

# IOT3000V-1 21H Internet-of-things

Del 1: lab-rapporter (45%)

Gruppe 4:

- 
- 
-

# Rapport 1

## Målet for eksperimentet

I dette eksperimentet er målet å lage en enkel og «tidlig» prototype som kunne lignet på et trådløst og mobilt vanningsystem som kan overvåke individuelle planter, eller deler av hager ved hjelp av sensor(er) og transivere. I denne første prototypen så er målet å benytte en «måleenhet» til å sende data lest fra sensor, fra den ene kontrolleren til den andre og skrive ut informasjonen som lese av «mottaker».

## Komponenter

Komponenter som ble brukt til dette eksperimentet:

- Arduino Uno x 2
- nRF24L01 Transceiver x2
- Soil moisture sensor
- Processing Board
- Wire-jumper kabler
  - Male-to-male x7
  - Male-to-female x3
  - Female-to-female x2

## Relevant teori

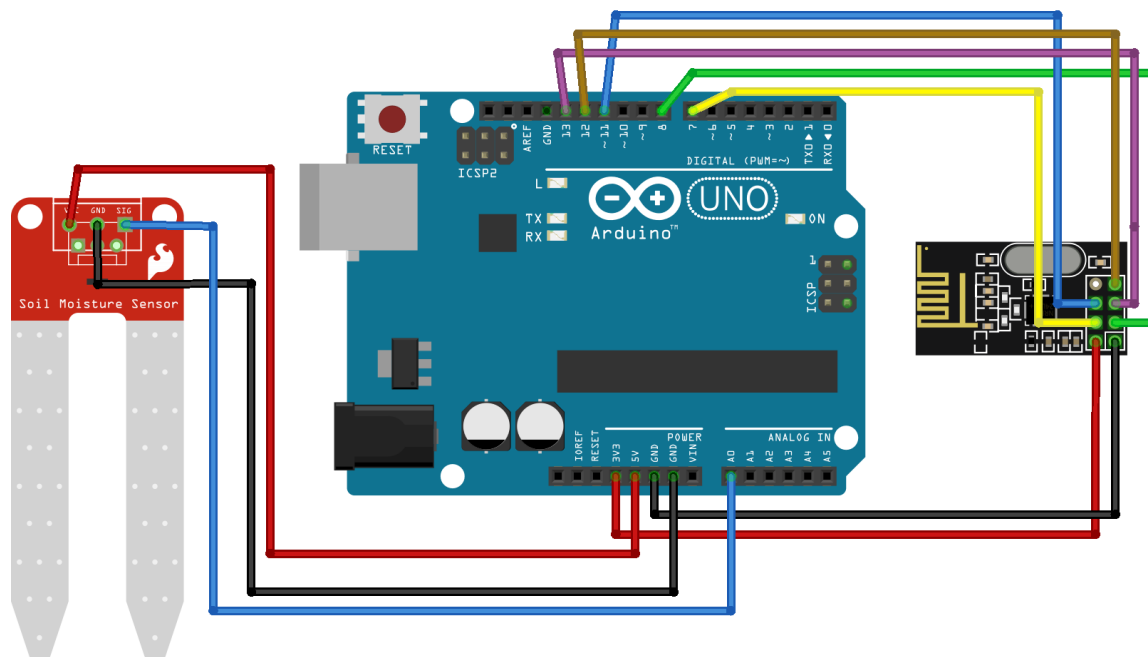
- IOT Lecture III, slide 56-64 – Lasse Berntzen
- IOT Lecture VIII, slides 2-13 – Lasse Berntzen
- nRF24L01 How it works – How to mechatronics
  - <https://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/>
- NRF24L01 Sending and Receiving Integer – Electronics Eternity
  - <https://www.youtube.com/watch?v=3KLRd3opWUo>

## Beskrivelse av oppsett

### Beskrivelse av Leser/sender

På Leser/sender- siden er det brukt Arduino Uno som mikrokontroller, en nRF24L01 Transceiver og en soil moisture sensor. I dette eksperimentet koblet vi også opp «Processing Board» til sensoren for digital avlesning, men benyttet ikke dette. Det er heller ikke tatt med i illustrasjon som vist i figur 1. Programkoden er så lastet opp til enheten, også kan denne enheten fritt brukes uten datamaskin ved hjelp av powerbank eller annen 12-volt strømforsyning.

### Illustrasjon for måleenhet



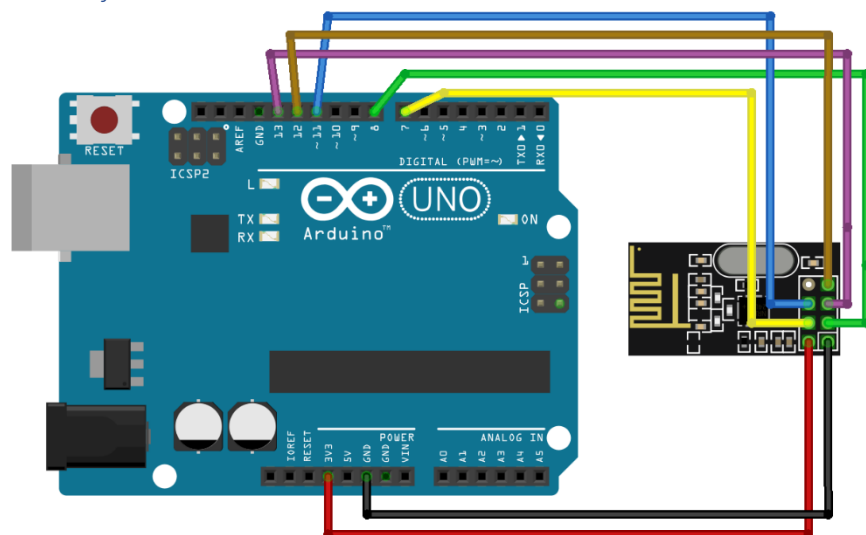
fritzing

Figur 1: Illustrasjon for måleenhet

### Beskrivelse av mottaker

På mottaker-siden er det brukt Arduino Uno som microkontroller og en nRF24L01 Transceiver som mottaker.

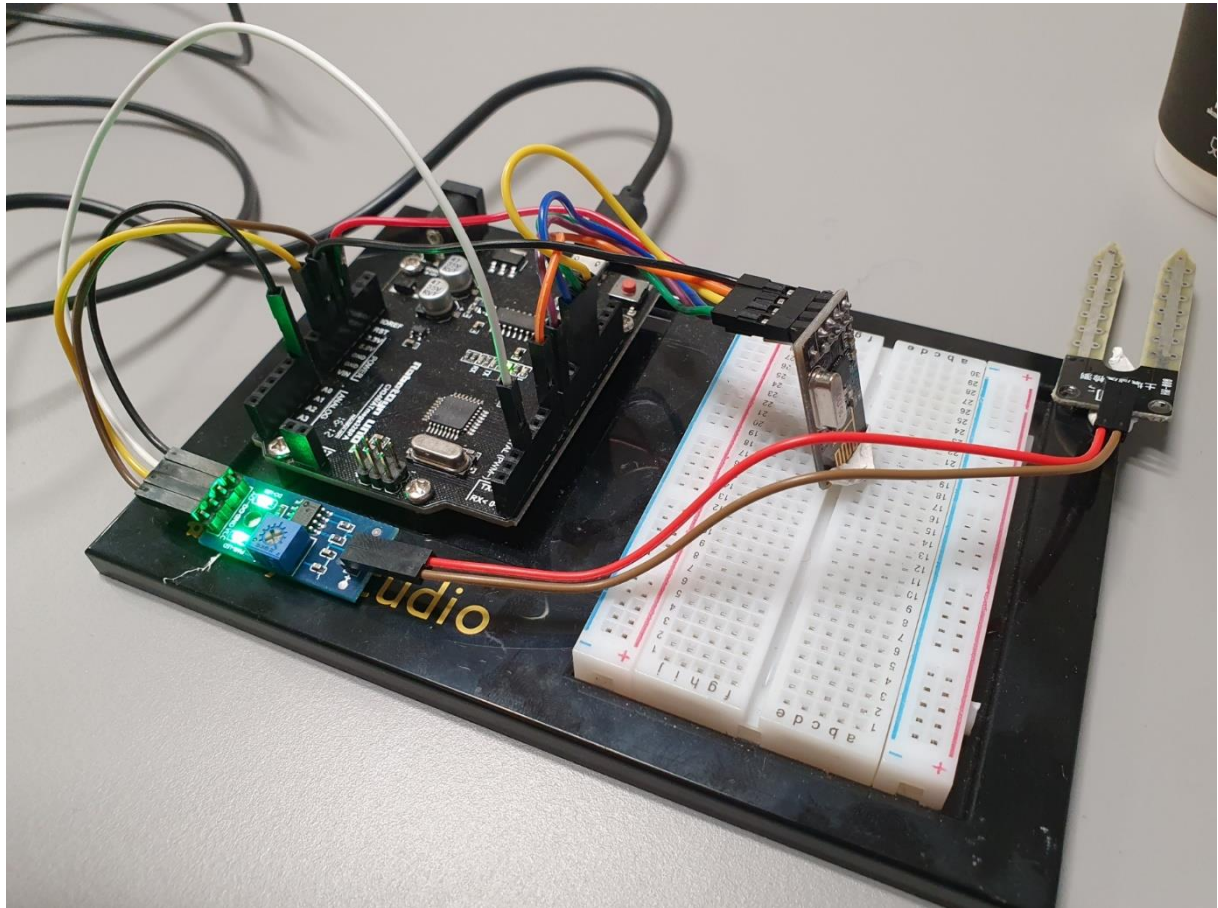
### Illustrasjon av mottaker



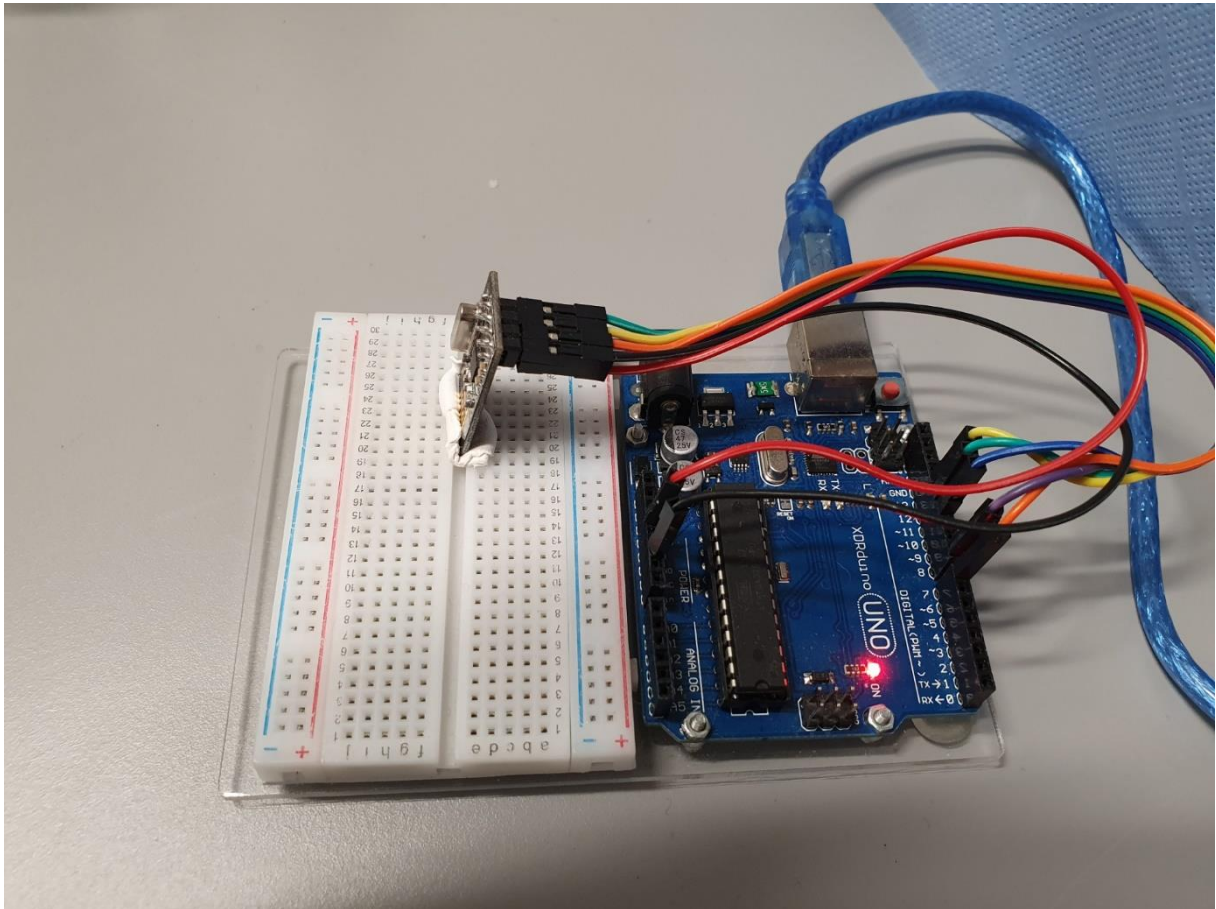
fritzing

Figur 2: Illustrasjon av mottaker

Bilder av måleenhet



*Bilder av mottaker*



## Programkode

Programkode for Måle- og sender-enhet

```
/* Kode hentet fra:
* Arduino Wireless Communication Tutorial
*   Example 1 - Transmitter Code
* by Dejan Nedelkovski, www.HowToMechatronics.com
* Library: TMRh20/RF24, https://github.com/tmrh20/RF24/
*
* Koden er modifisert av oss i forbindelse med eksamen i emnet
* IOT3000, H2021, ved USN Vestfold.
*
* Programkommentar: Oppsettet består av en Arduino, RF24 og Moisture-sensor.
* Data som blir avlest sendes videre til annet RF24-komponent som er
programmert
* til å fungere som receiver.
*
*/

/* Bibliotek-import */
#include <SPI.h>
#include <RF24L01.h>
#include <RF24.h>

/* Pins og kommunikasjonsadresse */
RF24 radio(7, 8); // CE, CSN
const byte address[6] = "00001";

void setup() {
  pinMode(A0, INPUT);
  Serial.begin(9600);
  radio.begin();
  radio.openWritingPipe(address);
  radio.setPALevel(RF24_PA_MAX);
  radio.stopListening();
}

void loop() {
  int value;
  value = analogRead(A0);
  Serial.println("Sending sensor value:"); // Driftsmelding
  radio.write(&value, sizeof(value));      // Skriver/sender data
  Serial.println(value);                   // Driftsmelding
  delay(1000);
}
```

Figur 3: Programkode for sender

## Programkode for mottaker

```
/* Kode hentet fra:
* Arduino Wireless Communication Tutorial
*     Example 1 - Receiver Code
* by Dejan Nedelkovski, www.HowToMechatronics.com
* Library: TMRh20/RF24, https://github.com/tmrh20/RF24/
*
* Koden er modifisert av oss i forbindelse med eksamen i emnet
* IOT3000, H2021, ved USN Vestfold.
*
* Programkommentar: Mottaker som består av Arduino Uno og RF24.
* Programmet mottar, leser og håndterer dataene som kommer inn og
* skriver ut beskjeder til bruker basert på dataene.
*/
// Bibliotek
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
/* Pins og kommunikasjonsadresse */
RF24 radio(7, 8);           // CE, CSN
const byte address[6] = "00001";

/* Oppstart */
void setup() {
    Serial.begin(9600); delay(1000);
    Serial.println("Moisture sensor");           // driftsmelding
    delay(500);
    Serial.println("Preparing to receive signal.."); // driftsmelding
    radio.begin();
    radio.openReadingPipe(0, address);
    radio.setPALevel(RF24_PA_MAX);
    radio.startListening();
    delay(2000);
    // sjekker om det er bytes som kan leses av
    if (radio.available()) {
        Serial.println("Receiver device is ready!");           // driftsmelding
    // brukerveiledning
        Serial.println("Moisture levels from dry to wet: 1024-1");
        delay(4000);
    } else {
        Serial.println("No signal received :(");           // driftsmelding
    }
}
```

Figur 4: Programkode for mottaker, del 1

```

/* Programkommentar:
* Lesing av mottatt data og utskrift:
* Ulike meldinger blir frembragt basert på verdiene gitt fra sensor.
* Her er det statements som dekker hele skalaen til moisture-sensoren, som
vil
* skape en utskrift til «serial monitor» ut ifra verdiene som blir gitt.
*/
void loop() {
// Test whether there are bytes available to be read.

if (radio.available()) {
    int value = "";
    radio.read(&value, sizeof(value));
    if (value == 1022 || value == 1023) {
        Serial.println("-----");
        Serial.println("Plant 1");
        Serial.print("Moisture value: ");
        Serial.println(value);
        Serial.println("Moisture level estimate: 99-100%");
        Serial.println("Soil not found, inspect moisture sensor device");
        Serial.println(""); delay(10000);
    }
    else if (value <= 1021 && value >= 950) {
        Serial.println("-----");
        Serial.println("Plant 1");
        Serial.print("Moisture value: ");
        Serial.println(value);
        Serial.println("Moisture level estimate: 90-98%");
        Serial.println("Soil is dry, add some water");
        Serial.println(""); delay(10000);
    }
    else if (value <= 949 && value >= 800 ) {
        Serial.println("-----");
        Serial.println("Plant 1");
        Serial.print("Moisture value: ");
        Serial.println(value);
        Serial.println("Moisture level estimate (70-89%)");
        Serial.println("Soil is somewhat damp");
        Serial.println(""); delay(10000);
    }
    else if (value <= 799 && value >= 700 ) {
        Serial.println("-----");
        Serial.println("Plant 1");
        Serial.print("Moisture value: ");
        Serial.println(value);
        Serial.println("Moisture level estimate (69-50%)");
        Serial.println("Soil moisture is healthy");
        Serial.println(""); delay(10000);
    }
}
}

```

Figur 5: Programkode for mottaker, del 2



```

else if (value <= 699 && value >= 550 ) {
  Serial.println("-----");
  Serial.println("Plant 1");
  Serial.print("Moisture value: ");
  Serial.println(value);
  Serial.println("Moisture level estimate (49-35%)");
  Serial.println("Soil is pretty wet");
  Serial.println(""); delay(10000);
}
else if (value <= 549 && value >= 386 ) {
  Serial.println("-----");
  Serial.println("Plant 1");
  Serial.print("Moisture value: ");
  Serial.println(value);
  Serial.println("Moisture level estimate (30-34%)");
  Serial.println("Soil is soaking wet");
  Serial.println(""); delay(10000);
}
else if (value <= 385 && value >= 350 ) {
  Serial.println("-----");
  Serial.println("Plant 1");
  Serial.print("Moisture value: ");
  Serial.println(value);
  Serial.println("Moisture level estimate: most likely just pure
water");
  Serial.println("Soil not found, only water. (Typical values) ");
  Serial.println("Inspect moisture sensor device");
  Serial.println(""); delay(10000);
}
else if (value <= 349 && value >= 1 ) {
  Serial.println("-----");
  Serial.println("Plant 1");
  Serial.print("Moisture value: ");
  Serial.println(value);
  Serial.println("Moisture level estimate: Too much water! ");
  Serial.println("Water is wet.");
  Serial.println(""); delay(10000);
}
else {
  Serial.println("Something went wrong");
  delay(10000);
}
}
}

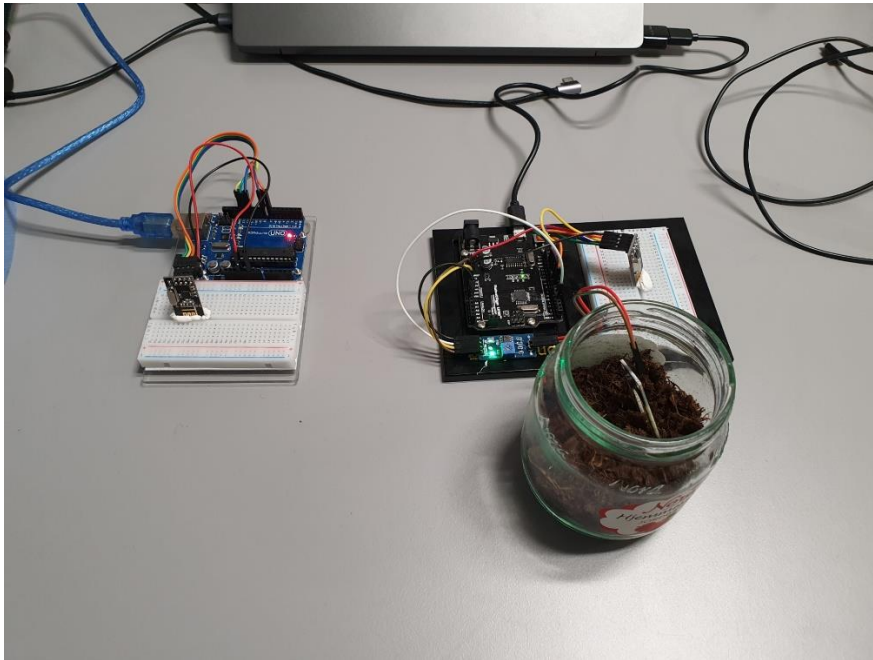
```

Figur 6: Programkode for mottaker, del 3

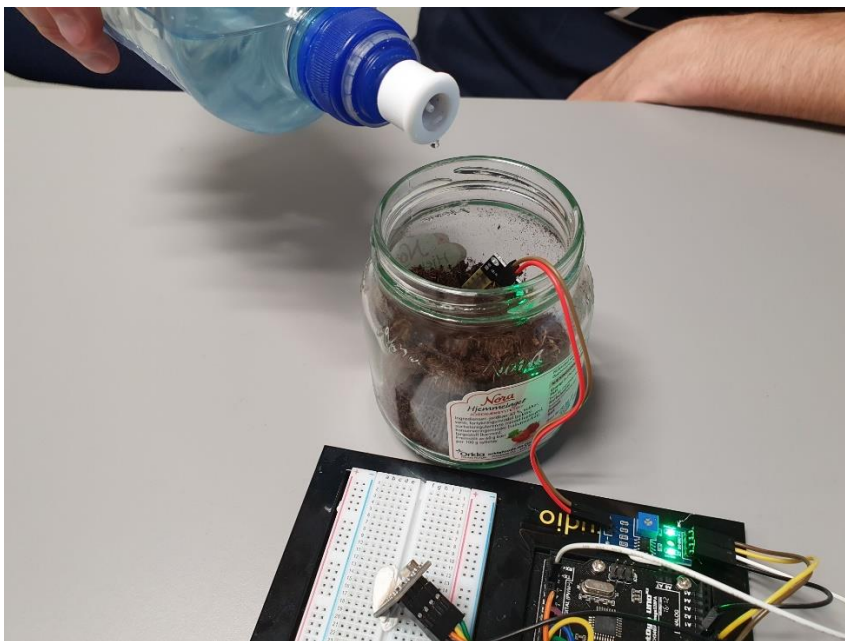
## Diskusjon av resultatet

Vi gjennomførte målene for dette eksperimentet. Men det ble ikke testet stor mer enn å måle litt med sensor i et glass med jord. Men konsept og kode fungerte utmerket. Program for mottaker ble skrevet underveis i testingen av konseptet. Vi startet med relativt tørr jord i glasset, og påførte så litt og litt vann for å bli bedre kjent med hvilke data som kommer fra sensoren. Vi testet også hva slags data vi fikk ved å sette sensor i glass med bare vann i. Det gjorde det mulig å forklare informasjonen til bruker i form av meldinger til «Serial monitor».

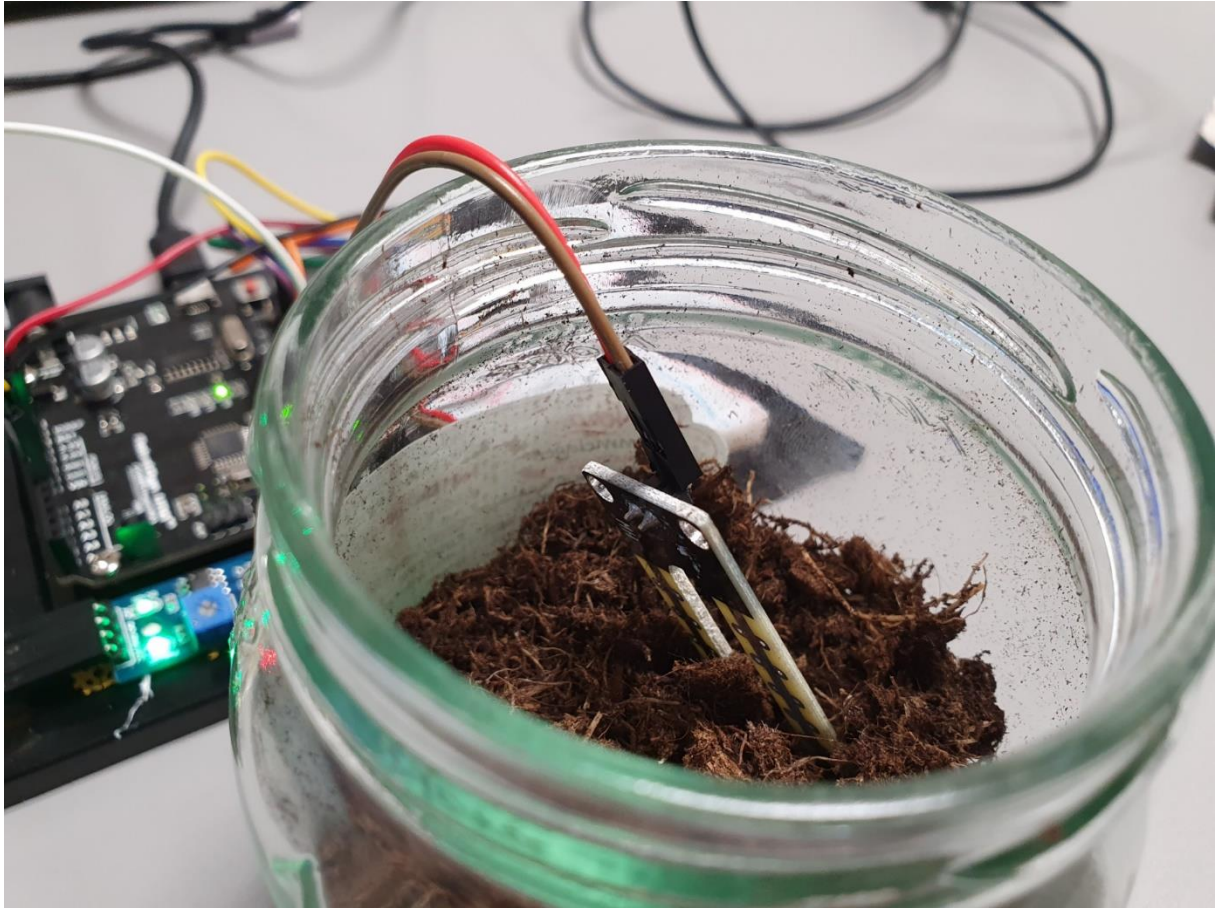
## Bilder av oppsett



Figur 7: Overblikk

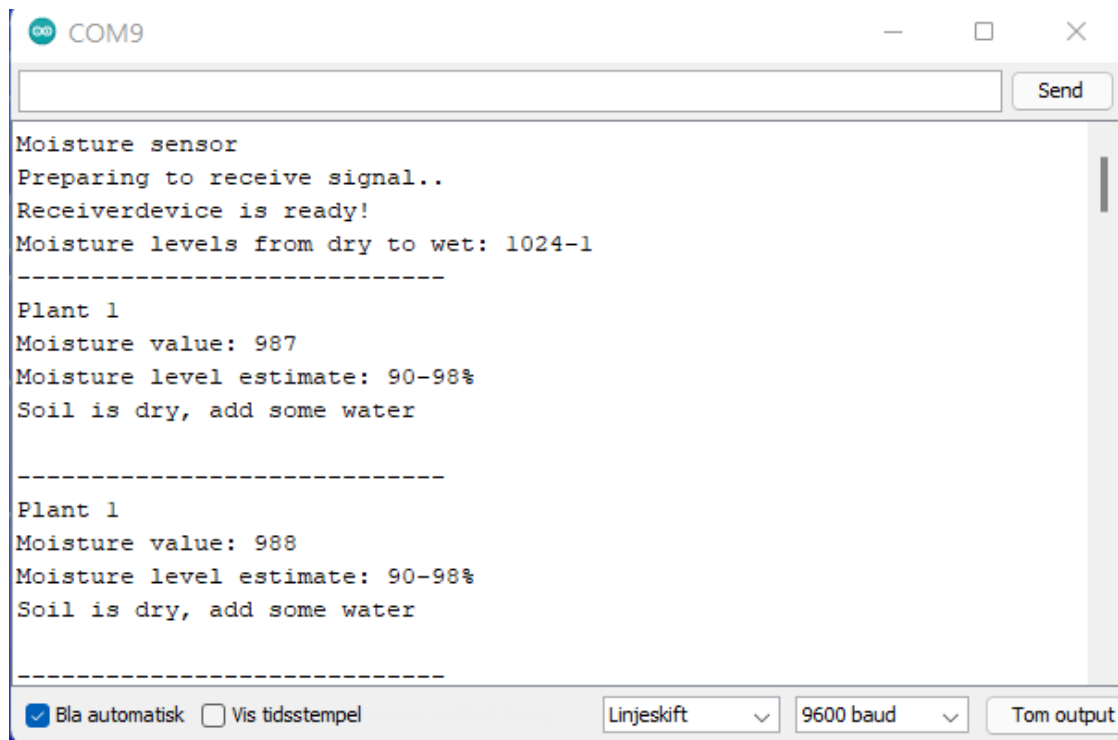


Figur 8: Tilsetter vann

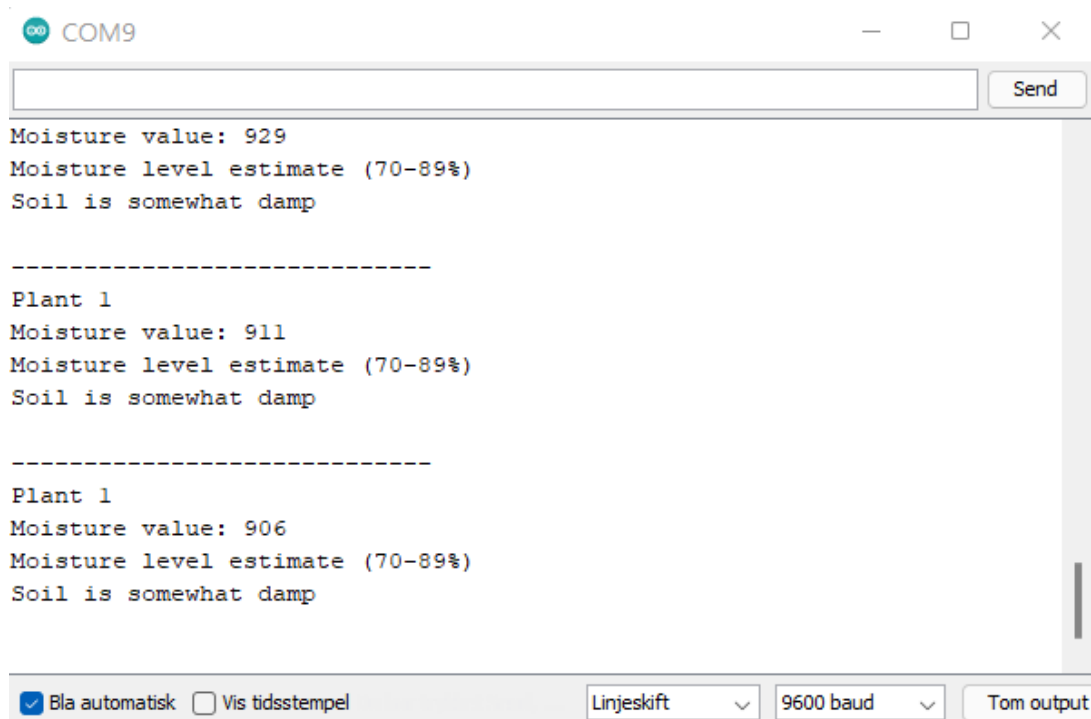


*Figur 9: Soil moisture sensoren i bruk*

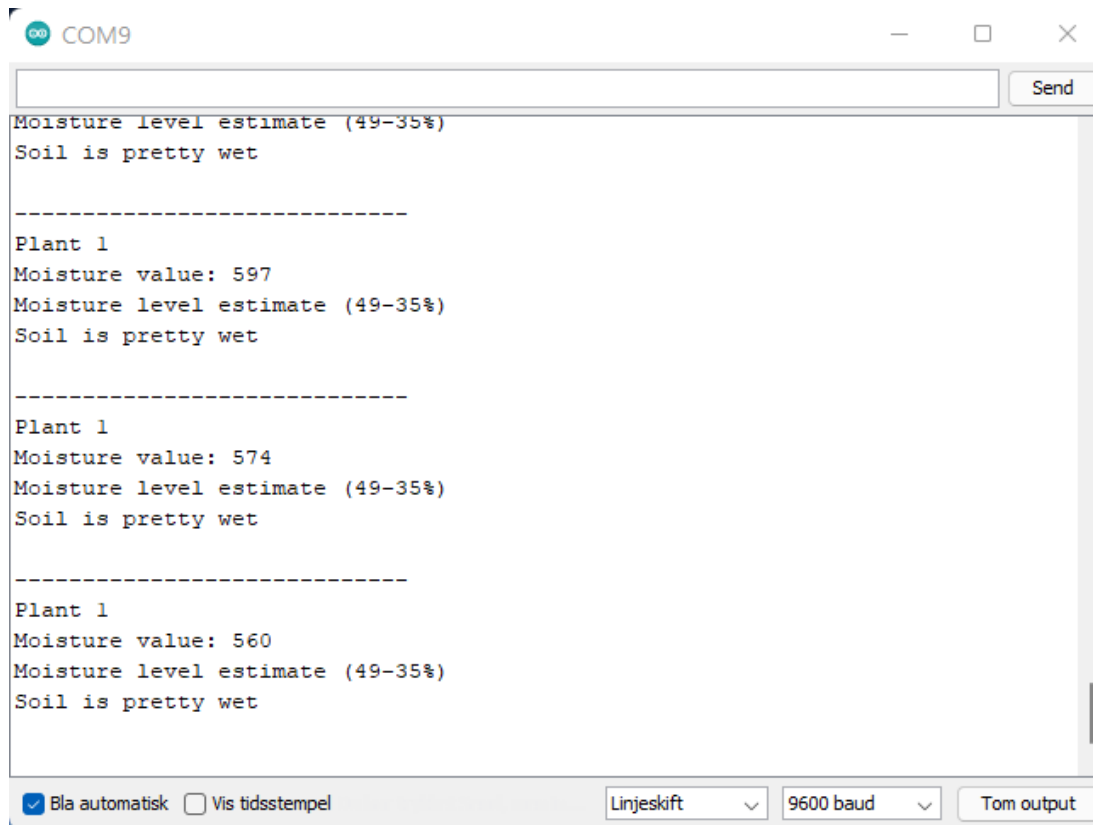
## Bilder fra mottaker – Serial Monitor



Figur 10: Serial monitor skjerm bilde 1



Figur 11: Serial monitor skjerm bilde 2



Figur 12: Serial monitor skjermbilde 3

```

15:07:49.128 -> -----
15:07:49.128 -> Plant 1
15:07:49.177 -> Moisture value: 289
15:07:49.177 -> Moisture level estimate: Too much water!
15:07:49.232 -> Water is wet.
15:07:49.232 ->
15:07:59.173 -> -----

```

Figur 13: Serial monitor skjermbilde 4

```

15:07:38.994 -> -----
15:07:38.994 -> Plant 1
15:07:38.994 -> Moisture value: 378
15:07:39.043 -> Moisture level estimate: most likely just pure water
15:07:39.098 -> Soil not found, only water. (Typical values)
15:07:39.152 -> Inspect moisture sensor device
15:07:39.205 ->

```

Figur 14: Serial monitor skjermbilde 5

## Refleksjon

### Forord

Eksperimentet ble til ved at vi gikk igjennom alle komponentene vi har fått utdelt dette semesteret, hvor vi «brainstormet» hva vi kunne bruke de ulike komponentene til. Noe som bidro til valget av komponenter og konsept var at vi ikke hadde testet ut denne type sensor tidligere i semesteret og tanken på det på å kunne overvåke planter og hage. Vi likte også tanken på at eksperimentet krevde noe andre elementer (glass med jord) utenom komponentene eller luft for å lese data.

### Utskrift av sensor-data

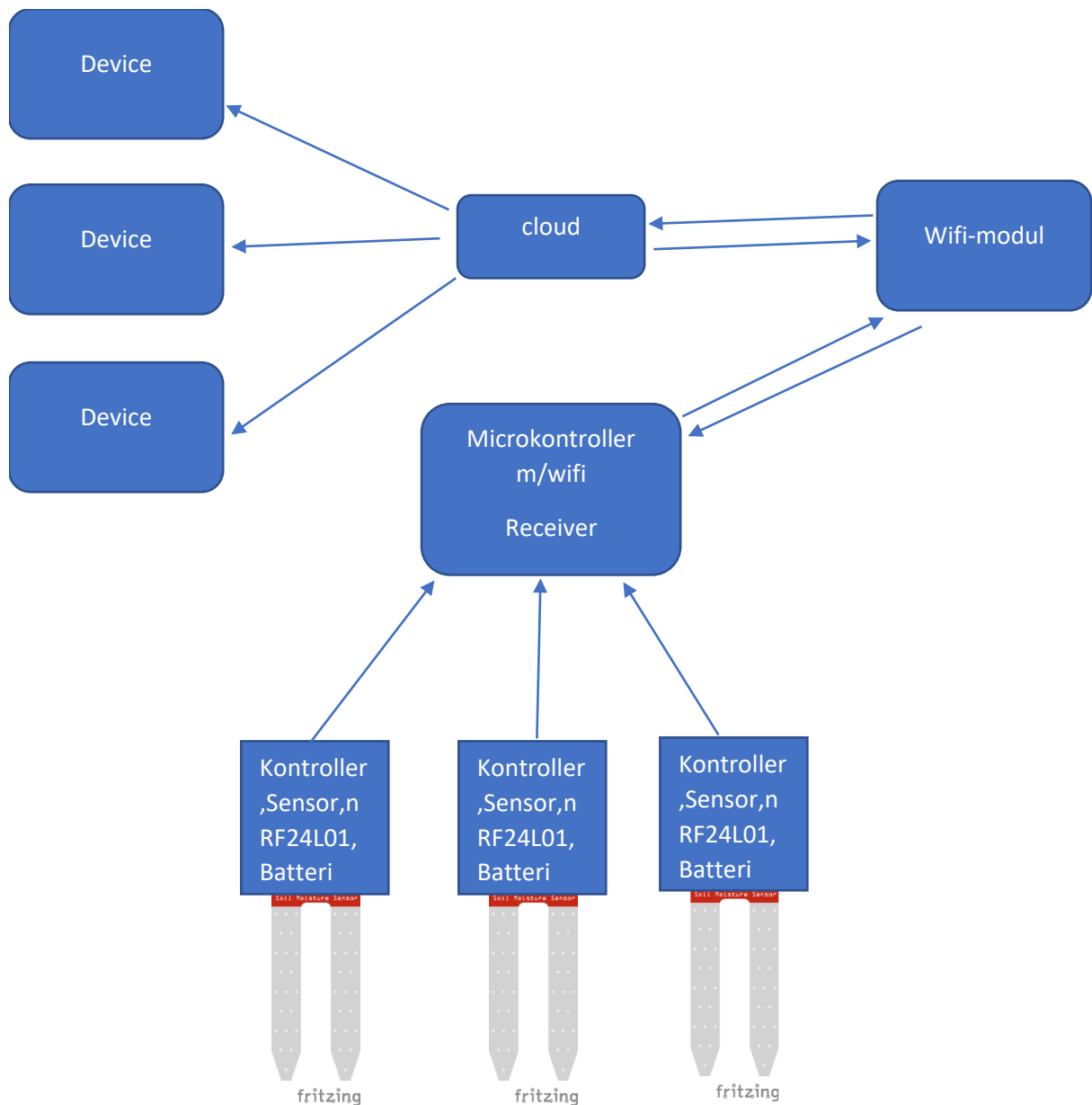
Det er verdt å nevne at meldingene vi får ut til «Serial monitor» bare er basert på funnene vi gjorde under testing av den første prototypen, og som betyr at denne informasjonen ikke er god nok for kommersiell bruk. Sensoren ble ikke kalibrert skikkelig, og vannet vi tilsatte i gjorde ble ikke «rørt inn» skikkelig.

### Prototype v1.1 – Videre mulige steg

Bruke andre typer mikrokontrollere som ESP8266 eller annen Arduino-kontroller med innebygd wifi for å koble det hele til datamaskin over internett for å ta det et steg lengre. Slik at mottakeren kan plasseres strategisk (stasjonært), og at vi har mulighet for å lese informasjonen fra sensor så ulike enheten så lenge man er tilkoblet nett.

Det finnes også ulike typer RF-transivere med støtte for ekstern antenne, noe som kan komme godt med dersom konseptet skal virke på større områder. Fysiske gjenstander som står imellom sender og mottaker vil kunne bli en utfordring. nRF24L01 har teoretisk sett en rekkevidde på 100 meter, men dette krever at ingen fysiske gjenstander står imellom, og at det ellers er lite «forstyrrelser i luften».

nRF24L01 har også mulighet til å kommunisere med 6 andre nRF24L01-enheter samtidig. Dette gjør det mulig i teorien å ha 6 enheter for måling av ulike planter eller områder i en hage, som alle kan rapportere til samme enhet. Ved hjelp av det overstående vil det være mulig å koble dette opp mot IOT-cloud, for å deretter lage publisere dataene på en web eller mobil-applikasjon.



Figur 15: Arkitektur

#### Prototype v.1.2 – Automatisering/smart-Home konsept

Hva med å koble det hele opp på mot en vannbeholder, legge/trekke noen vannrør, vannpumper og muligheten for å velge hvilke type plante du setter sensoren i, og «voila», systemet sørget at den planten til enhver tid har den gunstige fuktigheten i jorden.

## Rapport 2

Målet for eksperimentet

Komponenter

Relevant teori

Beskrivelse av oppsett

Programkode

Diskusjon av resultatet

Refleksjon



## Rapport 3

Målet for eksperimentet

Komponenter

Relevant teori

Beskrivelse av oppsett

Programkode

Diskusjon av resultatet

Refleksjon