstand. Men denne måling er foretaget uden at vandbeholderen er forbundet til vandpumpen. På figur 17 vises strømmålingen med multimeter, når vandet bliver suget fra vandbeholderen, hvilket medfører et højere strømforbrug på 188,1 mA, som er næsten tre gange større end uden vand.



Figur 17 Strømforbrug af vandpumpen. Venstre (uden vandbeholder) og højre (med vandbeholder)

3.2.3 Interface: Particle Argon til/fra Vejrtjeneste

For at få vejrudsigten skal der forbindes til en vejrtjeneste, som har et API. *Openweathermap.com* giver adgang til vejrdata. Billedet herunder viser deres *One Call API* som vil blive benyttet til dette projekt.

One Call API **NEW**

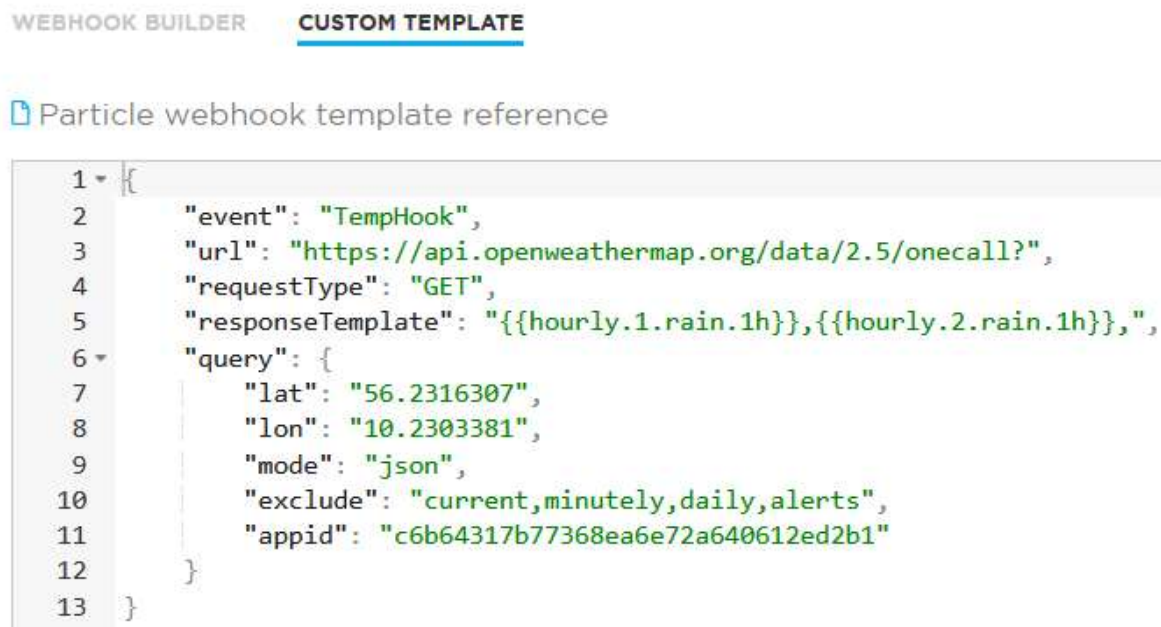
[API doc](#)[Subscribe](#)

- Make one API call and get current, forecast and historical weather data
- **Minute forecast** for 1 hour
- **Hourly forecast** for 48 hours
- **Daily forecast** for 7 days
- **Historical data** for 5 previous days
- **Government weather alerts**
- JSON format
- Available for both Free and paid subscriptions

Figur 18 Oversigt over de data der er adgang til

Til projektet skal der blot bruges *Hourly forecast* data, da vejrudsigten der kun skal kigges 2 timer frem.

Til at interagere med API'et vil particles eget webhook interface blive benyttet. På billedet herunder er det vist hvordan den er sat op. Den er også at finde på Github [webhooktemplate.json](#)[13].



Figur 19 Udklip fra den webhook der benyttes

"event": Navnet på webhooken, "TempHook"

"url": Adressen på forsendelsen.

"requestType": http metoden, i dette tilfælde GET.

"query": De parametre API'en skal bruge.

"lat": Breddegrad.

"lon": Længdegrad.

"mode": Hvordan forsendelsen ønskes formateret.

"exclude": Hvilke data man ikke ønsker i forsendelsen.

"appid": Den private nøgle man har fået af openweathermap.com.

Der laves en query hvor der liver bedt om data, i den query er det muligt at specificere hvilke data man ønsker, og her fravælges *current*, *minutely*, *daily* og *alerts*. Samtidig er det muligt, ved hjælp af Particles webhook at lave en *response template*.

En *response template* kan bruges til at sortere data i webhooken, det vil sige inden de sendes til Argon, og på den måde reducere den brugte hukommelse.

Templaten sørger for, at Particle Argon kun modtager 2 styk data, hver efterfulgt af et komma.

De data, der modtages, er udsigten til regn, angivet som forventet regn den seneste time, derfor vælges time 1 og time 2, i stedet for 0 og 1. Hvis der ikke er udsigt til regn, vil der ikke være nogen data, og svaret vil bestå af 2 kommaer.

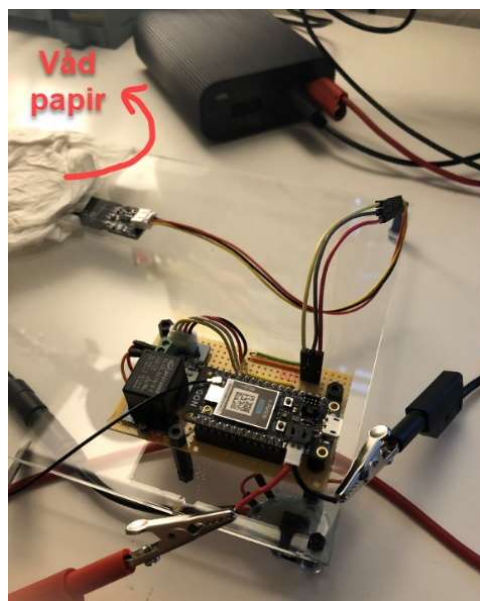
4 Power Profiling

Der er foretaget en samlet power profiling, hvor alle komponenter er forbundet til systemet. Herunder ses der resultater fra komponenterne, som er gennemgået i interface analysen.

Tabel 4 Oversigt over strømforbruget af komponenterne hver for sig

Komponent	Strømforbrug - AKTIV	Strømforbrug - INAKTIV
Fugtsensor	5.57 mA	0 mA
Relæmodulet	87.3 mA	902 nA
Vandpumpe	188.1 mA	0 mA

Herunder vises testopstillingen af komponenterne mht. power profiling af hele systemet. Particle Argon er forbundet til Otii og forsynet derfra med 3,7 V for at foretage power profilingen [12]. Vandpumpen bliver forsynet af en 12V DC-adapter.



Figur 20 Power profiling af hele systemet - Alle komponenter er forbundet

Strømforbrug er højere, når relæet er tændt, dvs. når vandingen er i gang. Herunder ses der at strømforbruget er 132 mA under vanding, imens det er 58,9 mA når der ikke vandes.



Figur 21 Power profiling af hele systemet.

Der vises alle strømforbrug under aktiv og inaktiv tilstand. For at tydeliggøre dette, er der skrevet to cases, hvor der bliver foretaget vanding (relay On) og uden vanding (relay Off). Desuden ses der også sleep mode strømforbrug, som er 5,41 mA. Under power profiling er alle `serial.println()` funktioner deaktiveret i programmet, da de også øger strømforbruget. Når systemet er klar til at bruge, har man i virkeligheden ikke brug for serial aflæsning.

Tabel 5 Oversigt over samlet strømforbrug

Cases	Samlet strømforbrug - AKTIV	Samlet strømforbrug - INAKTIV
Note: Alle komponenter er connected		
Tør eller mindre våd (relay ON)	132 mA + 188.1 mA (pumpe aktiv) Samlet = > 320,1 mA	5.41 mA
Mere våd eller Våd (relay OFF)	58.9 mA	

5 Implementering

5.1 Software

Implementeringen af webhook og vejrudsigt er til dels allerede forklaret i interface analysen, interaktionen med dem, vil blive berørt kort i implementeringen.

Software delen er tilgængelig på Github[13].

Til at starte med initialiseres diverse globale variabler og konstanter. Som det ses på billedet herunder, er der blandt andet tale om navne på webhooks, grænseværdier og pins. Debug variabelen sætter skal være sat til `false`, hvis man vil bruge produktet som tiltænkt, da der ellers ikke vil blive forespurgt på vejrudsigt og `sleepmode` vil være sat ud af kraft.

```
// Important stuff
#define HOOK_RESP "hook-response/TempHook"
#define HOOK_PUB "TempHook"
#define lessWet 3250 // Low Limit
#define moreWet 2025 // a bit higher Limit
#define Wet 1625 // Soaking Limit
int MoistSense = A0; // Moistsensor / ADC 0
int Relay = A1; // Moistsensor / Digital out
int analogvalue;
bool debug = true; // If we wanna debug
```

Figur 22 Variabler og konstanter

Softwaren består af blot 5 funktioner, *Setup*, *loop*, *myHandler*, *Sleep* og *water*.

Funktionen *setup*, er en form for initialisering af system funktioner og perifere enheder.

```
// setup() runs once, when the device is first turned on.
void setup()
{
    // Put initialization like pinMode and begin functions here.
    Particle.subscribe(HOOK_RESP, myHandler, MY_DEVICES);
    pinMode(Relay, OUTPUT);
    Serial.begin(115200);
}
```

Figur 23 En snip af setup()

Som det ses på billedet herover, bruges *setup* blandt andet til at initialisere *subscribe* funktioner, til at sætte eventuelle *pinmodes* og konfigurerings af den serielle forbindelse.

Loop funktionen sørger for at programmet aldrig afsluttes, men blot kører i ring. Den starter med at læse den analoge værdi fra fugtsensoren og afprøver derefter forskellige *if* statements for at finde et sted hvor den passer ind, for til sidst at slutte af med at eksekvere *sleep* funktionen.

If-sætningerne handler udelukkende om fugtigheden, læst af sensoren. Såfremt fugtigheden er meget lav, startes funktionen *water*. Hvis den er let eller mere fugtig, sender den bud efter en vejrudsigt og bevæger sig ned mod *sleep*.

MyHandler funktionen er en asynkron callback funktion. Den aktiveres ikke direkte af koden i *loop*, men aktiveres når der kommer en besked ind med emnet "*hook-response/TempHook*". Såfremt der ikke er udsigt til regn, vil beskeden være tom, bortset fra 2 kommaer. Hvis der er udsigt til regn vil beskeden være længere.

Når den lokale vejrudsigt er ankommet, vil der blive foretaget en beslutning om der vandes eller ikke, ud fra sensorens måling og vejrudsigten. Hvis sensoren måler let fugtighed, og der er regn indenfor de næste 2 timer, vil den ikke vande. Derimod hvis der ikke er udsigt til regn, vil den eksekvere *water*.

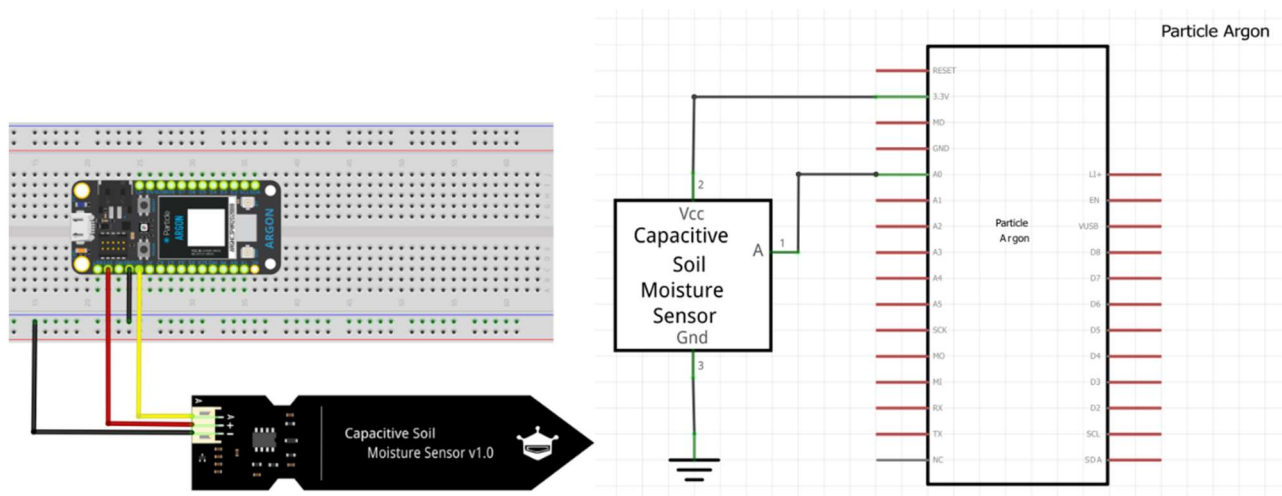
Sleep funktionen er beregnet til spare strøm, når der er styr på fugtigheden ved planten, skulle den i teorien kunne sove i et par timer, det er længden på vores vejrudsigt. Det ville dog blive en lang og kedelig demonstration, så derfor er den sat til 30 sekunder.

Water funktionen styrer vandingen. Funktionen tager 1 argument, og det er hvilken threshold vi skal over, det kunne for eksempel være *lessWet*, der har en værdi på 3250, eller 2,6V.

5.2 Hardware

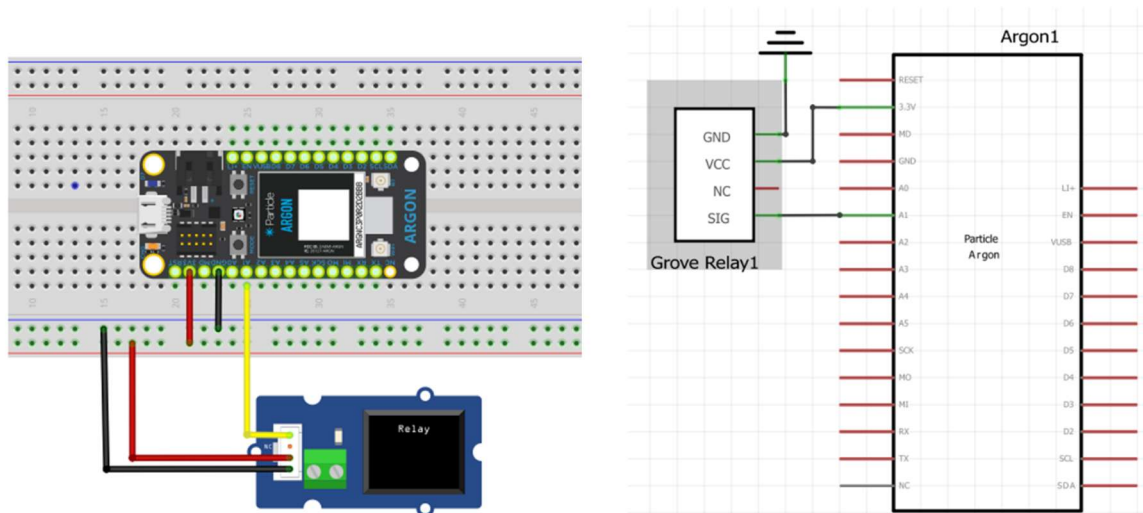
Implementeringen af hardware er foretaget med den source kode, som er gennemgået under software implementering. Alle HW breadboard og schematic implementeringer er tegnet med *Fritzing Electronic Design* program [9]

Herunder ses implementeringen af fugtsensoren med Particle Argon.



Figur 24 Implementeringen af fugtsensor med Particle Argon

Herunder ses implementeringen af relæet med Particle Argon. [6]



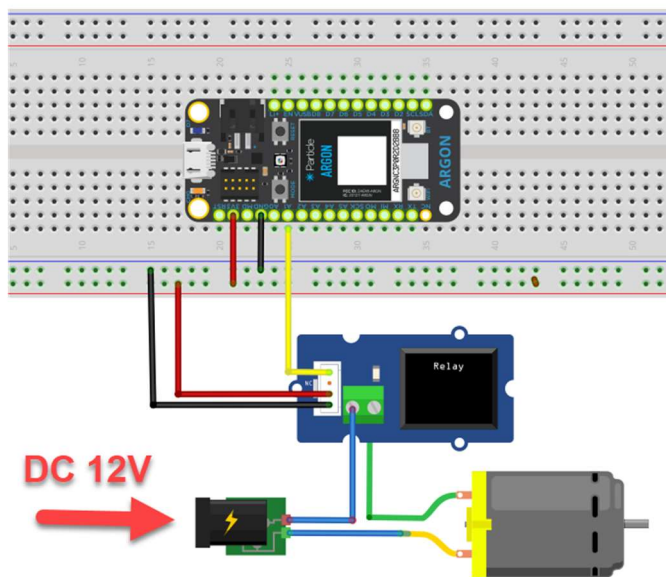
Figur 25 Implementeringen af relæmodulet med Particle Argon

Der er defineret nogle digitale værdier, som er konverteret fra analog-input [13]. Vanding skal foregå, når værdien kommer over 3250, som indikerer at der ikke er nok fugtighed. Derfor skal relæet tændes og vandingen skal startes inden fugtighedsniveau kommer under 3250. Derefter skal stoppes

```
void water(int temp)
{
  digitalWrite(Relay, HIGH);
  Serial.println("Starting to water!");
  |
  // 2,6V || 3250
  while (analogvalue > temp)
  {
    analogvalue = analogRead(MoistSense);
    Serial.println(analogvalue);
  }
  Serial.println("Stopped watering!");
  digitalWrite(Relay, LOW);
};
```

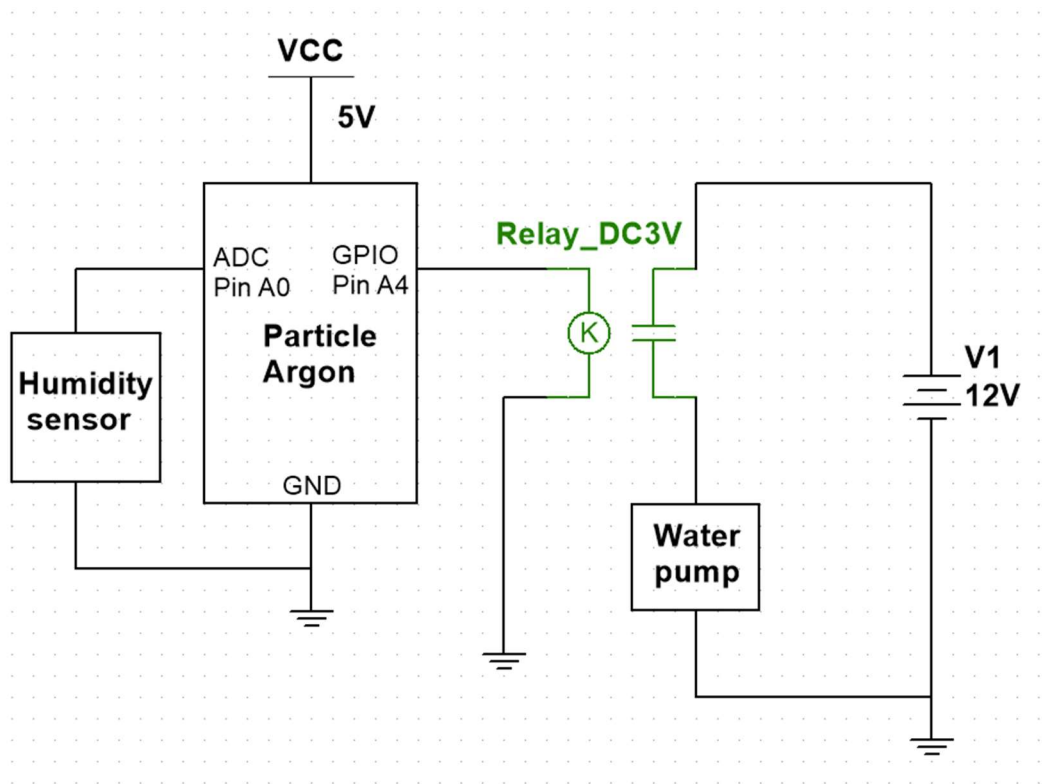
Figur 26 Et udsnit af main program, som viser relæet aktiveret og deaktiveret

Herunder ses implementeringen af vandpumpen [11]. Vandpumpen er forbundet til relæets kontakter, som reagerer på samme tid, når relæet aktiveres af Particle Argon.



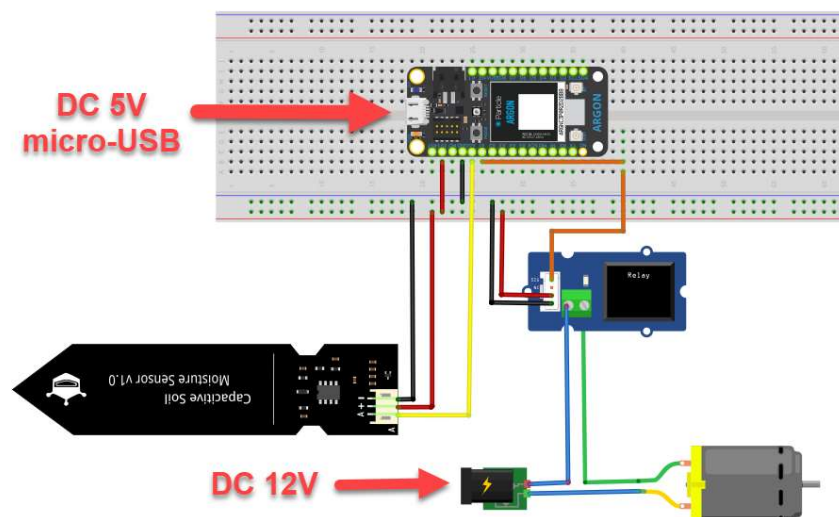
Figur 27 Implementeringen af vandpumpen

Herunder ses hele kredsløbet, som er designet i *Multisim*.



Figur 28 Visning af hele systemet i Multisim

Herunder ses hele kredsløbet på digital breadboard, som er designet i *Fritzing Electronic Design* programmet. [9]



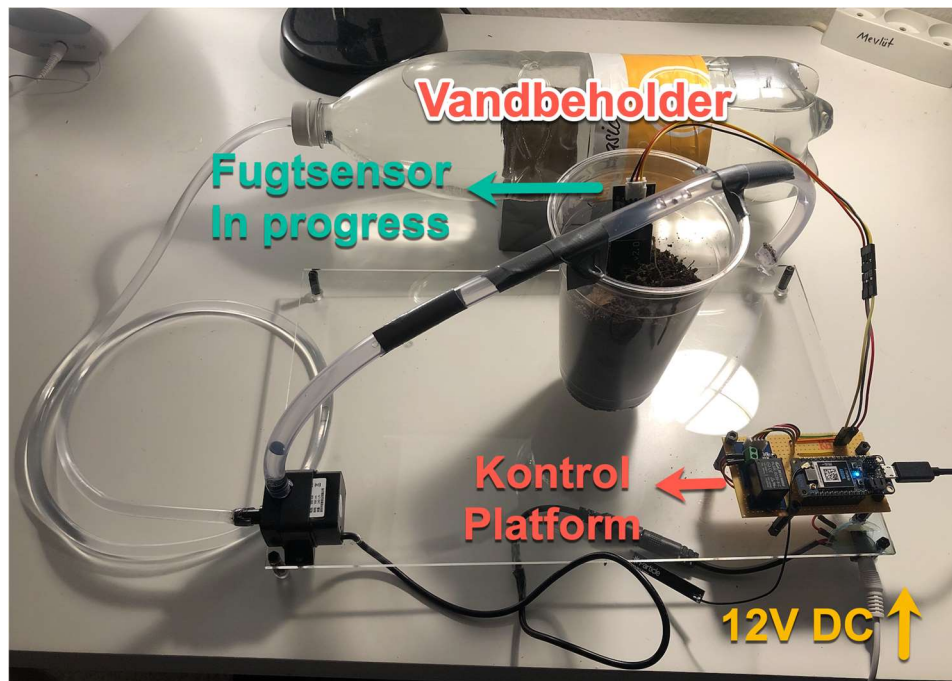
Figur 29 Visning af hele systemet i Fritzing Electronic Design programmet

Vandingssystem fungerer ud fra de data, der kommer fra fugtighedssensoren. Det vil sige at de tærskelværdier mht. fugtigheden i jorden er aflæst igennem analogt input i Particle Argon, der derefter vil kontrollere om vanding skal foretages eller ej. Der er også andre faktorer, der påvirker systemets kørsel. Hver anden time skal vejrudsigten tjekkes fra openweathermap.com og behandles, sammen med fugtighedsniveauet i jorden omkring planten, og derefter skal det vurderes om planten skal vandes eller ej.

6 Test / Verifikation

De udvalgte tests demonstrerer funktionaliteten i produktet. Test 1 er en form for sikkerhedskontrol, i forhold til planten, da den virker uden internet. Hvis føleren måler lav fugtighed, igangsættes vandingen.

Test 2 er en demonstration af en anden fordel ved produktet, besparelsen i vand, skåning af miljøet. Der vandes ikke hvis fugtigheden er lav, men tilstrækkelig, og der er udsigt til regn i nær fremtid.



Figur 30 Visning af hele systemet, som er monteret på plexiglas

Der er oprettet en debug model af koden, der gør at man selv kan bestemme udsigten til regn.

6.1.1 Test 1: Tør jord

Kræves: Tør jord, live- eller debugkode.

Trin 1:

Sensor og pumpe (med adgang til vand) tilsluttes og Argon tændes.

Forventet resultat:

Sensoren konstaterer at der er tørt. Argonen starter pumpen med det samme og lader den køre indtil jorden ikke måles som tør.

Resultat: Da Argonen starter op, måler sensoren en, for lav, fugtighed, og Argonen starter pumpen.

6.1.2 Test 2: Let fugtig jord (lessWet)

Kræves Let fugtig jord, live- eller debugkode.

Trin 1:

Sensor og pumpe (med adgang til vand) tilsluttes og Argon tændes.

Forventet resultat ved livekode: Argonen vil, såfremt der ikke kommer regn, starte pumpen, hvis der kommer regn vil den ikke.

Delresultat: N/A

Trin 2:

Forventet resultat ved debugkode:

Der sker ikke noget når Argon startes.

I CLI skrives der følgende: *particle publish hook-response/TempHook '1.20,,'*.

I teraterm vil den skrive at der er regn på vej, så den gør ikke yderligere.

Delresultat: Argonen konstaterer at der kommer regn indenfor de næste 2 timer, og foretager sig ikke yderligere.

Efterfølgende kan man skrive følgende: *particle publish hook-response/TempHook ',,'*

og se at pumpen går i gang.

Delresultat: Da Argonen konstaterer at der ikke kommer regn indenfor de næste 2 timer, starter den pumpen indtil fugtigheden er steget tilpas.

6.1.3 Resultat af Test / verifikation

Testene fungerede som forventet, og jorden endte med at være dejlig fugtig.

7 Konklusion og perspektivering

Det endte med et minimal viable produkt, der sagtens kan forbedres og udvides. De basale ting virker, og en eventuel plante ville aldrig dø af tørst.

Desværre blev tiden for knap til indførsel af de optionelle features der var stillet i udsigt, og som vel egentlig var noget af det nyskabende. Muligheden for ikke at vande mens planten bliver bagt i solen midt i en hedebølge, hvor man ellers kunne tilrettelægge en ordentlig vanding, men udenfor de hede solskinstimer, hvor planten i stedet for at få slukket sin tørst, ville blive kogt. Eller om vinteren, hvor man kunne holde en minimal mængde vand hos planten, ved at vande når der ikke er frost, og vande tilpas lidt, til at krukkerne ikke sprænges.

8 Referencer

Ref	Beskrivelse	Link
[1]	Capacitiv Jord fugtigheds sensor v2.0	https://www.ardu.dk/shop/capacitiv-jord-fugtigheds-sensor/ Ardu er desværre lukket i mellemtiden, og det har ikke været muligt at finde andre referencer.
[2]	Fugtsensor	https://www.ardu.dk/shop/fc-28-jord-fugtigheds-sensor/ Ardu er desværre lukket i mellemtiden, og det har ikke været muligt at finde andre referencer.
[3]	Magnetventil	https://www.ardu.dk/shop/elektrisk-vand-ventil-12v-dc-lige/ Ardu er desværre lukket i mellemtiden, og det har ikke været muligt at finde andre referencer.
[4]	Particle Argon with humidity sensor – IoT project introduction	https://www.youtube.com/watch?v=g43C1cRHEA4 Besøgt 13/12 20
[5]	E5IoT Project – Tommy and Mevlüt	https://youtu.be/UPIEpSagpd0 Besøgt 13/12 20
[6]	Grove_Relay_Schematic	https://easyeda.com/antimix/Grove_Relay-xuoGASa4m Besøgt 13/12 20
[8]	Ears Requirement Capture	EARS Requirement Capture - EUDP Besøgt 13/12 20
[9]	Fritzing - Schematic drawing program	https://fritzing.org/ Besøgt 13/12 20
[11]	Vandpumpe	https://arduinotech.dk/shop/12v-submersible-water-pump-240lh-vand-pumpe/ Besøgt 13/12 20
[12]	Particle Argon Power	https://docs.particle.io/datasheets/wi-fi/argon-datasheet/#lipo Besøgt 13/12 20
[13]	Teams GitHub IoTSemPro	https://github.com/Vojens/IOTSemPro Besøgt 13/12 20
[14]	Particle Argon Datasheet	https://docs.particle.io/datasheets/wi-fi/argon-datasheet/ Besøgt 13/12 20
[15]	QOITECH - Otii Power Analyser	https://www.qoitech.com/ Besøgt 13/12 20