

Automatic Waterpump



Namn	Amin Nazari
Utbildning	Mjukvaruutvecklare inom inbyggda system
Uppdrags-givare	Cervitrol AB, Fielder & Lundgren
Handledare/examinator	Tony Månsson
Datum	25/04–2025

Sammanfattning

Detta examensarbete handlar om utveckling av ett automatiskt bevattningssystem baserat på mikrokontrollern ESP32 nano, med komponenter som jordfuktighets- och vätskenivåsensor, MQTT-kommunikation och en LCD-display och en knapp. Projektets syfte var att skapa ett smart vattensystem som självständigt kan känna av om att vattna och efter det aktivera en vattenpump vid behov, samt möjliggöra fjärrstyrning och statusövervakning i realtid.

Systemet bygger på två skript. Det första skriptet ansvarar för automatisk aktivering av pumpen baserat på sensordata. Om jorden är torr och det finns vatten i behållaren, startas pumpen automatiskt.

Det andra skriptet hanterar LCD-displayen och en knapp som manuellt kan aktivera pumpen. All kommunikation sker via MQTT-protokollet med en publik broker där statusmeddelanden skickas och tas emot. LCD-displayen visar den aktuella statusen, och användaren kan också påverka systemet via MQTT-meddelanden, till exempel för att slå på pumpen manuellt. En knapp används för att skicka dessa kommandon, det ökar användarvänligheten.

Innehållsförteckning

INLEDNING.....	1
BAKGRUND.....	1
SYFTE.....	1
PROBLEMFORMULERING	2
ÄVGRÄNSNINGAR OCH FOKUS	2
METOD/ARBETSSÄTT	3
TEKNIK – BILL OF MATERIAL.....	5
ESP32 NANO – SYSTEMETS HJÄRNA	5
JORDFUKTIGHETSSENSOR – KÄNNER AV TORR JORD	5
VATTENNIVÅSENSOR – SER OM DET FINNS VATTEN	5
RELÄ – SLÄR PÅ OCH AV PUMPEN	6
VATTENPUMP – VATTNA BLOMMORNA.....	6
LCD-DISPLAY – VISAR STATUS	6
KNAPP – FÖR MANUELL STYRNING.....	6
PROGRAMMERING MED MYCROPYTHON OCH THONNY	6
RESULTAT.....	7
DISKUSSION	8
SLUTSATS.....	10
REKOMMENDATIONER.....	10
REFERENSLISTA	11
BILAGOR.....	15

Inledning

Bakgrund

Under utbildningen till mjukvaruutvecklare för inbyggda system har vi fått arbeta med olika tekniker för att kombinera hårdvara och mjukvara. För att fördjupa mina kunskaper valde jag att utveckla ett automatiskt bevattningssystem som examensarbete.

Jag har själv ett behov av denna typ av produkt, eftersom jag reser mycket och inte alltid kan vara hemma för att ta hand om mina växter. Tidigare har jag sett vänner att vattna dem, men det har flera gånger lett till att växterna över vattnats och dött. Jag ville undvika detta och behövde ett smart bevattningssystem som kan känna av när jorden är torr och automatiskt tillföra vatten.

Arbetet gav mig också chansen att använda det jag har lärt mig om inbyggda system i ett praktiskt projekt som jag själv har nytta av.

Syfte

Syftet med det här examensarbetet är att skapa ett automatiskt bevattningssystem som kan känna av om jorden är torr eller fuktig och om det finns vatten i en behållare. När det behövs ska systemet själv kunna starta en pump och vattna. Arbetet bygger på mikrokontroller ESP32 för att styra och följa systemet på distans.

Arbetet syfte ska ge en bättre förståelse för hur man utvecklar inbyggda system där hårdvara, mjukvara och nätverkskommunikation sammankopplas i ett praktiskt projekt. Och framför allt få mina blommor att må bra.

Följande funktioner som systemet ska uppfylla:

- En pump ska automatiskt aktiveras när jorden är torr och det finns vatten i behållaren.
- En LCD-display ska visa aktuell status för systemet.
- En knapp ska kunna användas för manuell aktivering av pumpen.
- En MQTT-anslutning ska möjliggöra fjärrövervakning och fjärrstyrning av systemet.

Problemformulering

För att jag ska bygga ett automatiskt bevattningssystem som verkligen fungerar så finns det flera saker jag måste lösa. Systemet behöver kunna känna av rätt information, ta rätt beslut efter informationen och ska fungera säkert.

- Hur kan användaren övervaka och styra systemet via MQTT och en knapp?
- Hur kan ett system automatiskt avgöra när växter behöver vattnas?
- Hur kan man kombinera olika data för att göra smarta beslut i det automatiska systemet.

Avgränsningar och fokus

I detta arbete ville jag fokusera på att bygga ett enkelt automatiskt bevattningssystem som använder sensorer, mikrokontroller, knapp, LCD-display, pump och MQTT-kommunikation. Målet har varit att skapa ett system som fungerar i praktiken och som jag själv har nytta av.

Jag har inte valt att:

- Utveckla en mobilapp eller webbgränssnitt
- Använda andra sensor typer, t.ex temp – eller ljussensor
- Bygga ett eget MQTT-broker system (använde en publik broker i stället)
- Designa en fysisk låda till produkten

Jag har valt att fokusera på:

- Få sensor, pump och MQTT att fungera ihop.
- Visa status på en LCD-display.
- Testa manuell styrning med en knapp.
- Bygga ett stabilt och enkelt system som är lätt att använda.

Metod/Arbetssätt

Arbetet började med att jag funderade över vilka komponenter som skulle behövas för att bygga ett automatiskt bevattningssystem. Jag tänkte igenom vilka sensorer som behövde användas, vad de skulle känna av och vad som skulle aktiveras när vissa värden uppnåddes. Jag jämförde olika typer av sensorer och valde ut de som passade bäst för mitt arbete. Därefter gjorde jag en lista över vilka komponenter som behövdes. Jag valde att använda en jordfuktighetssensor för att mäta fuktnivån i jorden och en vattennivåsensor för att kontrollera om det fanns vatten i behållaren. Från början var det inte min plan att ha med en knapp för manuell styrning eller en sensor som läser av vattennivå, men under arbetets gång insåg jag att det skulle göra systemet båda användarvänligt och stabilare. Jag gick till en affär och köpte två platslådor. En för att förvara vattnet och en för att skydda komponenterna. Vattenpumpen ligger i lådan med vattnet. Jag borrade små hål i lådorna så att jag kunde dra igenom strömkablar till pumpen, samt extra hål för att slangens ska kunna gå från pumpen till blomman. Och hål för strömkablarna för vätskenivåsensor. Ett chassi var inte bestämt heller i min planering men kände att det hade varit bättre att organisera komponenterna och samtidigt skydda dem.

Innan jag byggde ihop hela produkten testade jag varje komponent var för sig. Det hjälpte mig att förstå hur de fungerade och att upptäcka möjliga problem tidigt. Jag använde databladern för att förstå hur varje komponent skulle kopplas och programmeras. För att skriva och testa kodernan använde jag Thonny som utvecklingsmiljö eftersom det fungerar bra med ESP32 Nano när man programmerar i MycroPython.

MQTT-kommunikationen var ett område vi har lärt oss om på flera lektioner, där vi har gått igenom hur enheter kan kommunicera med varandra trådlöst. Det blev ett enkelt val att välja just MQTT protokollet som en lösning för att kunna övervaka och styra systemet på distans. Jag använde en publik MQTT-broker för att skicka och ta emot meddelande mellan ESP32:n och min dator.

När alla komponenter fungerade var för sig, så började jag bygga upp systemet stegvis. Jag började med att koppla upp sensorerna till ESP32 NANO och skriva kod för att läsa data från dem. Jag stötte på problem när jag skulle styra vattenpumpen. Eftersom till början saknade jag ett relä som kunde styra pumpen. När jag fick tag på ett relä kunde jag därefter koppla in och styra pumpen korrekt via mikrokontroller. Jag la också till en LCD-display och en

knapp för användaren enkelt skulle kunna styra pumpen manuellt och se aktuell status i systemet. Display visade till exempel om jorden var torr eller om vattennivån var låg.

Jag märkte att det ibland tog lite tid innan knappen reagerade, eftersom systemet först gick igenom sensorerna och andra funktioner innan det hann kolla om knappen var tryckt. För att lösa detta använde jag en interrupt vilket gör att systemet reagerar snabbare när knappen trycks ner utan att behöva vänta på hela loopen.

Och samtidigt märke jag av att knappen skicka fler signaler vid bara ett knappt tryck och jag förhindra det genom att lägga på ”debouncing” i koden. Det gör att systemet bara reagerar på ett tryck i taget, vilket ger en mer stabil funktion.

Jag testade också olika scenarier som till exempel torr eller låg vattennivå, för att se hur systemet reagerade. Det gav mig möjlighet att justera koden och säkerställa att systemet fungerar stabilt. Under testningen märkte jag att jag behövde justera tröskelvärdet för att systemet skulle reagera mer exakt när jorden blev våt och torr.

Teknik – Bill of material

Här är vad för teknik som användes i arbetet. Jag går igenom vilken mikrokontroller jag valde, vilka sensorer och andra komponenter som användes. Hur kommunikationen fungerar och vilket programmeringsverktyg jag jobba med.

ESP32 nano – systemets hjärna

ESP32 Nano är en smart mikrokontroll som styr hela systemet. Den fungerar som hjärnan i projektet. Den har inbyggd Wi-Fi, vilket gör att man kan kopplas upp trådlöst mot min dator. ESP32 har flera in- och utgångar där man kan koppla in sensorer, knappar, displayar och andra komponenter. I mitt arbete används den för att läsa jordfuktighet och vattennivå, visa information på en LCD-display, ta emot tryck från en knapp och styra en pump.

Jag valde denna mikrokontroller för den är enkel att använda och har de funktioner jag behövde. Den klarar av att både läsa data och kommunicera trådlöst med MQTT utan problem.

278*2 = 556 kr

Jordfuktighetssensor – känner av torr jord

Jordfuktighetssensor mäter hur mycket vatten det finns i jorden. Den fungerar genom att mäta motståndet mellan två metallpinnar. När jorden är torr så är motståndet högt, och när jorden är blöt så är motståndet lägre. Sensorn ger ett värde som mina mikrokontroller kan läsa.

29 kr

Vattennivåsensor – ser om det finns vatten

Vattennivåsensor kan känna av vätska genom plats eller andra icke metalliska behållare. Sensorn sätts utanpå en vattenbehållare och känner av om det finns vätska på insidan eller inte. Det gör att man inte behöver ha någon sensor i direkt kontakt med vatten.

179 kr

Relä – slår på och av pumpen

Reläet fungerar som en elektrisk strömbrytare. ESP32 kan inte driva pumpen direkt så används reläet för att koppla på eller av pumpens ström. Reläet styrs med en signal från mikrokontrollen och kan kopplas så att det aktiveras när jorden är torr.

42 kr

Vattenpump – vattna blommorna

Pumpen som används är en liten nedsänkbar pump. Den drivs med låg spänning men har tillräckligt tryck för att vattna en blomkruka. Pumpen är kopplad till reläet som styr när den ska sättas i gång.

59 kr

LCD-display – visar status

LCD-displayen som används är en 16x2. Det innebär att den kan visa 16 tecken och på 2 rader. Den visar aktuella statusen i systemet. Som till exempel om jorden är torr, om det finns vatten i behållaren eller om pumpen är i gång.

Displayen är kopplad till mikrokontrollen med hjälp av I2C-protokollet, vilket gör att man bara behöver två signalkablar (SDA och SCL) i stället för många. Det förenklar kopplingen och sparar plats på breadboarden.

99 kr

Knapp – för manuell styrning

Knappen används om användaren vill starta pumpen manuellt. När knappen trycks ner skickas en signal till mikrokontroller som då aktiverar pumpen med en fras "turn_on".

99 kr

Programmering med MycroPython och Thonny

Mikrokontrollen är programmerad med MycroPython. Jag använde Thonny som utvecklingsmiljö för att skriva och testa koden. Det fungerar bra för att ladda upp koden och se resultat direkt på mikrokontrollen.

Totala kostnaden: 1063 kr

Resultat

Detta arbete som jag utvecklat består av två ESP32 nano, en jordfuktighetssensor, en vattennivåsensor, en relämodul, en vattenpump, en LCD-display samt en manuell knapp. Systemet är uppbyggt på en breadboard med hjälp av kopplingskablar.

Resultatet visar att systemet kan känna av om jorden är torr eller fuktig samt kolla om det finns vatten i behållaren. När både villkoren om att jorden är torr och det finns vatten i behållaren uppfylls aktiveras vattenpumpen automatiskt via relät. Systemet fungerar även med en manuell aktivering med ett tryck som skickar ut kommandot "turn_on" för att starta pumpen.

Systemet status visas i realtid på en LCD-display. Den informerar om jordens status, vattennivå samt om pumpen är aktiv eller inte. All data skickas via MQTT-protokollet till en publik broker.

När jag gjorde tester visade systemet göra det som förväntades. Ibland kunde det vattna för mycket eftersom tröskelvärdet inte hade korrigerats korrekt. Men i helhet fungerade när dem två villkoren uppfylldes så aktiverades pumpen. Och vid låg vattennivå eller fuktig jord aktiverades inte pumpen alls.

#Automatiskt start av pump när jorden är torr och vatten finns:

```
if moisture_value == 1 and water_level_value == 1:  
    relay_pin.on()
```

Knappen som skickar "turn_on" via MQTT:

```
def handle_button(pin):  
    publish_status(b"turn_on")
```

#Visning av meddelande på LCD-display

```
def callback_print(topic, msg):  
    lcd.clear()  
    lcd.putstr(msg.decode())
```

Diskussion

Syftet med detta arbete var att bygga ett automatiskt bevattningssystem som ska känna av när jorden är torr och vattna växterna automatiskt. Systemet skulle också visa status och gå att styra på distans via MQTT. Jag tycker jag har uppfyllt det syftet.

De frågorna jag ställde i början har jag kunnat besvara genom att bygga och testa systemet. Systemet kan känna av när jorden är torr, kolla om det finns vatten i behållaren och starta pumpen automatiskt. Jag har också lagt till en knapp för manuell vattning och en LCD-display som visar den aktuella statusen. Data skickas via MQTT vilket gör att man kan övervaka systemet trådlöst.

Det som inte fungera direkt var pumpstyrningen. Jag saknade ett relä som kunde styra pumpen. När jag väl fick tag på ett relä så kunde jag styra pumpen beroende om jorden är torr och det finns vatten i behållaren. Sen var det också att jag behövde justera tröskelvärdet så rätt mängd vatten skulle pumpas tills det blev fuktigt tillräckligt nog.

En annan sak som jag märkte var att knappen ibland reagerade långsamt. Det berodde på att systemet går igenom flera kontroller i loopen som att läsa sensorer och visa information innan den kommer till att kolla om knappen är tryckt. Det gör att knappen måste vänta på sin tur, vilket gör att pumpen inte startar direkt. Jag löste detta senare genom att använda en interrupt i stället så att knappen kunde reagera direkt när man trycker. Det gjorde systemet mycket snabbare. Jag la också till något som kallas ”debounce”, vilket gör att knappen inte skickar flera signaler av misstag om man råkar trycka för snabbt. Det som hände innan var att jag tryckte på knappen en gång men det svara med att det blev mer knapptryck. När jag la till ”debounce” så gjorde det knapp styrningen mer stabil.

Det hela arbetet har lärt mig hur man bygger upp ett system med sensorer, hur mikrokontrollen fungerar och hur man använder MQTT för trådlös kommunikation. Jag lärde mig också hur viktigt det är att testa varje del för sig innan man bygger ihop allt.

En sak jag kunde ha gjort bättre var att planera chassit redan från början. Jag byggde det i efterhand och fick borra hål för kablar och slang. Det fungerade men det hade blivit enklare med planering och kanske fått ett snyggare, stabilare och säkrare chassi.

Jag visar redan status i systemet via en LCD-display och skickar meddelande med MQTT. Men jag hade kunnat lägga till ett tydligare larmsystem. Kanske jag borde kunde ha en funktion som skickar en notis eller ett ljudlarm om något är fel under en längre period, som att vattnet är slut i flera timmar eller om jorden är konstant torr. Det skulle öka säkerheten ytterligare.

Jag hade också kunnat bygga en enkel hemsida eller app som tar emot MQTT-data och visar status på ett tydligare sätt. Det hade gjort det enklare att följa systemet på distans.

Det här är ett enkelt system men det gör det också lätt att förstå och bygga vidare på. Jag har fokuserat på att få grunderna att fungera stabilt vilket jag tycker att jag har lyckats med.

Slutsats

Jag har byggt ett automatiskt bevattningssystem som fungerar enligt syftet. Systemet vattnar när jorden är torr och vatten finns i behållaren, och det går också att styra manuellt med en knapp via MQTT. Jag är nöjd med resultatet och har lärt mig mycket om både hårdvara och programmering. Systemet fungerar stabilt efter justeringar.

Rekommendationer

Rekommendationer och möjliga förbättringar beskrivs i kapitel 4. Diskussion. Dessa handlar bland annat om larmsystem, webbaserad visning och bättre planering av chassit.

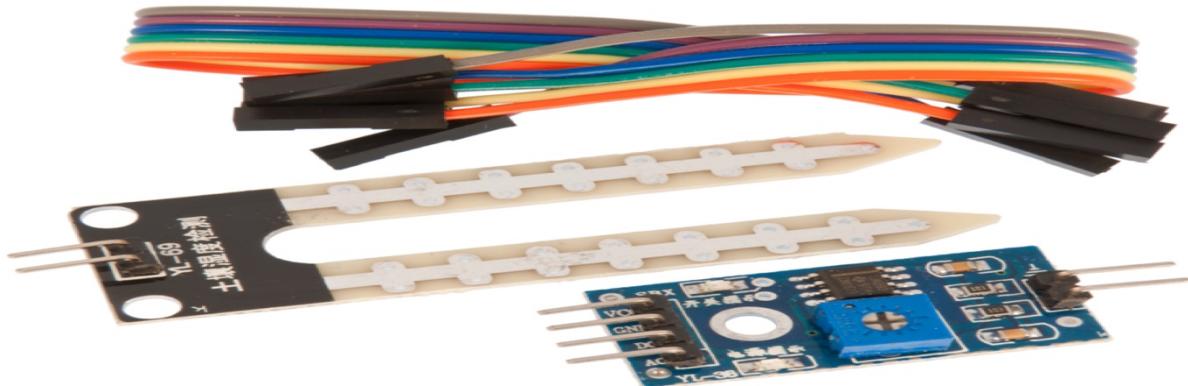
Referenslista

Fuktighetssensor:

[.../.../Downloads/jordfuktighetsensor.pdf](#) - Datablad

https://www.electrokit.com/jordfuktighetssensor?gad_source=1&gbraid=0AAAAD_OrGPwO1eEjW7exDbQY0-OVBiYt&gclid=Cj0KCQjwzYLABhD4ARIsALySuCQKw1AHlok85Nuw5rnUQ3h7kRUEIl9FhM-hfB2ceNa8j9I6ocfwq1saAiRqEALw_wcB

https://lastminuteengineers.com/soil-moisture-sensor-arduino-tutorial/#google_vignette?utm_content=cmp-true



Vattennivåsensor:

[.../.../Downloads/Vattennivåsensor.pdf](#) - Datablad

https://www.electrokit.com/vatskenivasensor-kontaktlos-kapacitiv-5-24vdc?gad_source=1&gbraid=0AAAAAAD_OrGPwO1eEjW7exDbQY0-OVBiYt&gclid=Cj0KCQjwzYLABhD4ARIsALySuCRjOqXW9nIYKh_EWpI

QJBIJc3ao94Ur7akfAMDGN5bXcdcUMiBcJZEaAoD4EALw_wcB



Dränkbar pump:

[.../.../Downloads/Dränkbar pump.pdf](#) - Datablad

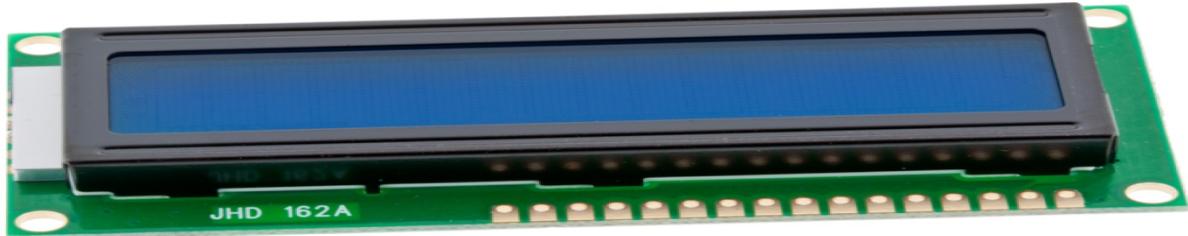
https://www.electrokit.com/drankbar-pump-3v?gad_source=1&gbraid=0AAAAAAD_OrGPwO1eEjW7exDbQY0-OVBiYt&gclid=Cj0KCQjwzYLABhD4ARIsALySuCR9j-73lxJ3jd-hIS36wYmGFHdjf95iWMvCG0lLwbST41clepcsbAaAgaVEALw_wcB



LCD-display:

https://www.electrokit.com/lcd-2x16-tecken-jhd162a-stn-bla/vit-led?gad_source=1&gbraid=0AAAAAAD_OrGPwO1eEjW7exDbQY0-OVBiYt&gclid=Cj0KCQjwzYLABhD4ARIsALySuCQcwKA9eJLaSEcr33QKn

gNEJU1D7o7UhMaYdt5pqoaRc6GXrHi7RpUaAu3LEALw_wcB



I2C-interface:

<https://www.electrokit.com/i2c-interface-for-lcd>

<https://www.geeksforgeeks.org/i2c-communication-protocol/>



Knapp:

[.../.../Downloads/knapp.jpg](#) - Datablad

https://www.electrokit.com/knappar-pcb-sortiment-12st?gad_source=1&gbraid=0AAAAAAD_OrGPwO1eEjW7exDbQY0-OVBiYt&gclid=Cj0KCQjwzYLABhD4ARIIsALySuCRvCW6nVoX65InoMmo9

orzfUvu6y3TW4IkxcfOaH2HL3pCwTam6Lf0aApV9EALw_wcB

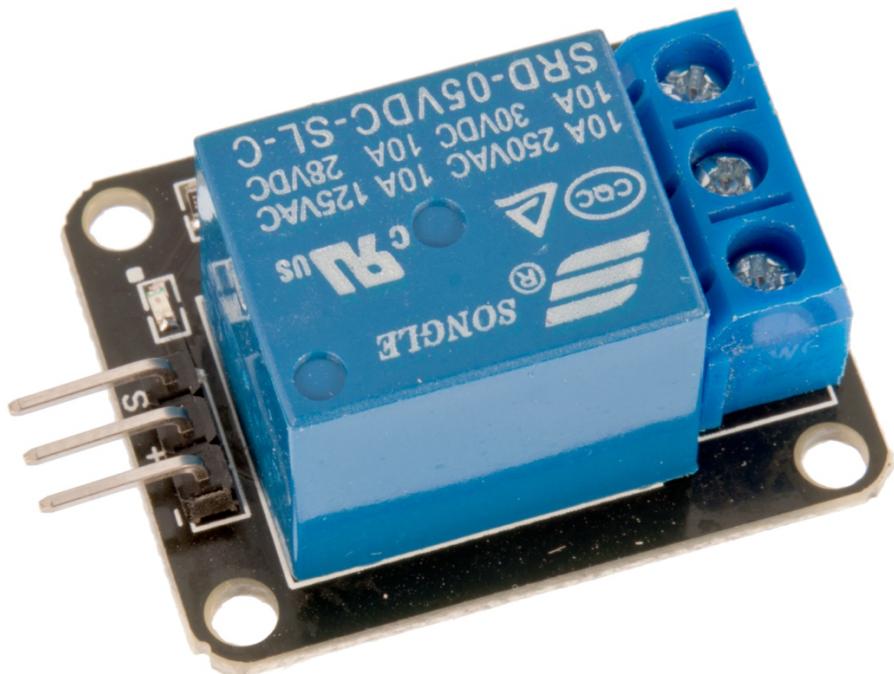


Relä:

../../../../Downloads/relä.pdf - Datablad

https://www.electrokit.com/relamodul-5v?gad_source=1&gbraid=0AAAAAAD_OrGPwO1eEjW7exDbQY0-OVBiYt&gclid=Cj0KCQjwzYLABhD4ARIsALySuCSrjVkJk-YsQTnMZBjNnS2ezeUCHtwKidIqt4wuif6qPRHtcUXz4saArM4EALw_wcB

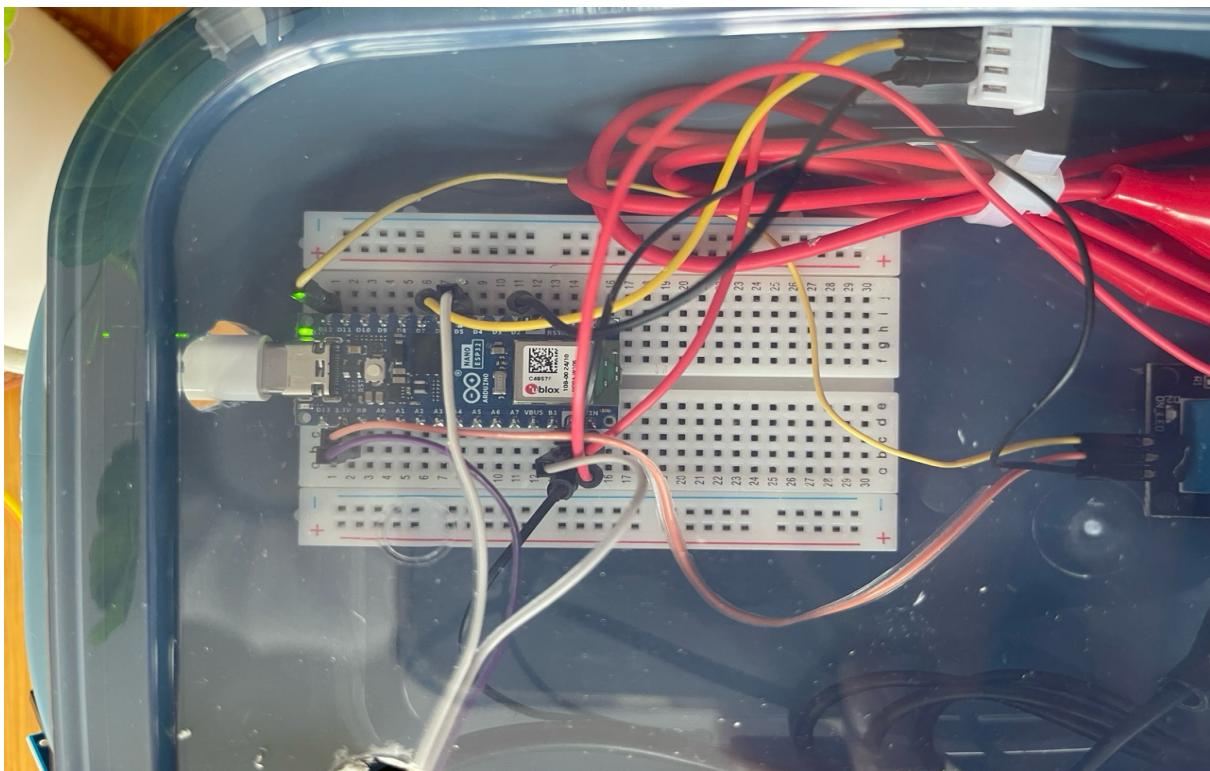
https://www.youtube.com/watch?v=n594CkrP6xE



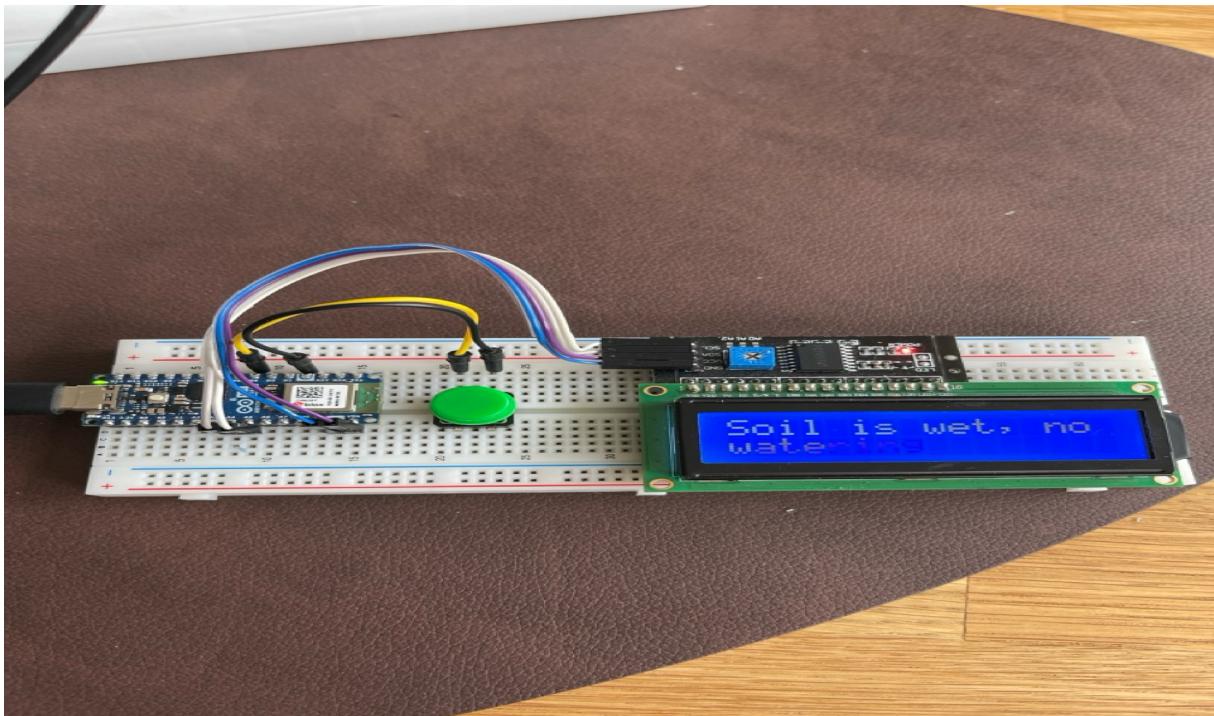
Bilagor



Inuti chassis och hela produkten.



Esp32 nano kopplade till komponenterna



Lcd-display med en knapp



Vätskenivåsensor