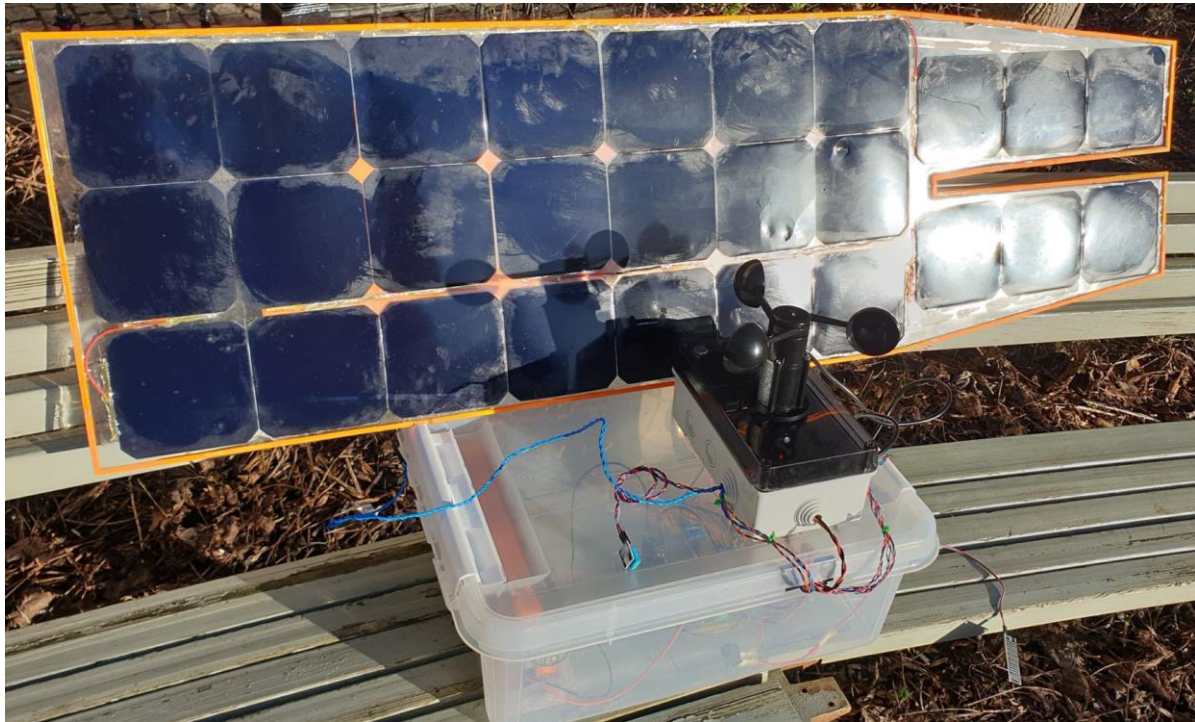




CHALMERS



Skalbar IoT plattform för mikroelnät

Konstruktion av prototyp för datainsamling, hantering och visualisering av data relaterat till mikroelnät

Examensarbete inom högskoleingenjörsprogrammet Mekatronik

LUAN MOLLAKUQE

Institutionen för Elektroteknik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2019/2020

Skalbar IoT Plattform för mikroelnät

Konstruktion av prototyp för datainsamling, hantering och visualisering av data relaterat till

mikroelnät

LUAN MOLLAKUQE



CHALMERS

Skalbar IoT Plattform för mikroelnät

Konstruktion av prototyp för datainsamling, hantering och
visualisering av data relaterat till mikroelnät
LUAN MOLLAKUQE

© LUAN MOLLAKUQE, 2019/2020

Handledare: Niels Boardman Jonsson, Olle Norelius, Infotiv AB
Examinator: Bertil Thomas, Chalmers, Institutionen för elektroteknik

Institutionen för Elektroteknik
Chalmers tekniska högskola
412 96 Göteborg
Sverige
Telefon: 031-772 1000

Sammanfattning

Med ett globalt växande energibehov, blir förnyelsebar energi i form av solenergi en allt mer relevant energikälla. Den ökade medvetenheten kring hotet av klimatet har även genererat folkopinioner som påverkat politiska beslut riktade mot solenergisektorn [1]. Användningen av solenergi i Sverige ökade med 82% från 2017 till 2018 [2].

Parallellt med den växande solenergimarknaden ökar även antalet uppkopplade elektroniska system och utrustning. Sakernas internet (IoT) är ett samlingsbegrepp, vilket syftar på smart elektronisk utrustning som kan samla in data och kommunicera via uppkopplade nätverk, utan mänsklig interaktion.

Genom att tillämpa uppkopplade enheter för att samla in data relaterat till mikronät, kan systemets effektivitet och solenergiproduktion analyseras, samt hur verkningsgrader påverkas av diverse väderförhållanden.

I detta projekt konstruerades en IoT plattform för att samla in, behandla och visualisera data relaterat till mikronät baserade på solenergi. Plattformen består av en sensornod med 10 olika sensorer för att samla in mikronätets effektdata samt väderförhållanden som potentiellt kan påverka mikronätets verkningsgrad. Sensornoden är uppkopplad till en centralenhet. Centralenheten hanterar insamlade data, som därefter skickas vidare till en server för lagring. En webbapplikation byggdes med hjälp av ramverket Flask och visualiseringsverktyget Grafana, för att visualisera insamlad sensordata i ett grafiskt gränssnitt över internet.

Prototypen som togs fram konstruerades på ett skalbart sätt, vilket möjliggör att utvidga användningsområdet genom att tillföra fler sensornoder och även ersätta, lägga till eller ta bort sensorer.

Nyckelord: Smart, IoT, skalbarhet, datainsamling, mikronät, solenergi

Abstract

With a growing energy demand globally, renewable energy alternatives are becoming increasingly relevant as a source of energy. The increased awareness of the climate crisis has also generated public opinions which affect the political decisions towards expanding the usage of solar power in the energy sector [1]. The use of solar energy in Sweden increased by 82% from 2017 to 2018 [2].

In parallel with the growing solar energy market, the number of connected electronic systems and equipment is also increasing. The term Internet of Things (IoT), describes these types of smart systems which collect data and communicate via network, without human interaction.

By using connected devices to collect data related to micro grids, one can analyze the system's solar energy production, efficiency and how efficiencies exist in different departments.

In this project a prototype of an IoT platform was constructed, to collect, process and visualize data related to microgrids based on solar energy. The platform contains a sensor node with 10 different sensors for collecting power data of a microgrid system and weather conditions that potentially affect the efficiency of the microgrid. The sensor node is connected to a central unit that handles collected data and thereafter transmits the data wirelessly to a server for storage. A web application was built using Flask framework and the visualization tool Grafana for visualization which can be accessed over the internet.

The prototype was developed in regard to scalability, which allows the user to replace, add or remove sensors from the sensor node, and also add more sensor nodes connected to the central unit.

Keywords: Smart, IoT, scalability, data collection, microgrid, solar energy

Förord

Denna rapport omfattar hårdvaru- och mjukvaruutveckling av en IoT plattform för att samla in data relaterat till mikronät baserade på solenergi. Arbetet utfördes i uppdrag av Infotiv AB, Göteborg, som examensarbete (15 högskolepoäng) på Chalmers tekniska högskola, institutionen för elektroteknik.

Ett stort tack till Infotiv AB som möjliggjorde detta projekt. Även ett stort tack till mina handledare Niels Boardman Jonsson, Olle Norelius. Er kunskap, hjälp och engagemang har varit till stor hjälp under projektet.

Luan Mollakuqe, Göteborg 2020

Terminologi

Mikroelnät/Mikronät	Småskaligt elnät
IoT	Sakernas internet (Internet of Things), uppkopplad elektronisk utrustning med integrerade sensorer och internetkommunikation.
PV	Fotovoltaik (Photovoltaic), teknik för att omvandla ljusenergi till elektrisk energi.
PCB	Kretskort (Printed Circuit Board)
DC	Likström
AC	Växelström
UART	Seriellt dataöverföringsprotokoll
Ramverk	Kodbibliotek med en existerande uppsättning av funktioner.
A/D-omvandling	Analog till digital signalomvandling
RTD	Motståndstermometer (Resistance thermometer)
Python	Programspråk

Innehållsförteckning

Sammanfattning	I
Abstract	II
Förord.....	III
Terminologi.....	IV
Innehållsförteckning	V
1 Inledning	1
1.1 Bakgrund	1
1.2 Syfte	2
1.3 Mål	2
1.4 Precisering av frågeställning	2
1.5 Avgränsningar	2
2 Teori/Teknisk bakgrund.....	4
2.1 Solceller.....	4
2.2 MPPT	4
2.3 Sensorer.....	4
2.3.1 DHT11 Temperatur- och luftfuktighetsmätare	4
2.3.2 Motståndstermometer	4
2.3.3 Vindmätare.....	5
2.3.4 Regnsensor	6
2.3.5 Strömmätare	6
2.3.6 Spänningsmätare	6
2.4 Batterikapacitet.....	6
2.5 A/D-omvandlare.....	7
2.6 Spänningsregulator.....	7
2.7 Flask framework.....	7
2.8 InfluxDB.....	8
2.9 Grafana	8
3 Metod	9
3.1 Förstudie.....	9
3.2 Kravspecifikation	9
3.3 Val av hårdvara och systemuppbyggnad.....	9
3.4 Mikronätssystem	9
3.5 Mjukvara	9
4 Genomförande.....	11

4.1	Förstudie.....	11
4.2	Kravspecifikation	11
4.3	Val av sensorer	12
4.4	Systemöverblick.....	12
5	Hårdvara.....	14
5.1	Testsystem.....	14
5.2	Sensornod	14
5.2.1	Kretskort	16
5.2.2	Temperatur och luftfuktighet	17
5.2.3	Vindhastighet	17
5.2.4	Regnsensor	18
5.2.5	Spänningsmätare	18
5.2.6	Strömmätare	18
5.3	Centralenhet	19
5.4	Mekanisk konstruktion.....	20
6	Mjukvara	21
6.1	Sensornod.....	21
6.1.1	Konfigurering.....	21
6.1.2	Huvudprogram	21
6.1.3	Filtrering	22
6.2	Centralenhet	22
6.3	Databas	23
6.4	Dataöverföring	24
6.5	Visualisering.....	24
7	Test och Kalibrering	25
7.1	Kalibrering	25
7.2	Test i kontrollerad miljö.....	25
7.3	Test i utomhusmiljö.....	27
8	Resultat	29
8.1	Skalbarhet.....	29
8.2	Tester.....	29
9	Diskussion.....	30
9.1	Begränsningar och problem	31
9.1.1	Temperaturmätning.....	31
9.1.2	Effektavvikelse	31
9.1.3	Mätning av batterikapacitet.....	31
9.2	Förslag på vidareutveckling	31

9.2.1	Val av sensorer.....	31
9.2.2	Dataöverföringsteknik.....	32
9.3	Slutsats	32
	Referenser	33
	Bilagor.....	34
	Bilaga 1: Systemdiagram	34
	Bilaga 2: Kretsschema för hela systemet	35
	Bilaga 3: Kretsschema för centralenheten	36
	Bilaga 4: Kretsschema för sensorklustret	37
	Bilaga 5: Centralenhetens mjukvara	38
	Bilaga 6: Testdata från inomhustest.....	39
	Bilaga 7: Testdata från utomhustest.....	40
	Bilaga 8: Portbeskrivning	41
	Bilaga 9: Mjukvara	42
	Bilaga 10: Datablad för solceller	43
	Bilaga 11: Testlådans innehåll	44
	Bilaga 12: Sensornodens innehåll.....	45

1 Inledning

1.1 Bakgrund

Mikronät är småskaliga elnät som kan fungera oberoende eller som ett komplement till regionalt distribuerade elnät. Genom att använda mikronät möjliggörs bland annat tillgång till el på platser där det inte tidigare varit möjligt. Om mikronät används som kompletterande energikälla leder det därtill minskad belastning på de distribuerade elnät som försörjer bostäder, näringsliv och industrier. Samtidigt främjas miljön på grund av minskad användning av icke förnybara energikällor så som fossila bränslen med höga utsläpp av koldioxid, svaveloxid och kväveoxider, eller kärnkraft som resulterar i radioaktivt avfall [3].

Med ett globalt växande energibehov, blir solenergi en allt mer efterfrågad energikälla som har en mindre påverkan på miljön i form av utsläpp än nuvarande icke förnybara energikällor. Med en ökande medvetenhet kring hotet av klimatet har även folkopinioner påverkat politiska beslut som främjat utveckling och tillämpning av solenergi [1]. Under 2018 växte antalet solenergi-kapacitet baserat på fotovoltaik-teknik (PV) med över 103.2 GW i hela världen, vilket ger en total kapacitet på 409.1 GW i världen. Enbart i Sverige installerades ytterligare 158 MW under år 2018, vilket gör att den totala kapaciteten av PV-paneler uppgår till 426 MW, eller 0.3% av totala elkonsumtionen i Sverige [2].

SWEDEN

FINAL ELECTRICITY CONSUMPTION 2018	141	TWh
HABITANTS 2018	10	MILLION
AVERAGE YIELD	950	kWh/kW
2018 PV ANNUAL INSTALLED CAPACITY	159	MW _{DC}
2018 PV CUMULATIVE INSTALLED CAPACITY	426	MW _{DC}
PV PENETRATION	0,3	%

Figur 1: Sveriges elkonsumtion 2018 [2].

Parallellt med den växande solenergimarknad och ökad miljömedvetenhet, ökar även antalet uppkopplade system och IoT-enheter. Det handlar om smart utrustning med integrerad elektronik med tillhörande sensor(er) som möjliggör datainsamling kopplat till utrustningen, som därefter kan överföras och hanteras direkt över nätet. Dessa sensorer kan kommunicera via lokala anslutningar så som Wi-Fi, Bluetooth samt NFC, eller mobilnät så som 3G/4G/5G.

Transportföretag kan med hjälp av IoT exempelvis lokalisera och bevaka sina fraktcontainrar. Industrier kan undvika plötsliga utrustningsfel och avvikelser genom att hålla koll på utrustningens status. Privatpersoner kan använda IoT för att via mobilen aktivera tvättmaskinen, öppna bilen eller tända och släcka sina lampor i hemmet direkt från mobilen [4].

Mot denna bakgrund är datainsamling av solenergiproduktion med hjälp av uppkopplade system intressant från slutanvändarens perspektiv. Genom att ansluta sig till

mikronätsystemet med uppkopplade sensorer, kan man enkelt komma åt relevant data beroende på ändamål. Användaren kan, till exempel, koppla upp sig för att analysera hur mycket solenergi som producerats under ett visst tidsintervall och hur väderförhållanden påverkat energiproduktionen.

1.2 Syfte

Syftet med projektet är att tillverka en prototyp av en IoT plattform för att samla in, hantera och visualisera data relaterat till mikroelnät. Genom att samla in data från mikroelnät och omgivningens påverkan i form av väderförhållanden, kan en överblick av systemet och de påverkande variablerna visas. Insamlade data och korrelationen mellan väder och solenergi kan därefter analyseras. Prototypen kommer bli basen för datainsamling för framtida projekt inom förnybar energi.

1.3 Mål

Målet med arbetet är att konstruera en plattform bestående av hårdvara, mjukvara och sensorer för att samla in data relaterat till mikronäts energiproduktion samt väderförhållanden.

Plattformen ska konstrueras skalbart, beroende på mikronätets storlek och även öppna upp möjligheten för tillämpningar på andra system där liknande datainsamlingsverktyg är intressant.

Delmål:

- Ta fram lämplig hårdvara och sensorer för prototypens ändamål
- Konstruera ett kretskort för noden där sensorerna monteras
- Mjukvara för sensornod
- Mjukvara för centralenhet
- Databas för lagring av sensordata
- Visualisering av data via användargränssnitt
- Nätverksbaserad hemsida för att nå användargränssnittet

1.4 Precisering av frågeställning

- Vilken data är relevant att samla in med hjälp av prototypen för framtida forskning och utveckling inom mikronät?
- Hur kan insamlad data användas?
- Vilka delar av systemet kan utvecklas skalbart för att i framtiden bredda användandet av systemet?

1.5 Avgränsningar

- Projektet avser enbart en prototyp, det vill säga en demoversion som agerar som en modell för att påvisa konceptet.
- Prototypen består av två enheter:

- En sensornod baserat på en Arduino Mega 2560, egentillverkat kretskort och sensorer.
 - En centralenhet baserat på Raspberry Pi 3 Model B+.
- Budgeten för projektet är max 2000 SEK.
- Budgeten avser ej mikronätet (solpanel, batteri, MPPT-regulator).

2 Teori/Teknisk bakgrund

I följande avsnitt presenteras tekniska bakgrunden till de begrepp och komponenter som används i projektet.

2.1 Solceller

Solceller, eller fotovoltaiska celler, är halvledardioder baserade på kisel. Genom att använda den fotoelektriska effekten som är ett fysikaliskt fenomen, omvandlar cellerna solenergi i form av fotoner till elektrisk energi (likström). Energin kan därefter användas av elektrisk utrustning eller lagras i exempelvis uppladdningsbara batterier.

2.2 MPPT

MPPT-regulator "Maximum power point tracking regulator", är en regulator som används för att reglera spänning och ström som genereras av solpaneler, så att den optimala effekten levereras till batteriet som laddas upp. Genom att mäta spänningen på batteriet och solpanelen, anpassar regulatorn spänningen på solpanelen så att den genererar den optimala spänningen och strömmen för att ladda upp batteriet.

Regulatorn ser även till att bryta kretsen mellan solpanel och batteri om spänningen på solpanelen hamnar under ett visst tröskelvärde, vilket är viktigt eftersom solpanelen alltid måste ha högre spänning än batteriet. I annat fall kommer strömmen flöda från batteriet till solpanelen, vilket på grund av negativ potentialskillnad kan orsaka skador på systemet.

2.3 Sensorer

2.3.1 DHT11 Temperatur- och luftfuktighetsmätare

DHT11 är en enkel lågprissensor som kan mäta temperatur inom intervallet 0–50°C och luftfuktighet mellan 20–80%. Sensorn har en noggrannhet på 5% för luftfuktighetsmätning och $\pm 2^\circ\text{C}$ för temperaturmätning.

2.3.2 Motståndstermometer

Motståndstermometer (RTD) är en typ av motstånd vars resistans beror på temperatur. PT1000 en platinabaserad motståndstermometer med ett nominellt motstånd på 1000 Ω vid 0°C. Sensorn är av typen positiv temperaturkoefficient (PTC), vilket innebär att vid hög temperatur ges en hög resistans och vid låg temperatur ges en låg resistans.

För att beräkna temperaturen används följande formel:

$$T = \frac{-R_0 * A + [(R_0 * A)^2 - 4 * R_0 * B * (R_0 - R)]^{\frac{1}{2}}}{2 * R_0 * B} [^\circ\text{C}] \quad (1)$$

Där:

T = Temperatur i $^\circ\text{C}$

R = Uppmätt motstånd i Ω

R_0 = 1000 Ω (nominella motståndet vid 0°C för PT1000)

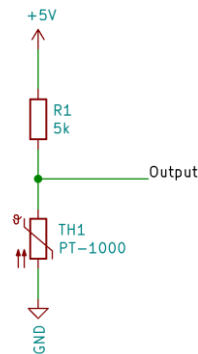
A, B, C är koefficienter enligt följande:

$$A = 3.9083 * 10^{-3} \text{ } ^\circ\text{C}^{-1}$$

$$B = -5.775 * 10^{-7} \text{ } ^\circ\text{C}^{-2}$$

$$C = 3.9083 * 10^{-12} \text{ } ^\circ\text{C}^{-4}$$

Den maximala strömmen som får passera genom PT1000-sensorn är 1mA. Därav krävs ett 5k Ω motstånd i serie (enligt ohms lag $R = U/I$) för att inte det ska gå för hög ström genom sensorn. Se figur 2 för kopplingsschema.



Figur 2: Kretsschema för PT1000

PT1000-sensorn kan koppas med två-, tre- eller fyra-ledarkoppling. Fördelen med fyra-ledarkoppling är att strömmatning och mätning av motståndet i sensorn är separerade, till skillnad från två-ledarkoppling, där mätning och strömmatning går genom samma ledare. Två-ledarkoppling är enklare att installera än de övriga alternativen, men resulterar i större mätfel på grund av variationer i ledningsmotståndet som uppstår, speciellt vid längre kabellängder. Därav bör två-ledarkoppling endast användas vid applikationer där sensorn är lokaliserad i närhet till den avläsningsenheten och där det inte ställs höga krav på mätvärdets precision.

2.3.3 Vindmätare

För att mäta vindhastighet används en vindsensor som drivs på 7-24V. När sensorn utsätts för vind genereras en spänning V_s mellan 0.4 till 2V. Spänningen kan avläsas och omvandlas till vindhastighet V_{vind} i intervallet 0-32.4 m/s genom interpolering med följande formel:

$$V_{vind} = \frac{V_s - V_{min}}{V_{max} - V_{min}} * 32.4 \text{ [m/s]} \quad (2)$$

Där:

V_{vind} = Vindhastighet i m/s

V_s = 0.4 – 2V uppmätt utspänning från vindsensorn

V_{min} = 0.4V

V_{max} = 2V

2.3.4 Regnsensor

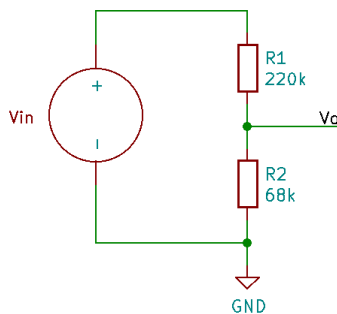
Regnsensorn YL-83 består av ett kretskort och en 5.4*4cm avläsningsplatta. Plattan har en spänning mellan 0-5V. När plattan utsätts för regn, kortsluts kretsen på plattan och spänningen blir 0V.

2.3.5 Strömmätare

ACS712 är en strömsensor som kan mäta strömmar i intervallet $\pm 20A$. Sensorn har en känslighet på 100mV/A, vilket innebär att sensorn ger 100mV utslag, för varje 1A som mäts. Sensorn ger ifrån sig mätvärdesbrus på $\pm 11mV$ vilket motsvarar ca $\pm 0.1A$ felmarginal som bör uppmärksammas vid användning av denna typ av sensor.

2.3.6 Spänningsmätare

För att mäta likspänning används spänningsdelare med två motstånd med kvoten 3/1. Det vill säga, R_1 är cirka 3 gånger större än R_2 , vilket ger ett mätintervall på 0-21V, som ger en utspänning V_o på 0-5V som Arduino kan läsa av. Se figur 3 för kretsschema.



Figur 3: Spänningsdelare för mätintervallet 0-21V

Formel:

$$V_o = V_{in} * \frac{R_2}{(R_1 + R_2)} \quad (3)$$

Där:

V_{in} = Mätspänning 0 – 21V

V_o = Utspänning 0 – 5V

$R_1 = 220k\Omega$

$R_2 = 68k\Omega$

2.4 Batterikapacitet

För att mäta batteriets kapacitet (SoC) används en metod där batteriets nuvarande spänning jämförs med maximala och minimala kapaciteten enligt formel (4). Batteriets kapacitet representeras av SoC; där 0% är helt urladdat batteri och 100% är fulladdat.

Formel:

$$SoC = \frac{V - V_{min}}{V_{max} - V_{min}} * 100 \quad (4)$$

Där:

SoC = Batteriets kapacitet i %

V = Batteriets uppmätta spänning

V_{min} = Batteriets spänning vid 0% kapacitet

V_{max} = Batteriets spänning vid 100% kapacitet

2.5 A/D-omvandlare

Majoriteten av sensorerna som används i arbetet ger en analog utsignal i intervallet 0-5V. För att kunna tolka och använda signalvärdet, måste den analoga signalen omvandlas till en digital signal. Enkortsdatorm Arduino Mega 2560 som används i projektet har en integrerad A/D-omvandlare med en upplösning på 10-bit ($2^{10}=1024$). Detta innebär att den analoga signalen på 0-5V omvandlas och representeras av en diskret digital signal i intervallet 0–1023. 0 motsvarar 0V och 1023 motsvarar 5V. Till exempel, en analog på 2.5V skulle motsvara en digital signal 512. Se beräkning (5) till (7).

$$\frac{\frac{A}{D} \text{ upplösning}}{\text{Referensspänning}} = \frac{\text{Digital signal}}{\text{Analog signal}} \quad (5)$$

$$\frac{1024}{5} = \frac{\text{Digital signal}}{2.5} \quad (6)$$

$$\text{Digital signal} = 2.5 * \frac{1024}{5} = 512 \quad (7)$$

Som formeln visar, är A/D-omvandlingen beroende av referensspänningen (5V för arduino). Därav är en stabil referensspänning kritisk för att få pålitliga mätvärden.

2.6 Spänningsregulator

För att reglera likspänning används en justerbar spänningsregulator av typen LM2596, för att omvandla hög spänning till låg spänning. Spänningsregulatorn kan omvandla 4-35V till 1.23-30V och leverera 3A (1.5A kontinuerligt) och har en verkningsgrad på cirka 90%.

2.7 Flask framework

Flask är ett Python-baserat ramverk för att skapa webbapplikationer på ett snabbt och enkelt sätt [5]. Se figur 4 för ett kodexempel på en webbapplikation som skapats med Flask.

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'
```

Figur 4: Exempel på en minimal flask webbapplikation skriven i Python3. programkoden skapar en hemsida som visar *"Hello, World!"*. (Flask User's Guide, 2020)

För att distribuera webbapplikationen, används Gunicorn3 som webbserver [6]. Webbservern hanterar datatrafiken mellan användaren som besöker hemsidan och webbapplikationen.

2.8 InfluxDB

InfluxDB är en databas optimerad för att lagra och hämta tidsserie-baserade data, vilket är passande för applikationer där data är tidsberoende, till exempel insamling av sensordata eller analys av realtidssystem.

2.9 Grafana

Grafana är ett öppet-källkodsverktyg som används tillsammans med valfri databas, för att visualisera data [7]. De erbjuder dynamiska instrumentpaneler, där användaren själv kan bestämma hur panelen ska se ut, vilken data som ska visualiseras och flertal olika graf-alternativ.

3 Metod

I följande avsnitt presenteras metoden samt de verktyg som använts för framtagning av prototypen som ska samla in, hantera och visualisera data relaterat till mikronät.

3.1 Förstudie

En förstudie genomfördes i syfte att skapa en övergripande uppfattning av projektets ämnesområde. I förstudien samlades information om solenergi, sensortyper, datahanteringsmetoder. Därtill undersöktes lämplig teknik för projektet samt vilken teknik som används idag och hur en används. Förstudien innefattade även analys av potentiella faktorer som kan påverka mikronätets energiproduktion, vilket stod till grund för val av hårdvarukomponenter och sensorer.

3.2 Kravspecifikation

Med en överblick av ämnet utfördes en kravspecifikation som baserades på krav och önskemål som förväntas i projektets slut, av både projektansvarige och uppdragsgivare.

3.3 Val av hårdvara och systemuppbyggnad

Utefter kravspecifikation och förstudie bestämdes vilka sensorer som ska användas för att samla in nödvändiga mätvärden. Därtill bestämdes även vilka slags enheter som krävs för att hantera sensorerna och bearbeta sensordata som samlas in.

Med en övergripande bild av ämnet och vilken hårdvara som krävdes, utvecklades en modell av prototypens systemuppbyggnad. Modellen visade hur systemets alla delar är sammankopplade och var sensorerna placerades.

3.4 Mikronätssystem

Under utvecklingsprocessen användes laborationsutrustning för att testa prototypens delfunktioner. Vid test av den slutgiltiga versionen av prototypen, togs ett testsystem fram, som agerade som mikronätsmodell.

3.5 Mjukvara

För att uppnå skalbarhet i mjukvaran användes C och objektorienterad C++ som programspråk för sensornoden. Varje sensor är ett eget objekt som är en instans av en klass. Klassen innehåller information och funktionalitet som definierar sensorn. På så sätt kan flera enskilda sensorer inom samma klass använda samma kod. Detta möjliggör skalbarhet i mjukvaran då man enkelt kan lägga till eller ta bort sensorer från mjukvaran. C/C++ är inte heller bundet till någon specifik hårdvara, vilket gör att hårdvaruval och sensorval inte begränsas.

Klasserna är vanligtvis uppdelade i två filer; header fil (.h) och implementeringsfil (.cpp). I headerfilen deklarerar klasser, variabler och funktioner, medan implementeringsfilen innehåller definitioner.

Python3 användes för att programmera mjukvaran till centralenheten (Raspberry Pi 3B+), som samlade in och behandlade data från sensornoden. Centralenhetens uppgift är att bearbeta och hantera sensordata innan den skickades vidare till en databas via Wi-Fi för lagring.

En databas lokaliserad på en server användes för att hantera insamlad data. En webbapplikation med ett grafiskt gränssnitt konstruerades för att visualisera data via internet.

4 Genomförande

I följande avsnitt presenteras genomförandet av prototypframtagningen.

4.1 Förstudie

För att bestämma vilka sensorer och hårdvara som krävs för att samla in data relaterat till energiproduktion gjordes en förstudie för att se vilka värden som är nödvändiga för att göra en analys av ett energiproducerande system.

För att beräkna den elektriska effekten P som genereras från solpanelen, lagras i batteriet och används av belastningen, krävs spänningsmätare och strömmätare. Se ekvation 4.1.1.

$$P = U * I \text{ [W]} \quad (4.1.1)$$

Enligt databladet för solcellerna (Sunpower C60 Bin H), som används i projektet, påverkas den genererade effekten beroende på temperaturen som den utsätts för. Normalt sätt testas solcellerna vid 25°C. För varje 1°C temperaturändring, minskar effekten med -0.32%. Mätning av temperaturen på solpanelen samt utomhustemperatur är därav ett rimligt val. Se bilaga 10 för solcellernas datablad.

Förundersökningen visade även att väderförhållanden kan ha en viss påverkan på systemets energiproduktion. Damm, sand och annan smuts som samlas på solpanelerna kan sänka verkningsgraden av solcellernas elproduktion, då de hindrar solenstrålningen att nå solcellerna [8]. Sensorer som mäter vindhastighet och regn, är mot denna bakgrund intressanta för projektet, då dessa två naturliga faktorer möjligt kan rengöra panelerna.

4.2 Kravspecifikation

Prototypen baserades på följande krav och önskemål utefter förstudien:

Tabell 1: Krav på hårdvara

Funktion	Krav	Önskemål	Vald
Mäta utomhustemperatur	X		X
Mäta luftfuktighet		X	X
Mäta vindhastighet	X		X
Mäta om det regnar eller inte		X	X
Mäta mängd regn		X	
Mäta ström från input, lagring och belastning	X		X
Mäta spänning från input, lagring och belastning	X		X
Mäta temperatur på solpanel och batteri	X		X
Lägga till och ta bort sensorer	X		X
Lägga till fler, eller ta bort sensorkluster		X	X

Tabell 2: Krav på mjukvara

Funktion	Krav	Önskemål	Vald
Möjlighet att konfigurera vilka sensorer som är aktiva på sensornoden	X		X
Mjukvarukompatibilitet för att ta bort och lägga till fler sensorer	X		X

Tabell 3: Krav på datahantering

Funktion	Krav	Önskemål	Vald
Lagra sensordata på en server	X		X
Visualisering av data via webbgränssnitt		X	X

4.3 Val av sensorer

Sensorer valdes utefter kravspecifikationen för att uppfylla systemets krav och funktioner. Sensorerna valdes främst med hänsyn till funktion kontra pris.

Motståndstermometer av typen PT1000 valdes som sensor för temperaturmätning av solpanel och batteri. Fördelen med sensorn är att den ger tillräckligt exakta mätvärden med 2-ledarkoppling jämt emot till exempel PT100 som har lägre nominellt motstånd och kräver 4-ledarkoppling för mer exakta mätvärden.

För att mäta spänning konstruerades spänningsdelare med hjälp av två motstånd för att klara av systemets spänningsintervall. Det är en simpel konstruktion, till ett lågt pris, vilket underlättar framtida ändringar om mätspannet skulle behövas ändras. Spänningen mäts över batteriet samt solpanelen. Genom att mäta spänningen över batteriet kan batteriernas kapacitet uppskattas med hjälp av potentialskillnaden mellan batteriets uppmätta kapacitet och kapaciteten vid fulladdning.

För att mäta strömmen vid batteri, solpanel och belastning användes tre ASC712 sensorer, vilka kan mäta strömmar i intervallet $\pm 20A$. Detta medför även att sensorn därtill kan avgöra strömmens riktning. Se bilaga 1 för placering av sensorerna.

DHT11 valdes som utomhustermometer och luftfuktighetsmätare. Sensorn är billig, lättillgänglig och har ett rimligt mätintervall på 0–50°C och 20–80% luftfuktighet, vilket är ett acceptabelt mätintervall för prototypen som tas fram.

Adafruits anemometer valdes för att mäta vindhastighet. Sensorn klarar av vindar upp till 32.4m/s och har en robust konstruktion. Sensorn placeras i närhet till solpanelen.

4.4 Systemöverblick

Ett systemdiagram som illustrerar projektets hårdvara togs fram. Se bilaga 1. Diagrammet beskriver de fyra huvudkomponenterna:

1. Testsystem (solpanel, batteri, MPPT)

2. Sensorkluster, inklusive sensorer
3. Centralenhet för datahantering (Raspberry Pi 3 B+)
4. Server (lagring av data samt distribution av webbapplikation)

Diagrammet visar även mätpunkterna av temperatur (°C), spänning (V) och ström (A) på solpanel, batteri och belastningen, samt kommunikationsmetoder som används (UART & WIFI).

5 Hårdvara

I följande kapitel beskrivs hårdvara och konstruktion av prototypen.

5.1 Testsystem

Testsystemet i projektet består av en 95W-solpanel som är baserad på monokristallina solceller (en typ av fotovoltaiska solceller), ett 12V-batteri för energilagring samt en MPPT-regulator. Syftet med regulatoren är att optimera den utgående effekten från solpanelen till batteri och belastning, samt skydda systemet mot omvänd strömriktning om spänningen på solpanelen är längre än batterispänningen.

Batteriet som användes är ett Li-ion batteri med 12V spänning (12.6V vid fulladdat batteri och 9V vid helt urladdat batteri).

För att undvika att solpanelens spänning faller under batteriets spänning, ställdes MPPT-enheten in på tröskelvärde 15.1V, vilket innebär att kretsen mellan solpanelen och batteriet bryts, om solpanelens spänning faller under tröskelvärde.

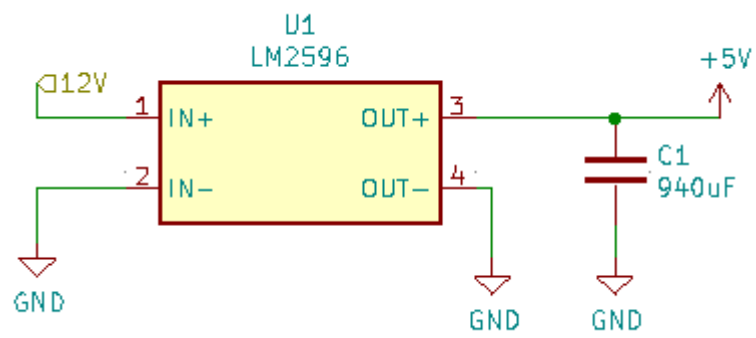
För att starta och stänga av systemet, monterades två strömbrytare vid batteri och solpanel. Se bilaga 11.

5.2 Sensornod

Enkortsdatorn Arduino Mega 2560 användes som sensornod eller ”nod” för att samla in data från sensorerna. Sensorerna som valdes är baserade på 5V spänning och ger antingen en analog eller digital utsignal, förutom vindsensorn, som behöver extern 7-24V strömförsörjning för att drivas. Arduinon är därför ett lämpligt alternativ för ändamålet då den har ett flertal in- och utportar som kan läsa och skriva både analoga och digitala signaler. Därtill kan arduinon driva alla sensorer utan strömförsörjning.

Testsystemet (solpanel och batteri) är baserat på 12V, och arduinon, sensorer och Raspberry Pi drivs på 5V. Därför användes spänningsregulatorer av typen LM2596 för att reglera spänningen från 12V till 5V.

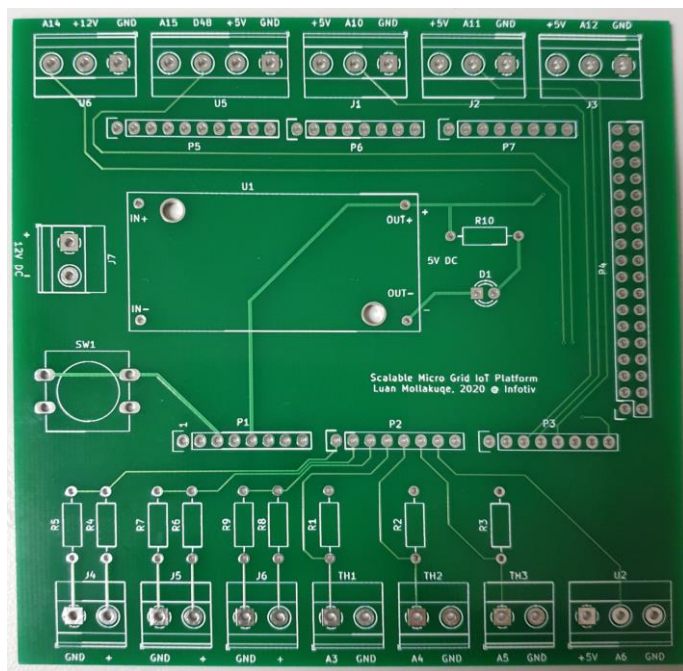
Majoriteten av sensorerna är analoga och därav beroende av en stabil spänningsförsörjning. Därför monterades två kondensatorer (totalt 940 μ F) på spänningsregulatorn för att stabilisera utspänningen från spänningsomvandlaren. se figur 5.



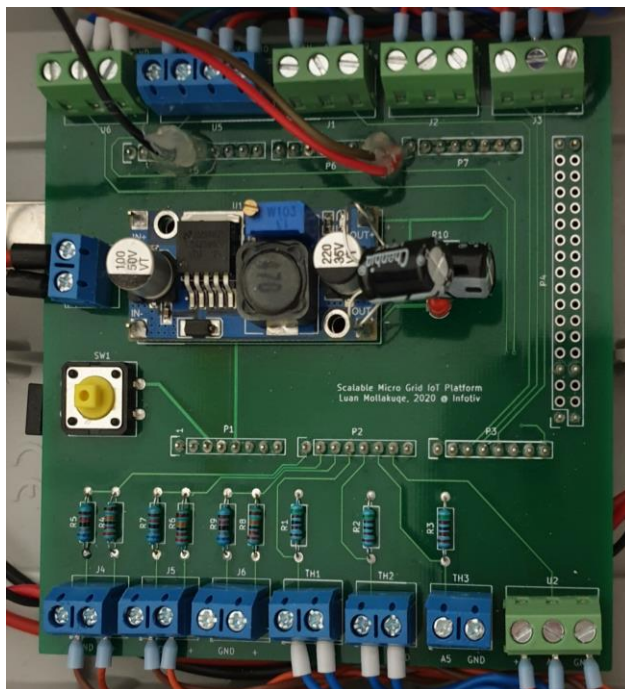
Figur 5: Kretsschema för spänningsomvandlare inklusive kondensator. Urklipp från bilaga 4.

5.2.1 Kretskort

När funktionalitet bekräftades för de enskilda sensorerna, utvecklades även ett kretskort som monteras ovanpå arduinon för att koppla sensorerna till övriga systemet. Som följd minimerades mängden kablar och användarvänligheten främjades med prototypen. Se figur 6 och 7 för det slutgiltiga kretskortet. Se bilaga 4 för sensorklustrets kretsschema.



Figur 6: Kretskort utan komponenter, konstruerad för att monteras ovanpå en Arduino Mega 2560



Figur 7: Kretskort inklusive komponenter

Sensorerna kopplades till respektive skruvplint på kretskortet och monterades därefter på önskad mätplats för att samla mätdata. Genom att använda skruvplintar på kretskortet,

som är internt kopplade till enskilda analoga och digitala portar på arduinon, kan sensorer bytas ut och ersättas med valfri sensor som är kompatibel med arduino.

Figur 7 visar även den yt-monterade spänningsregulatorn, som omvandlar inkommande spänningen på 12V till 5V, samt en gul reset-switch, som användes för att återställa arduinon vid behov.

Sensornoden konstruerades för att rymma tolv sensorer; tio portar används och två är lediga. Se bilaga 8 för detaljerad information kring vilka skruvplintar, portar, signaltyper och signalnamn i mjukvaran som sensorerna kopplades till, samt portar som används på centralenheten (Raspberry Pi).

5.2.2 Temperatur och luftfuktighet

Temperaturen mäts på tre olika platser; batteriet, solpanelen och utomhustemperaturen. För utomhustemperaturen användes en digital DHT11-sensor som kan läsa av både temperatur (0–50°C) med $\pm 2\%$ felmarginal & luftfuktighet (20–80%) med $\pm 5\%$ felmarginal. Sensorn placerades utanför sensorklustret för att mäta omgivningens temperatur och luftfuktighet.

För att mäta temperaturen på batteriet och solpanelerna användes PT1000-sensorer. På solpanelen monterades sensorn på baksidan av solpanelen för att mäta hur varm panelen blir när den utsätts för solljus. Den andra sensorn monterades på en av battericellerna.

5.2.3 Vindhastighet

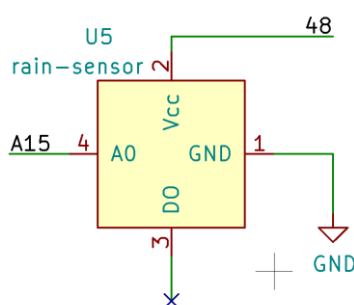
För att mäta vindhastighet användes en anemometer från Adafruit. Sensorn drivs på 7–24V, till skillnad från de övriga sensorerna som drivs på 5V, därav kopplades sensorn direkt till batteriet. Sensorn ger en analog utsignal på 0.4–2V, vilket motsvarar vindhastigheten på 0 m/s vid 0.4V och 32.4 m/s vid 2V. Sensorn placerades nära solpanelen i syfte att avläsa hur vinden påverkar solpanelen.

Omvandlingen från den analoga mät-signalen utfördes via mjukvaran i sensornoden. Se filen "Windsensor.cpp" och "Windsensor.h" under bilaga 9 för programkod.

5.2.4 Regnsensor

Regnsensorn placerades i närhet till solpanelen. När regnsensorn utsätts för regn kortsluts mätinstrumentet (avläsningsplattan), vilket ger en analog signal mellan 0-5V, motsvarande ”regn eller inte regn”.

Det finns en risk att avläsningsplattan rostar när ström går igenom avläsningsplattan samtidigt som den utsätts för vatten. Därav modifierades sensorn genom att koppla regnsensors in-spänning till en digital signalport (port 48) på arduinon. Med hjälp av mjukvara i arduinon kan den digitala porten aktiveras (5V) när sensorn avläses och avaktiveras (0V) när den inte används. På så sätt minimerades risken att sensors avläsningsplatta rostar. Se figur 8 för regnsensors kretsschema, samt bilaga 4 för sammankoppling till arduino.



Figur 8: Kretsschema för regnsensor. Urklipp från bilaga 4.

5.2.5 Spänningsmätare

Då arduinon enbart kan mäta 0-5V, konstruerades två spänningsdelare för att mäta vid solpanelens och batteriets spänningar som varierar mellan 0-17V. Spänningen vid batteriet används för att beräkna hur mycket kapacitet som finns kvar i batteriet (SoC), samt effektberäkningar för ingående och utgående effekt (beroende på om batteriet laddas upp eller laddas ur). Se beräkningsmetoden för SoC i kapitel 2.4.

Spänningsmätarna dimensionerades för att hantera spänningar upp till 21V. Den uppmätta spänningen på 0-21V omvandlas till 0-5V med hjälp av spänningsdelaren, så att arduinon kan läsa av signalen. Se bilaga 4 för kretsschema och kapitel 2.3.6 för teorin bakom spänningsmätare.

Dessa spänningsmätare kan dimensioneras om för att öka mätintervallet genom att ersätta motstånden med andra värden på kretskortet.

5.2.6 Strömmätare

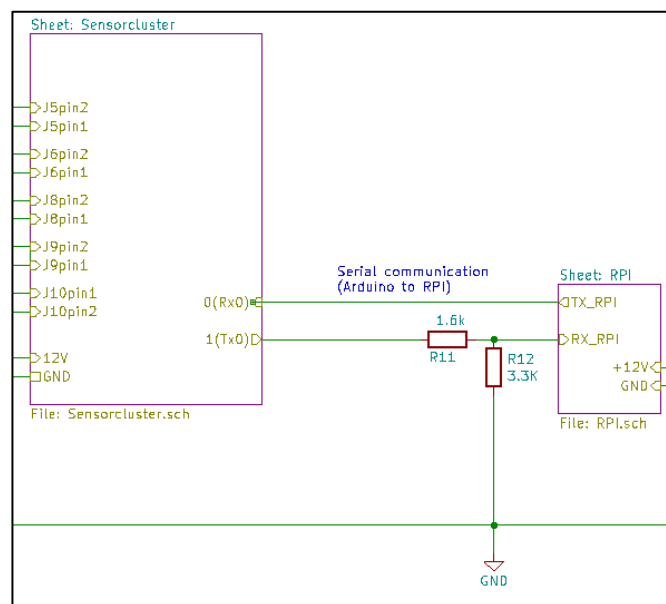
Strömmätarna (ASC712) är kapabla till att mäta $\pm 20A$ strömmar och placerades på tre punkter; solpanelen, batteriet och belastningen. Data som samlades in med dessa sensorer användes tillsammans med spänningsdelarna för att beräkna in- och utgående effekten från batteriet, utgående effekten från solpanelen samt belastningens effektkonsumtion.

Med kända effekter (baserat på ström- och spänningssensorerna) från hela systemet, utfördes energiberäkningar för att analysera in och utgående effekt, förluster och verkningsgrad i systemet.

5.3 Centralenhet

Utöver sensornoden användes ytterligare en enkortsdator (Raspberry Pi 3 B+) som centralenhet för att samla in och bearbeta all sensordata innan det skickas vidare till en server för lagring.

Sensornoden kopplades till centralenheten via seriekommunikation (UART). Då sensornodens seriekommunikations-portar är 5V och centralenhetens portar är 3.3V, användes en spänningsdelare för att sänka signalspänningen från arduinon till 3.3V. Se figur 9.



Figur 9: kretsschema för seriekommunikation mellan Raspberry Pi och Arduino. Urklipp från bilaga 2.

5.4 Mekanisk konstruktion

Den mekaniska konstruktionen delades in i två delar, sensorkluster (figur 10) och testlåda.

Sensorklustret innehåller sensornoden och centralenheten. Testlådan innehåller batteri, kopplingsplatta för att koppla samman batteri, solpanel, MPPT, ström- och spänningsmätare. Se bilaga 11 för testlådan och dess innehåll samt bilaga 12 för innehållet i sensorklustret.



Figur 10: Sensorkluster

För att underlätta kabelhanteringen, borrades tre hål i sensornoden för sensorernas kablage samt strömförsörjningens kablage (från batteriet i testlådan till sensornoden). Detta underlättar även framtida möjligheter att ersätta och byta ut sensorerna på ett enkelt sätt.

6 Mjukvara

I följande kapitel beskrivs mjukvaran för sensornoden, centralenheten och webbapplikationen.

6.1 Sensornod

Sensornodens programvara skrevs i C/C++. Enskilda klasser programmerades för varje sensortyp. Se bilaga 8 för fullständig programkod.

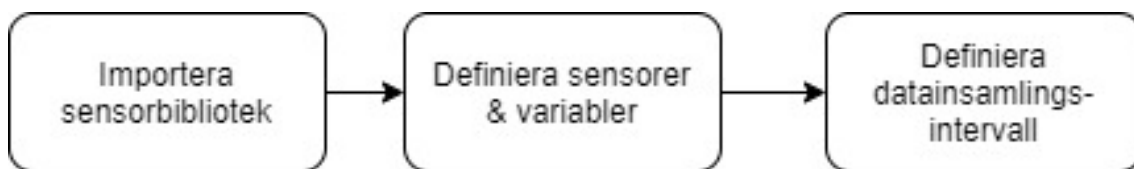
Programvaran är uppdelad i 3 delar;

1. Konfigurering
2. Huvudloop
3. Dataöverföring

6.1.1 Konfigurering

Programvaran skrevs på så sätt att användaren kan konfigurera vilka sensorer som ska vara verksamma när sensornoden är aktiv.

Sensorbiblioteken för varje sensortyp importerades. Dessa bibliotek användes därefter för att definiera varje enskild sensor och dess nödvändiga variabler för att läsa av sensorerna. Därefter definieras datainsamlingsintervallet för hur frekvent data ska samlas in från sensornoden och skickas till centralenheten. Se figur 11 för diagram över konfigureringsmetoden. Se bilaga 9 för sensornodens fullständiga programvara.



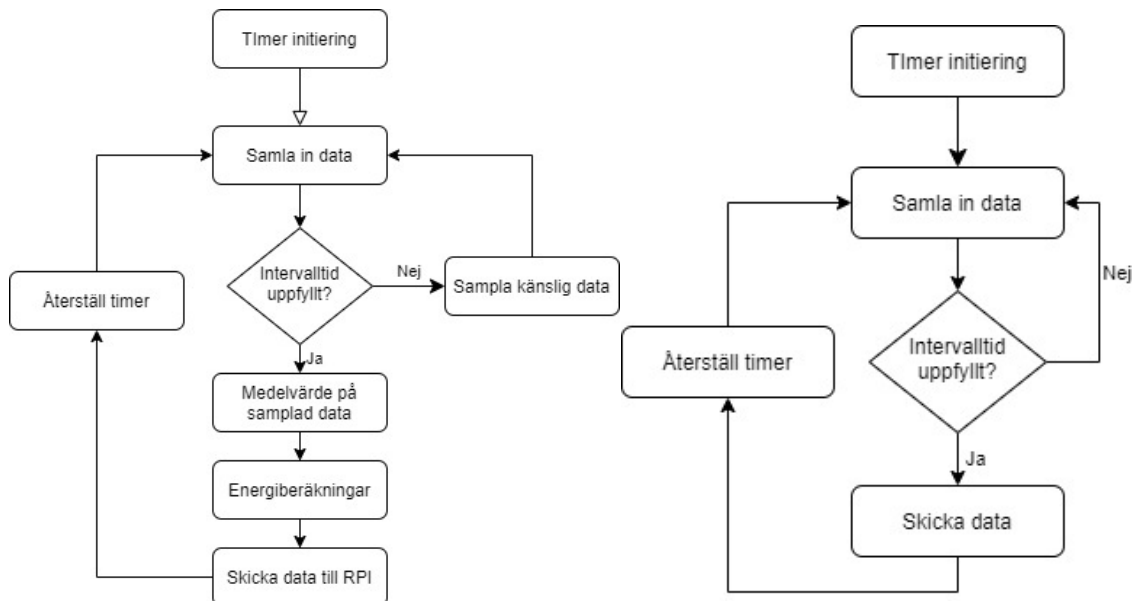
Figur 11: Diagram för konfigurering av sensornodens mjukvara

6.1.2 Huvudprogram

Huvudprogrammet delades in i två programslingor; en för känslig data och en för icke-känslig data. Båda slingorna samlar in och skickar vidare datan till centralenheten via UART om tidsintervallet är uppfyllt. Se figur 12 för programslingorna.

Känslig data (ström, spänning och effekt) samlas in under det inställda intervallet på tre sekunder. Därefter skapas ett medelvärde av datan som skickas till centralenheten lagring i databasen.

Till skillnad från den känsliga datan, så skapas inget medelvärde för den icke-känsliga datan (temperaturer, vindhastighet, regn, luftfuktighet). Denna data är inte lika kritisk, därför räcker det med en datapunkt per tidsintervall för den icke-känsliga datan.



Figur 12: Diagram för huvudprogrammets två programslingor (känslig data till vänster, icke känslig data till höger).

6.1.3 Filtrering

Under utvecklingsprocessen uppstod kraftiga fluktuationer i sensorernas mätvärden. Detta löstes genom att använda kondensatorer vid inspänningen till arduinon för att stabilisera spänningen som alla analoga mätvärden baserades på, som nämndes i kapitel 5.2.

A/D-omvandlaren hinner inte ”vila” innan den ska läsa av nästa analog port (nästa sensor), därför utfördes avläsningar som inte sparas, innan varje avläsning från sensorerna. Detta gör att A/D-omvandlaren hinner återställas och samla in korrekt mätdata, då de första två mätvärdena filtreras bort. Därefter skapas ett medelvärde av tio samplings från den analoga porten. Se figur 13.

```

void WindSensor::getRawData(){
    analogRead(m_pin); // dummy read
    analogRead(m_pin); // dummy read
    m_sensorRawData = data_sampling(analogRead(m_pin), 10);
}

```

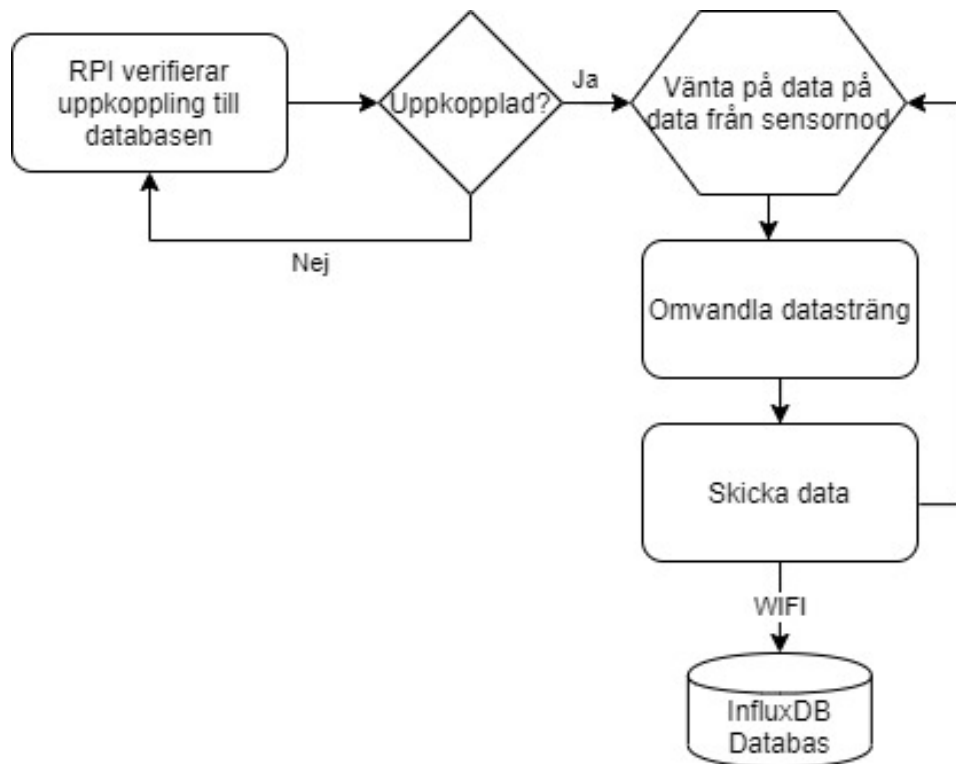
Figur 13: Exempel på vindsensors avläsningsfunktion med två avläsningar som inte sparades, följt data-sampling. Urklipp från "Windsensor.cpp" från bilaga 9.

6.2 Centralenhet

En centralenhet baserad på Raspberry Pi 3 B+ (RPI) användes i syfte att behandla sensornodens insamlade data och öppna möjligheten att lägga till fler sensornoder i systemet. RPI agerade mellanhand för sensornoden som samlar in mätdata och databasen där insamlad data lagras.

Mjukvaran för centralenheten konstruerades på så sätt att den alltid läser av serie-porten som är kopplad till sensornoden. Om sensordata finns att hämta från sensornoden, så omvandlas datan enligt databasens protokoll och skickas vidare till databasen för lagring.

Se figur 15 för flödesschema, samt kapitel 6.3 och 6.4 för ingående beskrivning av dataöverföringen.



Figur 14: flödesdiagram för centralenhetens mjukvara

Felmeddelande sparas i en logg, lokalt på centralenheten, i det fall det skulle ske något problem med databehandlingen i centralenheten eller uppkoppling till databasen.

6.3 Databas

För att lagra insamlad sensordata, användes InfluxDB som databas. Databasen installerades och konfigurerades på en server som var lokaliserad hos Infotiv. För att koppla centralenheten till databasen, användes följande kod i centralenhetens programvara:

```
# Influx Config
client = InfluxDBClient(host='10.0.4.110', port=8086, retries=0)
client.switch_database('microgrid_test')
```

Figur 15: Konfigurering av InfluxDB på RPI. urklipp från "read_data.py" bilaga 5

Den första raden kod i figur 16, ser till att InfluxDB klienten i centralenheten kopplas till servern där databasen är lokaliserad. Den andra raden definierade vilken databas data ska förvaras på, i detta fall "microgrid_test", vilket var databasen som användes för att lagra data under test och kalibrering av systemet.

6.4 Dataöverföring

Som tidigare förklarat i kapitel 5.3 kommunicerar sensornoden och centralenheten med seriell kommunikation (UART). Sensordata skickas i form av textsträngar enligt följande struktur, "Plats, sensornamn, sensorvärde, prefix", där varje värde separeras med ett komma-tecken. Se figur 16.

```
void serialPrintData(){  
    /*print sensitive data  
    protocol: Location, sensor name, value, prefix */  
    SerialProtocol(LOCATION, "SPV", solarpanelvoltage, "V");  
    SerialProtocol(LOCATION, "BV", batteryvoltage, "V");  
}
```

Figur 16: Överföringsfunktion för solpanel- och batterispänning. Urklipp från "main.ino", bilaga 9.

Textsträngen omvandlades därefter i centralenheten enligt databasens protokollstruktur innan datan skickas till databasen för lagring via WIFI. Se bilaga 5.

6.5 Visualisering

För att visualisera insamlad data byggdes en webbapplikation baserat på ramverket Flask, med integrerad Grafana, för att visualisera data i form av grafer. Unicorn3 användes för att distribuera och komma åt webbapplikationen utanför det lokala nätverket. Se <https://microgrid.infotivlab.se> för slutgiltiga webbapplikationen.

För att distribuera applikationen kördes följande kod på servern där webbapplikationen är lokaliserad:

```
gunicorn3 -w 4 -b 0.0.0.0:5043 flask_app:app &
```

Figur 17: skript för att distribuera webbapplikationen

Koden kör webbapplikationen "flask_app" som ligger i mappen "app", med fyra arbetande processer (-w 4) och distribuerar applikationen på den lokala adressen 0.0.0.0:5043. Adressen till webbapplikationen portades därefter till microgrid.infotivlab.se, som är en publik adress tillgänglig för vem som helst via internet.

Om servern startas om för underhåll stängs webbapplikationen ner. För att lösa detta skapades även ett autoupstart-skript som startar webbapplikationen varje gång servern startas om. Se "startscript.sh" under bilaga 9.

7 Test och Kalibrering

7.1 Kalibrering

Sensorerna testades enskilt för att kontrollera att de mäter korrekta värden. De sensorer som går att kalibrera, kalibrerades med hjälp av konfiguration av mjukvaran.

DHT11-sensorn som mäter temperatur och luftfuktighet testades först i inomhusmiljö genom att jämföra med flertal andra termometrar. Temperaturen stämde överens med de termometrarna som det jämfördes med och höll sig inom sensorns felmarginaler på 2%.

PT1000-sensorerna kalibrerades på samma sätt som DHT11-sensorn, genom att jämföra med andra termometrar. Fel marginalerna korrigerades genom att ändra mjukvaran. Varje grad Celsius motsvaras av 4Ω skillnad i motstånd i PT1000-sensorn. Därför adderades den skillnaden i mjukvaran för att beräkningarna skulle bli korrekta.

Strömmätaren (ASC712) ställdes in på 100mV/A enligt databladet, då sensorn är kalibrerad från fabrik för 20A. Sensorn testades genom att mata olika strömmar genom sensorn med hjälp av ett externt reglerbart laborationsnätaggregat. Uppmätta sensorvärden stämde överens med den matade strömmen från nätaggregatet, men fluktuerade med $\pm 0.15A$. Fel marginalen enligt databladet var $\pm 0.11mV$, vilket motsvarar ca 0.11A efter omvandling.

Vindsensorn är kalibrerad från fabrik. Funktionaliteten verifierades genom att snurra manuellt på vindsnurran samt test i utomhusmiljö. Mätvärdena från utomhusmiljön motsvarade den uppskattade vindhastigheten som uppgavs av SMHI.se under testtillfället.

Regnsensorn funktionalitet testades genom att läsa av sensorn när den är torr samt när den är toppad i ett vattenglas, där avläsningsplattan blir kortsluten. Regnsensorn fungerade som väntat.

7.2 Test i kontrollerad miljö

Efter individuella tester av sensorerna, testades hela systemet i en simulerad miljö, där ett spänningsaggregat användes för att simulera solpanelen. Detta för att underlätta test av hela systemets funktionalitet och trovärdighet.

Genom att mata systemet med en känd spänning och ström, kan systemets trovärdighet utvärderas. Inomhustemperaturen i den kontrollerade miljön var ca 21.5–23°C.

Tröskelvärdet på MPPT-regulatorn är inställd på 15.1V, vilket innebär att kopplingen mellan solpanelen och batteriet bryts om spänningen är under 15.1V. Om spänningen går över 15.1V är de sammankopplade och batteriet kan laddas upp av solpanelen.

Under testtillfället i en kontrollerad miljö påverkas inte temperatur, vindhastighet, regn och luftfuktighet. Därav analyserades främst ström, spänning och SoC under testet. Under de olika testfallen testas olika nivåer av ström och spänning från solpanelen (reglerbara nätaggregatet) för att se hur effekten i systemet påverkas av olika effektlägen.

Notera att ”spänningskälla” motsvarar en simulation av en solpanel i detta test.

Testfall:

- #1: Spänningskälla avstängd.
- #2: spänningskälla på, men spänning under MPPT-regulatorns tröskelvärde.
- #3: Spänningskälla 16V, 0.5A.
- #4: Spänningskälla 16V, 1A.
- #5: Spänningskälla 16V, 2A.
- #6: Spänningskälla 16V, 3A.

Tabell 4: Testfall under kontrollerad miljö

Testfall	Spänningskälla (W)	Batteri (W)	Belastning (W)	Förluster (W)	Kommentar
#1	0	4	2	40–50%	Spänningskälla avstängd. Systemet drivs endast på batteriet.
#2	0	4	2	40–50%	Spänning under gränsvärdet för MPPT tillfördes. Ingen effekt går igenom MPPT.
#3	8	2	2	40–50%	Spänning över gränsvärdet tillfördes (16V). Batteriet börjar laddas upp med 0.5A
#4	15	10	2	24%	Solpanelens ström ökades till 1A.
#5	30	24	2	13%	Solpanelens ström ökades till 2A.
#6	45	38	2	11%	Solpanelens ström ökades till 3A.

Testfallen visade att systemet fungerar som väntat. Ingen effekt passerar genom MPPT-regulatorn om solpanelens spänning är under regulatorns tröskelvärde på 15.1V. När solpanelens spänning går över 15.1V (till exempel testfall #3 där spänningen är inställd på 16V), så passerar effekten från solpanelen till batteriet. Systemet fungerar som önskat. Kraftiga förluster uppstod under test 1, 2 och 3. Detta diskuteras närmare i kapitel 8, resultat.

SoC visade att batterikapaciteten ökade desto högre effekten från solpanelen var. Detta diskuteras närmre i kapitel 9.1, begränsningar och problem.

7.3 Test i utomhusmiljö

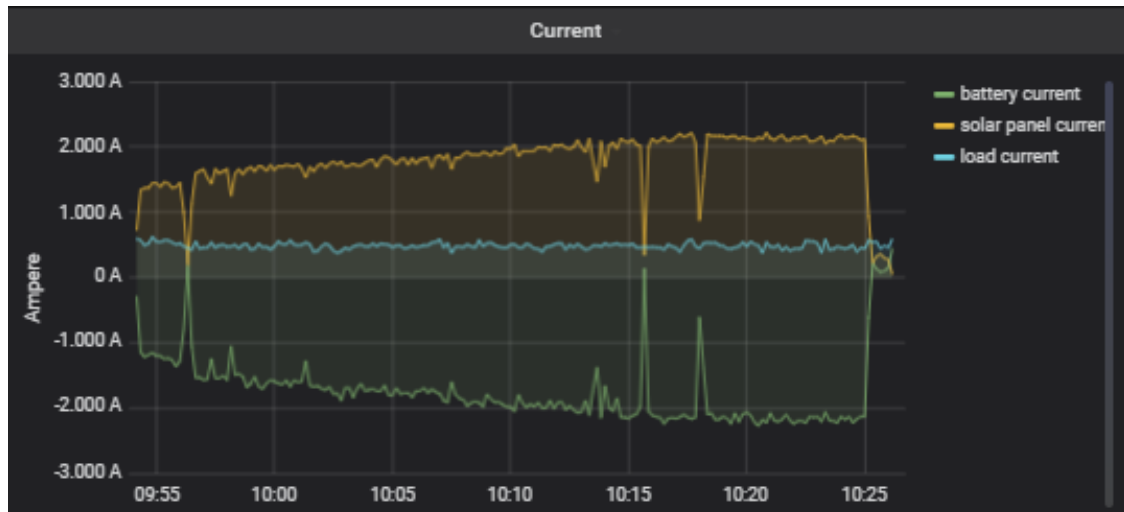
Det andra testet utfördes utomhus under klara väderförhållanden utan moln den 24 februari 2020 vid domkyrkan, Göteborg. Se bilaga 7 för resultatet och <https://microgrid.infotivlab.se/outdoortest> för en interaktiv hemsida för testdata.



Figur 18: Testrigg inklusive solpanel och batteri och sensorkluster vid Domkyrkan, Göteborg.

Testet visade en utomhustemperatur mellan 14–16°C, batteritemperatur på 25°C samt solpanelstemperatur på ca 15–20°C. Dessa mätvärden uppfattades som rimliga med tanke på att batteriet och solpanelstemperaturen blev varmare än utomhustemperaturen.

Solpanelen genererade runt 1.5-2A, vilket motsvarar cirka 22-30W med solpanelsspänningen på 15.2V. Runt klockan 10:15 visades även en kraftig spik i den genererade strömmen från panelen, vilket orsakades av att solpanelen hamnade i skugga. Se figur 18. Batteriet laddades upp från 11.7V till 11.9V under tiden som systemet kördes. Systemet samlade in data och fungerade som tänkt.



Figur 19: Graf över solpanel, batteri och belastningens ström som genererades/konsumerades under utomhustestet

8 Resultat

Resultatet av arbetet blev en prototyp bestående av sensorkluster inklusive 10 sensorer (12 möjliga) och en centralenhet som är uppkopplad via WIFI till en server där insamlad data lagras i en databas. Servern driver även en webbapplikation för att visualisera insamlad data från sensornoden. Alla delmål av projektet som definierades i kapitel 1.4 uppfylldes.

8.1 Skalbarhet

Resultatet av skalbarheten i projektet definieras av följande punkter:

- Möjligheten att lägga till, ta bort, ersätta sensorer i sensorklustret.
- Mjukvarans användarvänlighet.
- Möjligheten att lägga till fler sensornoder kopplat till centralenheten.

Genom att skruvplintar användes på sensornodens kretskort, är det enkelt att ta ersätta eller lägga till sensorer ur ett hårdvaruperspektiv. Mjukvaran programmerades objektorienterat, vilket främjade enkelheten att korrigera och lägga till nya rader kod till den befintliga koden.

Centralenheten är förberedd för att hantera flera sensornoder genom att enbart lägga till en avläsningsport för den nya sensornoden i centralenhetens mjukvara.

Databasen hanterar och sorterar all data enligt dataprotokollet för InfluxDB. Databasen begränsar inte heller hur många olika sensornoder som kan läggas till, så länge centralenheten skickar vidare data till databasen enligt korrekt dataprotokoll.

8.2 Tester

Testerna för att verifiera systemets funktionalitet och sensorernas trovärdighet visade delade resultat.

Sensornod, centralenhet, webbapplikation och databas fungerade som väntat. Data kunde samlas in från alla sensorer, behandlas i centralenheten, skickas vidare till databasen och därefter visualiseras med hjälp av Grafana i webbapplikationen.

Testerna visade dock fluktueringar i mätdata som samlades in. Testfallen i inomhusmiljö visade 40–50% effektförluster vid längre effektmätningar och 11% förluster vid större effekter. PT1000-sensorerna för temperaturmätning av batteri och solpanel, fluktuerade med ca 1–2°C. Dessa problem kommer behandlas närmare i kapitel 9. Se bilaga 6 för fullständig testdata.

9 Diskussion

I följande kapitel kommer projektets frågeställningar besvaras, samt gå igenom problem som uppstod under projektet och förslag på vidareutveckling av prototypen.

- Vilken data är relevant att samla in med hjälp av prototypen för framtida forskning och utveckling inom mikronät?

Undersökningen som gjordes i början av projektet visade att de relevanta mätvärdena är främst systemets effektlöden och detta för att avgöra hur mycket elektricitet som produceras, lagras och konsumeras. Insamling av spänning- och strömdata var mot denna bakgrund relevant. Därtill visade studien att solpaneler påverkas av väderförhållanden i form av vind, regn, temperatur; vilket gör att insamling av data relaterat till väderförhållanden är relevant.

- Hur kan insamlad data användas?

Datainsamling är relevant vid både privat bruk, där användaren har mindre solpanelssystem på taket, samt för större mikronätssystem som driver flertal hushåll. Det är även relevant vid forskningssyfte för att analysera hur väderförhållanden påverkar elproduktionen hos mikronät baserade på solenergi.

Data som samlas in kan användas för att bevaka mikronätets elproduktion under både korta och långa tidsintervall. Genom att analysera avvikelser i insamlad data, kan även potentiella felkällor och skador hittas om systemet skulle gå sönder, till exempel sprickor i solpanelerna som påverkar elproduktionen.

Genom att samla in data relaterat till elproduktionen samt väderförhållanden, kan analyser utföras för att se systemet presterar under längre perioder av till exempel torrt väder, där smutsbildning kan ske på panelerna. Därutöver se hur systemet presterar efter regnfall och kraftiga vindar. Analyser kan även utföras för att avgöra den verkliga lönsamheten av solpanelssystem på olika geografiska platser där väderförhållanden skiljer sig åt.

- Vilka delar av systemet kan utvecklas skalbart för att i framtiden bredda användandet av systemet?

Skalbarheten i projektet definieras av två saker; möjligheten att ersätta, lägga till och ta bort sensorer från sensornoden, samt möjligheten att koppla upp flera sensornoder till centralenheten.

Sensornoderna konstruerades på så sätt att sensorerna kan bytas ut, vilket gör att varje nod kan anpassas med olika sensorer beroende på behov. På så sätt är sensornoden inte låst till att användas till specifikt solpanelssystem, utan kan tillämpas på till exempel mindre vind- eller vattenkraftverk.

Genom att sensornoden och centralenheten separerades från varandra, öppnades möjligheten att bredda plattformens användningsområde, genom att flertal olika

sensornoder med olika typer av sensorer kopplas upp till samma centralenhet. Till exempel kan en sensornod vara enbart dedikerad till att samla in väderdata och en annan nod för energidata från mikronätet.

9.1 Begränsningar och problem

9.1.1 Temperaturmätning

Varje grad Celsius motsvaras av 4Ω ändring i PT1000-sensors motstånd, det vill säga att motståndet i sensorn är 1080Ω vid 20°C . Det uppmätta motståndet fluktuerade ca $4\text{--}7\Omega$, vilket motsvarar ett fel på cirka $1\text{--}2^\circ\text{C}$ i mätvärdena. Detta beror på störningar som uppstår vid avläsning av sensorn och A/D-omvandlingen av signalen. Detta hade möjligtvis kunnat lösas genom att montera en kondensator (till exempel $0.1\mu\text{F}$) mellan den analoga porten på sensornoden och GND för att minimera störningarna.

9.1.2 Effektaffvikelser

Under testet i inomhusmiljö visades kraftiga effektaffvikelser. Detta beror på följande anledningar:

- Effektförluster i form av värme från spänningsomvandlingen i MPPT-regulatorn mellan solpanel och batteriet. Enligt databladet för regulatorn är verkningsgraden 96%, vilket motsvarar en effektförlust på 4%.
- Effektförluster i form av värme från spänningsomvandlingen från 12-5V från batteriet till belastningen. Både sensornoden och centralenheten hade en varsin spänningsomvandlare.
- Strömmätarna har enligt databladet störningar på $\pm 0.1\text{A}$, vilket påverkar effektberäkningarna. Vid 0.5A belastning motsvarar strömmätarnas störningar ca 20%.

9.1.3 Mätning av batterikapacitet

Metoden som användes för att beräkna batteriets kapacitet (SoC) visade sig vara missledande när batteriet utsätts för en belastning eller när det laddas upp. Spänningsmetoden som användes är mer rimlig att använda om batteriet inte utsätts för någon belastning. För att uppnå en mer trovärdig SoC medan batteriet är under belastning, bör även strömladdningen per tidsenhet tas i åtanke och jämföras med effektkurvan för batteritypen som används.

9.2 Förslag på vidareutveckling

9.2.1 Val av sensorer

DHT11-sensorn är inte den mest optimala sensorn för att mäta utomhustemperatur och luftfuktighet. Därför bör sensorn bytas ut med till exempel DHT22, som har lite bättre mätintervall för både temperatur och luftfuktighet.

9.2.2 Dataöverföringsteknik

I projektet är sensornoden och centralenheten kopplade via seriekommunikation (UART), vilket inte är det mest optimala kommunikationssättet om systemet ska utvidgas med flera sensornoder i framtiden. Centralenheten som baserades på RPI 3 B+ har redan en integrerad WIFI krets, men det har inte arduinon på sensornoden. Därav är det bättre att montera en WIFI modul på framtida sensornoder som läggs till, för att skicka över insamlad data till centralenheten via trådlös kommunikation.

Inga större ändringar behöver utföras för att möjliggöra WIFI-kommunikation. Det enda som krävs är ändring i mjukvaran för centralenheten, för att läsa av inkommande data från WIFI istället för UART. Överföring av data från centralenheten till databasen på servern sker fortfarande på samma sätt.

9.3 Slutsats

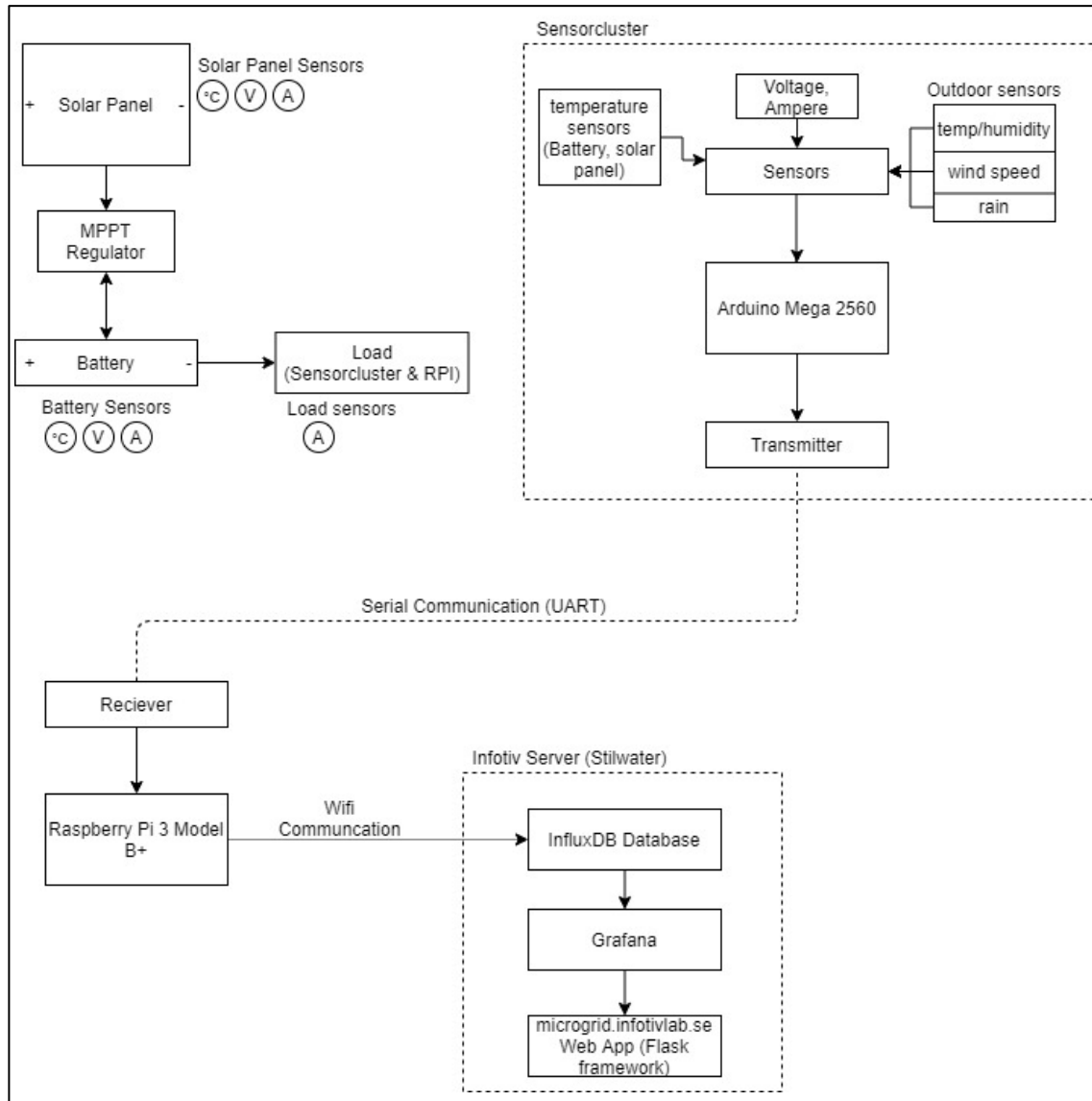
Projektets mål, med att skapa en skalbar och uppkopplad plattform för att samla in data relaterat till mikronäts energiproduktion samt väderförhållanden, uppfylldes. Genom att hela plattformen konstruerades på ett skalbart sätt, kan även problemen med mätvärdesavvikelserna lösas genom att ersätta de sämre presterande sensorerna med mer högpresterande sensorer.

Referenser

- [1] Vinnova, "Solenergi - En fördjupningsstudie av solenergi," *Hetast på Marknaden*, vol. 06, 2009.
- [2] G. M. o. I. Kaizuka, "Trends in Photovoltaic Applications," The IEA Photovoltaic Power Systems Programme (IEA PVPS), 2019.
- [3] D. Henning, "Energin påverkar miljön," Naturvårdsverket, 09 Augusti 2019. [Online]. Available: <https://www.naturvardsverket.se/Miljoarbete-i-samhallet/Miljoarbete-i-Sverige/Uppdelat-efter-omrade/Energi/Energin-paverkar-miljon/>. [Använd 18 Mars 2020].
- [4] Lopez Research LLC, "An Introduction to the Internet of Things (IoT)," Lopez Research LLC, San Fransisco, 2013.
- [5] "Flask User's Guide," 17 Mars 2020. [Online]. Available: <https://flask.palletsprojects.com/en/1.1.x/>. [Använd 17 Mars 2020].
- [6] B. Chesneau, "Gunicorn documentation version 20.0.4," 26 November 2019. [Online]. Available: <https://docs.gunicorn.org/en/stable/>. [Använd 18 Mars 2020].
- [7] "Grafana," Grafana Labs, 17 Mars 2020. [Online]. Available: <https://grafana.com/grafana/>. [Använd 17 Mars 2020].
- [8] J. J. F. Mejia, "The Effect of Dust on Solar Photovoltaic Systems," *Energy Procedia*, vol. 49, pp. 2370-2376, 2014.

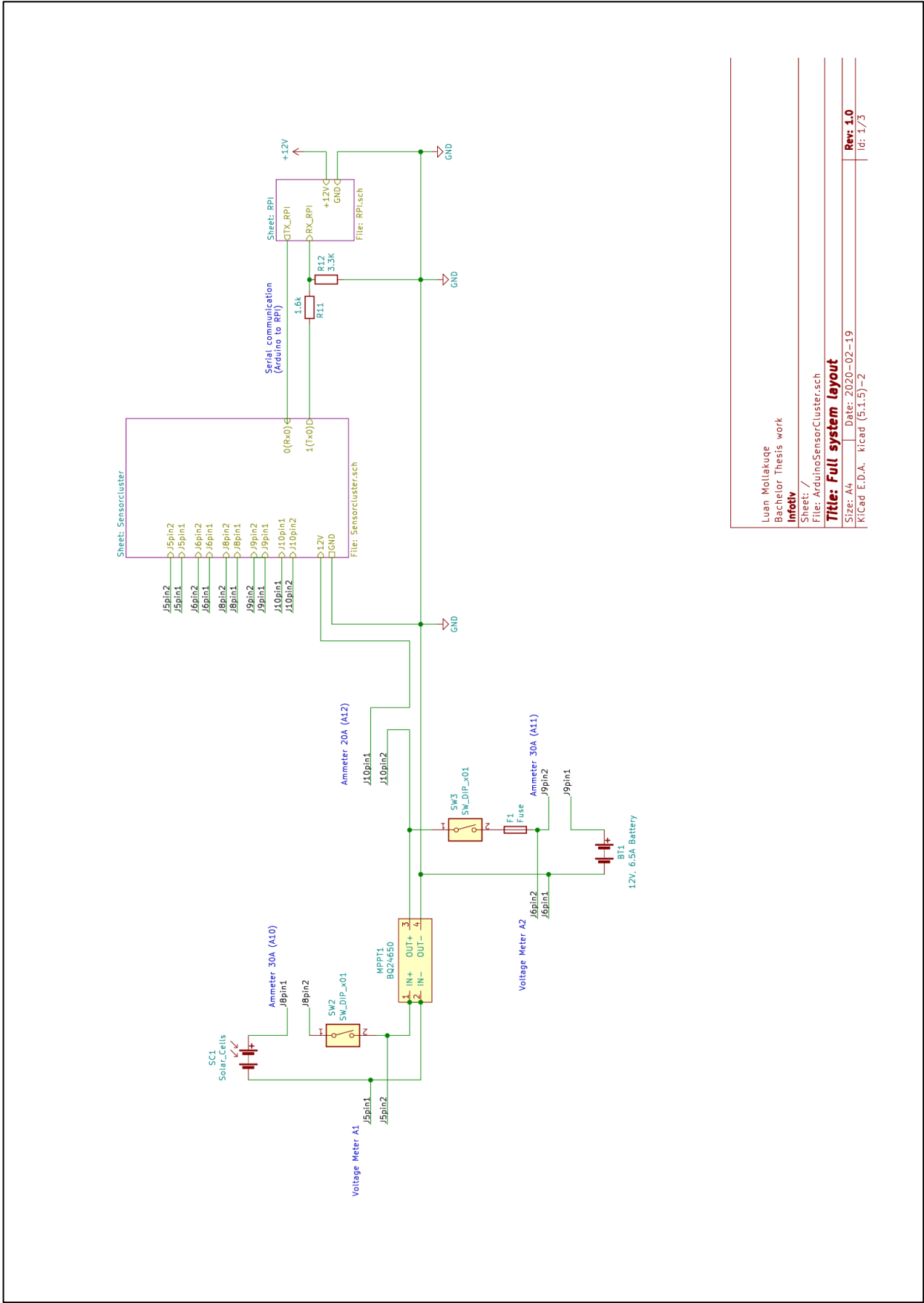
Bilagor

Bilaga 1: Systemdiagram



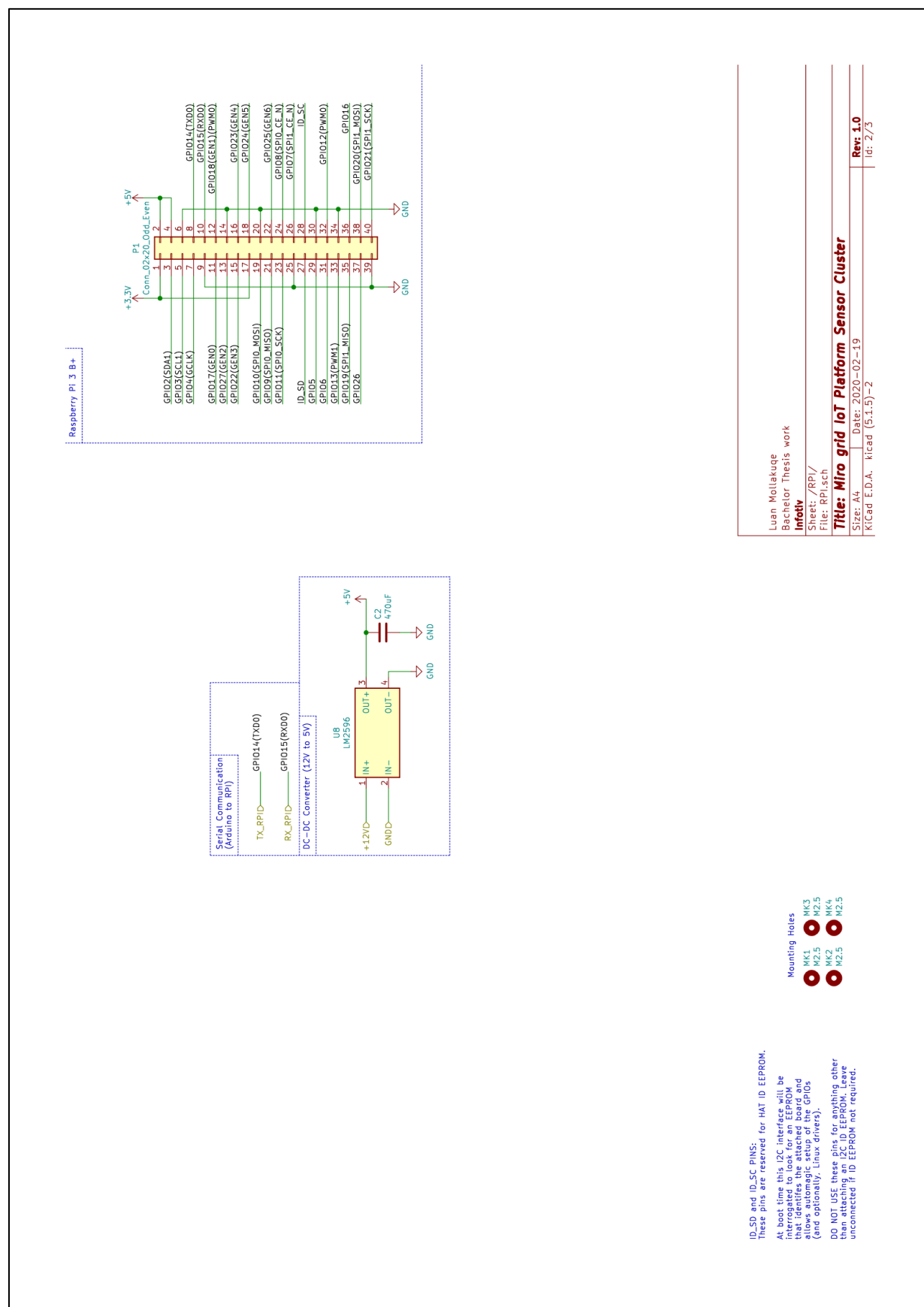
Figur 20 Systemöverblick av hela projektets delsystem

Bilaga 2: Kretsschema för hela systemet



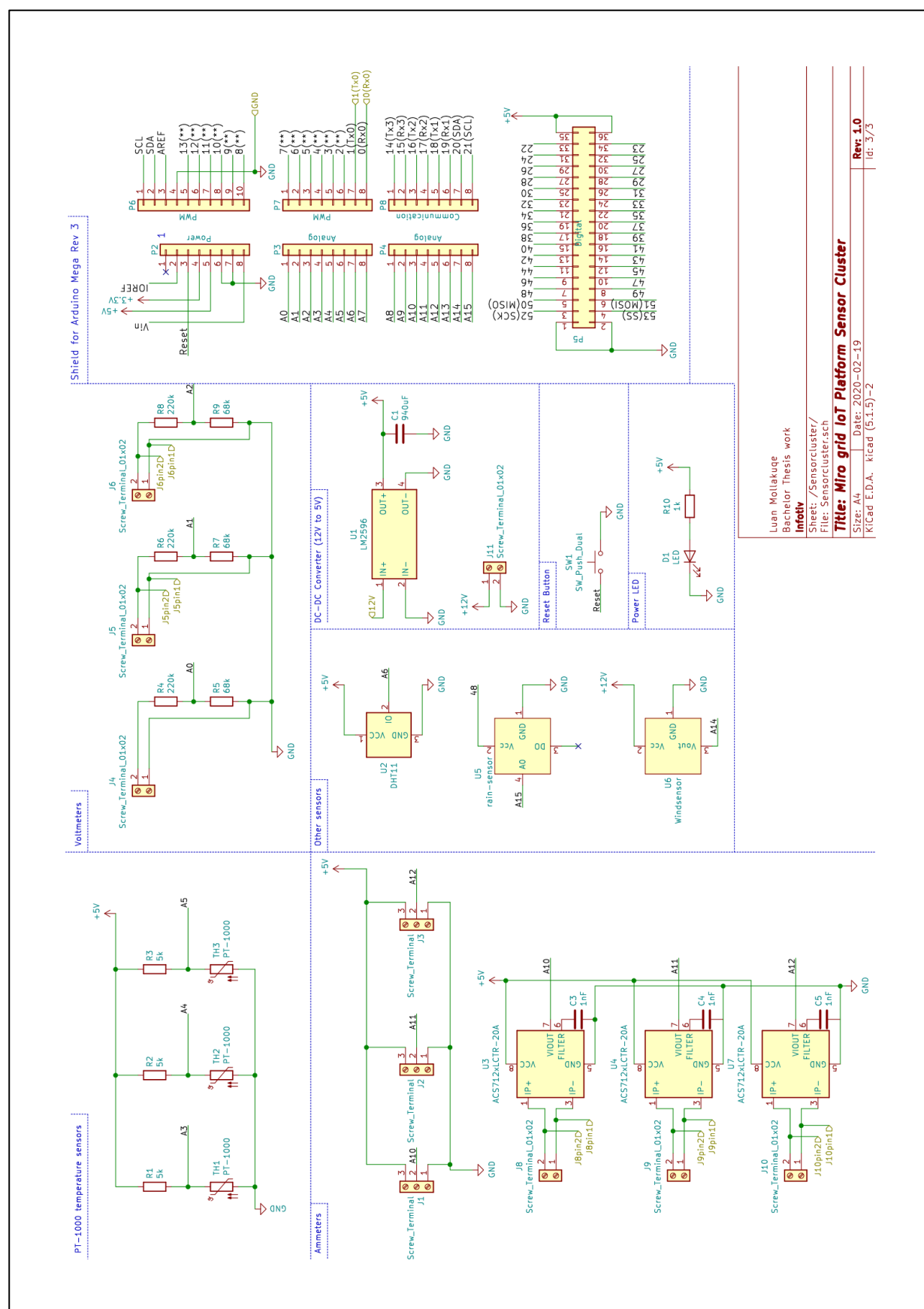
Figur 21: Kretsschema för hela systemet

Bilaga 3: Kretsschema för centralenheten



Figur 22: Kretsschema för Raspberry Pi 3 B+

Bilaga 4: Kretsschema för sensorklustret



Figur 23: Kretsschema för Sensorkluster baserat på Arduino

Bilaga 5: Centralenhetens mjukvara

```
#!/usr/bin/env python3
import os
import time
import serial
from influxdb import InfluxDBClient
import time

# Influx Config
client = InfluxDBClient(host='10.0.4.110', port=8086, retries=0) # retries = 0, try to connect until Success
client.switch_database('microgrid_test')

measurement = "sensorcluster"

def parse_data(line):
    location, sensortype, value, prefix = line.strip().split(",") # split at "," and strips \n
    value = float(value)
    data = [
        {
            "measurement": measurement,
            "tags":{
                "location": location,
                "sensortype": sensortype,
            },
            # "time": timestamp,
            "fields":{
                "sensorvalue": value,
                "prefix": prefix
            }
        }
    ]
    return data

# init serial
ser = serial.Serial('/dev/ttyS0', 115200)
time.sleep(3)

# Main Program
while 1:
    if ser.in_waiting > 0:
        line = ser.readline().decode() # decodes bytes to str
        data = parse_data(line) # parses string to line_protocol for influxDB
        # print(data)
        client.write_points(data)
```

Figur 24: mjukvaran i Raspberry Pi, read_data.py

Bilaga 6: Testdata från inomhustest



Figur 25: Data från test under kontrollerad miljö inomhus. hämtat från <https://microgrid.infotivlab.se/live>

Bilaga 7: Testdata från utomhustest



Figur 26: Testdata från utomhustest vid Domkyrkan, Göteborg. hämtat från <https://microgrid.infotivlab.se/outdoortest>

Bilaga 8: Portbeskrivning

Tabell 5: Tabell över vilka skruvplintar, portar, signaltyp, signalnamn (i mjukvaran) som sensorerna kopplades till.

Kontakt/ skruvplint (PCB)	PIN	Signal- typ	In/ut- signal	Signalnamn (I mjukvaran)	Spänning (V)	Spännings- intervall (V)	kommentar
J4	A0	Analog	In	solarPanelVoltage	5	0-5	Voltmeter
J5	A1	Analog	In	batteryVoltage	5	0-5	Voltmeter
J6	A2	Analog	In	Not used	5	0-5	Voltmeter
TH1	A3	Analog	In	batteryTemp	5	0-5	PT1000
TH2	A4	Analog	In	solarPanelTemp	5	0-5	PT1000
TH3	A5	Analog	In	Not used	5	0-5	PT1000
U2	A6	Analog	In	OutdoorTempSensor	5	0-5	DHT11
J1	A10	Analog	In	solarPanelCurrent	5	0-5	Ström- mätare(20A)
J2	A11	Analog	In	batteryCurrent	5	0-5	Ström- mätare(20A)
J3	A12	Analog	In	Not used	5	0-5	Ström- mätare(20A)
U6	A14	Analog	In	windSensor	5	0.4-2	Vindsensor
U5	A15	Analog	In	Rainsensor	5	0-5	Regnsensor
U5	48	Digital	Ut	-	5	0-5	Regnsensor
RX	Rxo	UART	In	-	5	0-5	Serieport
TX	Txo	UART	Ut	-	5	0-5	Serie port
5v VDC	5		In/Ut	-	5	5	In/ut- spänning

Tabell 6: portar, signaltyp för Raspberry Pi

Kontakt	PIN	Signal- typ	In/ut- signal	Signalnamn (I mjukvaran)	Spänning (V)	Spännings- intervall (V)	kommentar
TX	8	UART	Out	ttyS0	3.3		Serial port
RX	10	UART	In	ttyS0	3.3		Serial port
5v VDC	2		In	-	5	5	Power input

Bilaga 9: Mjukvara

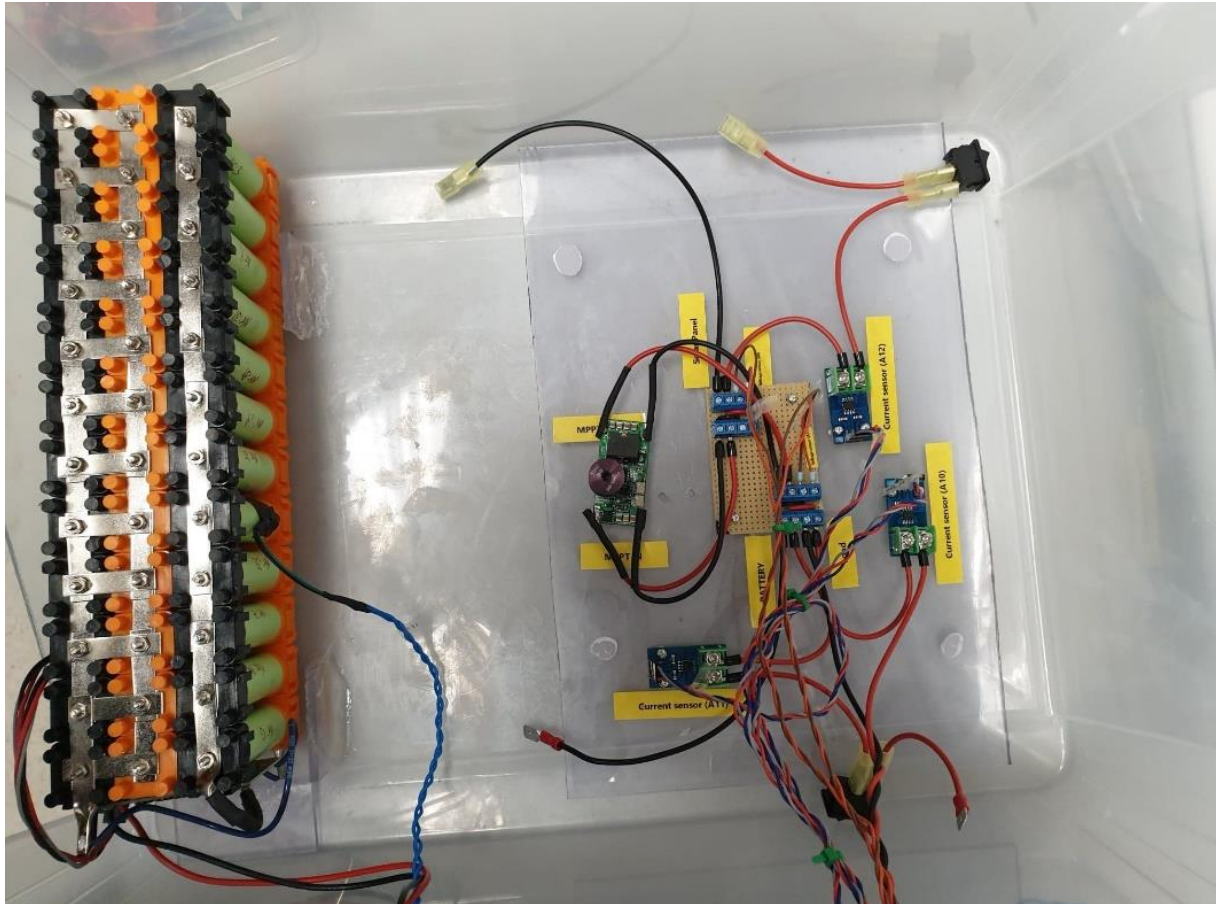
Mjukvara	Innehåll	Länk
Sensornod	Currentsensor.cpp Currentsensor.h RainSensor.cpp RainSensor.h RtdSensor.cpp RtdSensor.h Sampling.cpp Sampling.h Votagedivider.cpp Votagedivider.h Windsensor.cpp Windsensor.h Main.ino	https://gitlab.infotivlab.se/luamol/microgrid
Centralenhet (RPI)	read_data.py	https://gitlab.infotivlab.se/luamol/microgrid-rpi-software
Webapplikation	flask_app.py startscript.sh templates/live.html templates/outdoortest.html templates/testdata.html	https://gitlab.infotivlab.se/luamol/microgrid-webbapp

Bilaga 10: Datablad för solceller

Electrical Characteristics of Typical Cell at Standard Test Conditions (STC)						
STC: 1000W/m ² , AM 1.5g and cell temp 25°C						
Bin	P _{mpp} (W _p)	Eff. (%)	V _{mpp} (V)	I _{mpp} (A)	V _{oc} (V)	I _{sc} (A)
G	3.34	21.8	0.574	5.83	0.682	6.24
H	3.38	22.1	0.577	5.87	0.684	6.26
I	3.40	22.3	0.581	5.90	0.686	6.27
J	3.42	22.5	0.582	5.93	0.687	6.28
All Electrical Characteristics parameters are nominal Unlaminated Cell Temperature Coefficients Voltage: -1.8 mV / °C Power: -0.32% / °C						

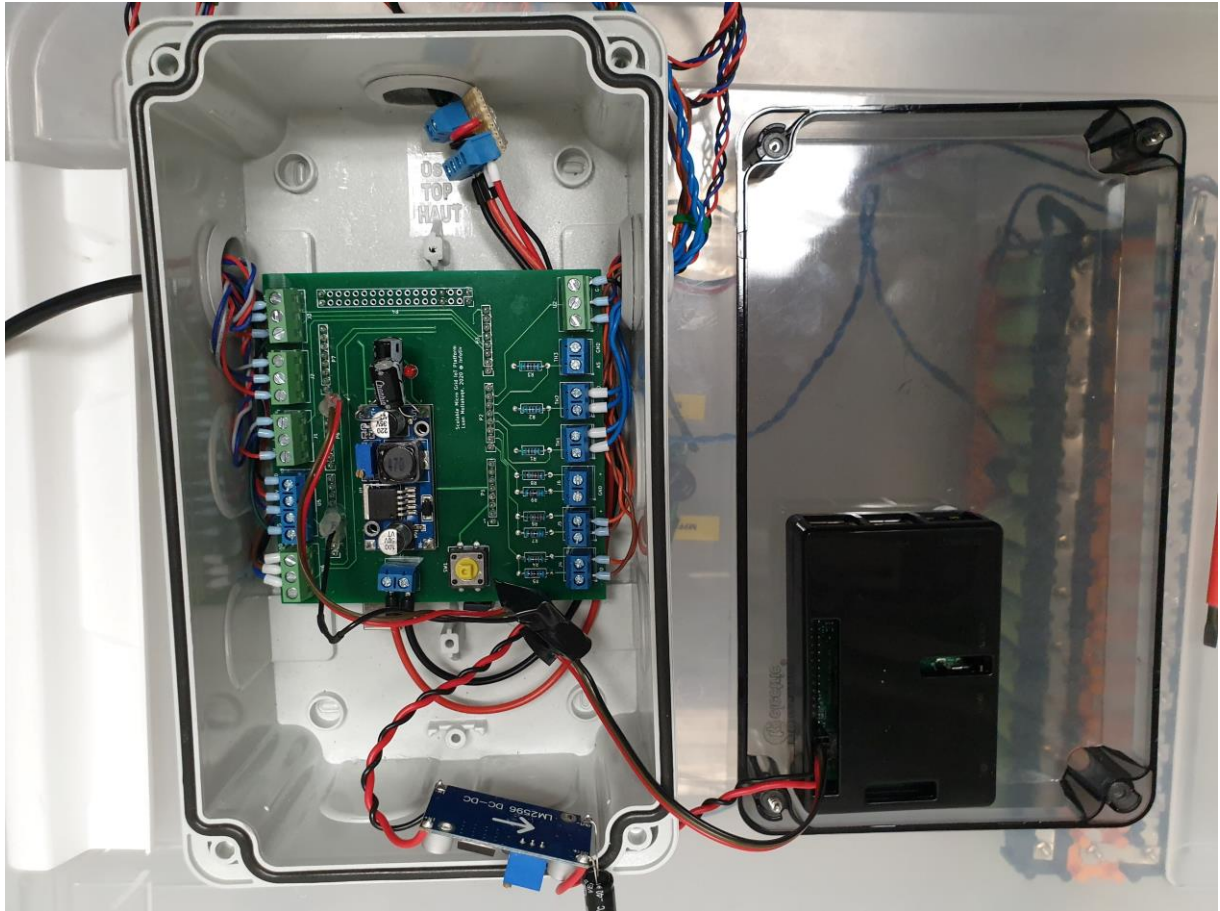
Figur 27:Datablad för SunPower C60 bin H solceller

Bilaga 11: Testlådans innehåll



Figur 28: Testlåda inklusive kopplingsplatta, strömbrytare, strömsensorer och kopplingspunkter för spänningsmätare.

Bilaga 12: Sensornodens innehåll



Figur 29: Sensornodens mekaniska konstruktion och innehåll. Till vänster syns kretskortet som är monterat på en Arduino och samtliga kopplingar. Till höger syns Centralenheten (RPI).