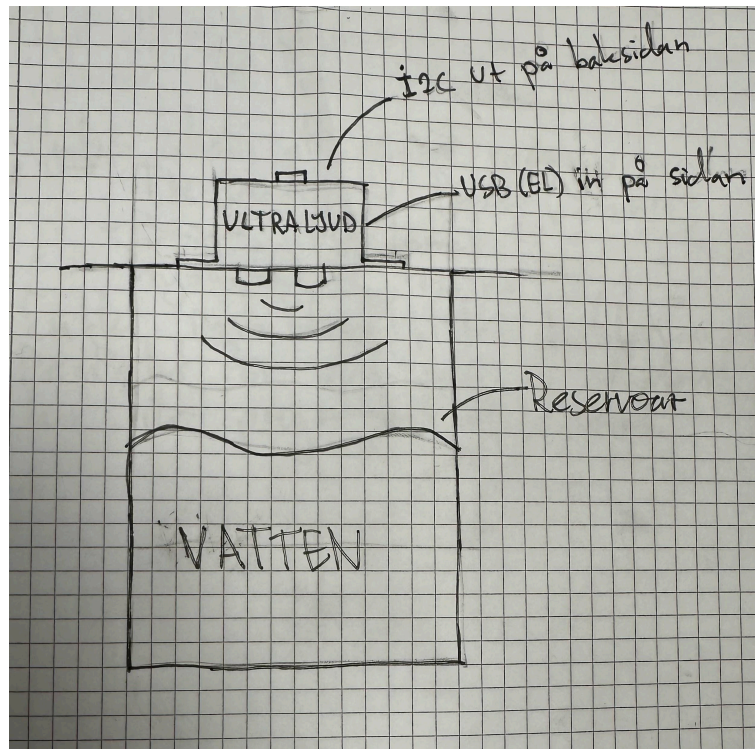


# Vattennivåsensor (Analog) - Dokumentation



Figur 1 - Ritning över konceptet

Vår vattennivåsensor använder sig av en ultraljudssensor som skickar ut vågor som sedan studsar mot vattenytan. Eftersom ultraljudssensorn mäter avståndet mellan sensorn och vattenytan kan vi med hjälp av enkel matematik räkna ut vattennivån i reservoaren. Detta genom att ta avståndet från ultraljudssensorn till botten av reservoaren, subtraherat med det avstånd som sensorn mäter upp. Denna data skickas sedan med hjälp av I2C-protokollet till en master-arduino, som sedan använder informationen genom att exempelvis visa vattennivån på en LCD-skärm.

## Komponenter

1 arduino micro :

[https://store.arduino.cc/en-se/products/arduino-micro?srltid=AfmBOoq6ut6BAKGfV4DVKaZKa\\_LHpmBvgw-AfOK-OMMMvDHNoMQhuob](https://store.arduino.cc/en-se/products/arduino-micro?srltid=AfmBOoq6ut6BAKGfV4DVKaZKa_LHpmBvgw-AfOK-OMMMvDHNoMQhuob)

1 arduino ultraljud sensor :

<https://www.electrokit.com/avstandsmatare-ultraljud-hc-sr04-2-400cm>

# Kod

Det här är koden som kopplar samman Arduinon och vår sensor, och har som uppgift att hämta och skriva ut datan som sensorn tar upp, genom två pins på sensorn. Koden säger till sensorn att sända ut en puls och vänta på svar. Sedan beräknas sträckan beroende på hur lång tid det tog för signalen att återvända. Datat sparas i centimeter, och skickas till mästern Arduinon. Det är onödigt att skicka en float med i2c eftersom dessa kräver många bytes av info. Därför gör vi om det till en integer med en decimal noggrannhet innan vi skickar datan.

**OBS! Vi använder oss av arduinos i2c kompatibla modul wire. Nedan finns en länk till dokumentation för denna modul.**

<https://docs.arduino.cc/language-reference/en/functions/communication/wire/>

```
#include <Wire.h>

const int TRIG_PIN = 9;
const int ECHO_PIN = 10;
const float CM_TO_BOTTOM = 24;

void setup()
{
    pinMode(ECHO_PIN, INPUT);
    pinMode(TRIG_PIN, OUTPUT);

    digitalWrite(TRIG_PIN, LOW);

    Wire.begin(9);
    Wire.onRequest(getDistance);

    Serial.begin(9600);
}

void loop()
{
    delay(100);
}

void getDistance()
{
    float timing = 0.0;
    float distance = 0.0;
    float waterLevel = 0.0;
```

```

digitalWrite(TRIG_PIN, LOW);
delay(2);

digitalWrite(TRIG_PIN, HIGH);
delay(10);
digitalWrite(TRIG_PIN, LOW);

timing = pulseIn(ECHO_PIN, HIGH);
distance = (timing * 0.034) / 2;

Serial.println(distance);

waterLevel = CM_TO_BOTTOM - distance;

Serial.print("Water Level: ");
Serial.print(waterLevel);
Serial.println("cm | ");

int c = int(waterLevel * 10.0f);

Wire.write((const char*)&c, 4);
Serial.print(c);
Serial.print(" ");
Serial.println(waterLevel);
}

```

Nedan kommer koden som krävs för att ta emot våran data:

```

while(Wire.available())
{
    int v;

    byte b[4];

    b[0] = Wire.read();

    b[1] = Wire.read();

    b[2] = Wire.read();

    b[3] = Wire.read();

    memcpy((void*)&v, b, 4);
}

```

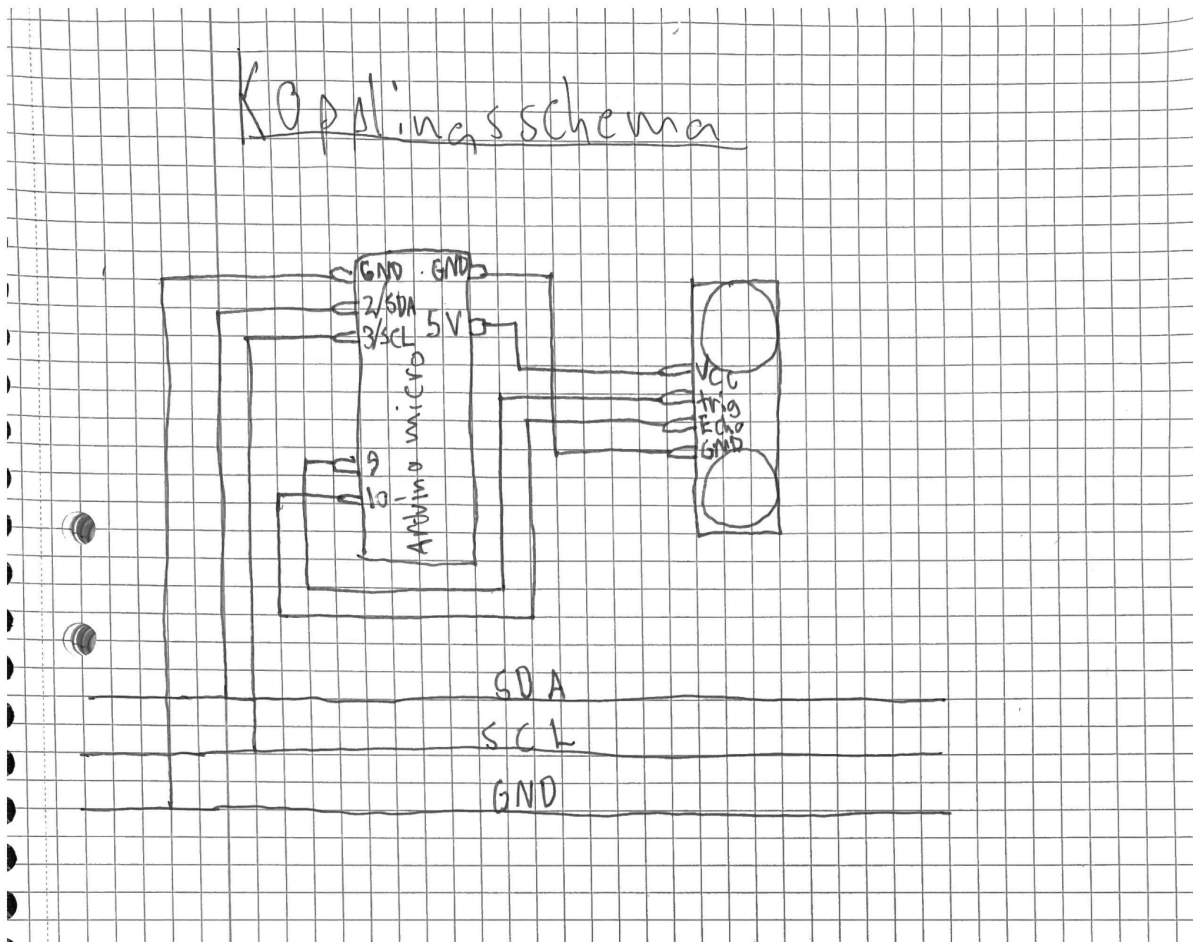
```

Serial.println(((float)v) / 10.0f);

}

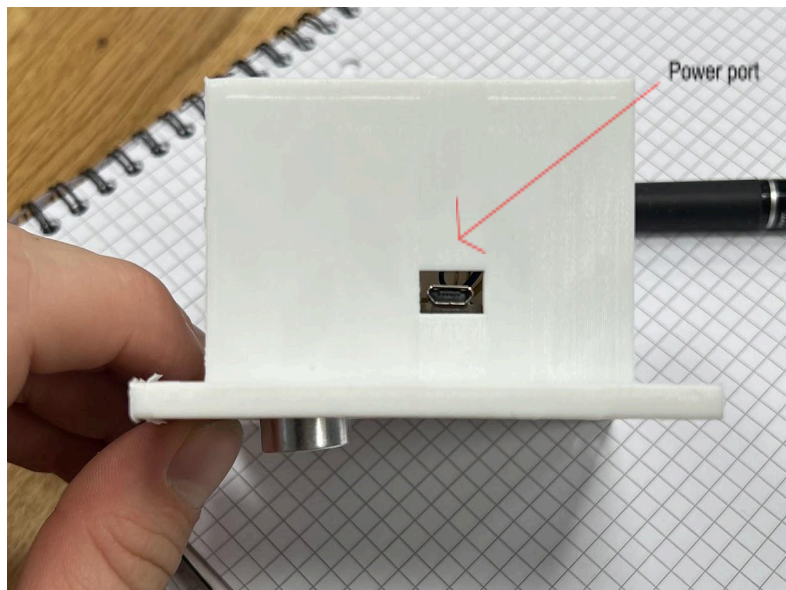
```

## Kopplingsschema

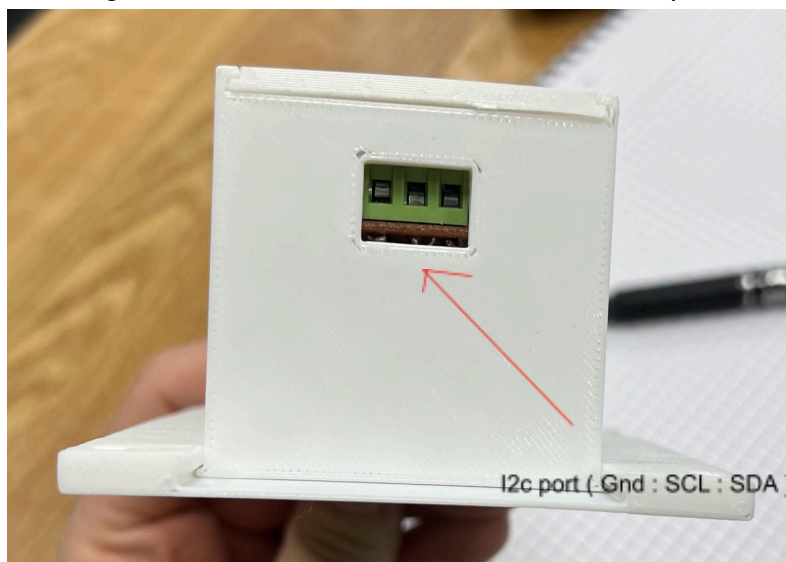


Figur 2 - Kopplingsschema

## Info: Behållare



Figur 3 - Behållare från sidan - USB/POWER port



Figur 4 - Behållare framifrån - i2c port

### Länk till Fusion fil:

Platta: <https://a360.co/40oi8Jj>

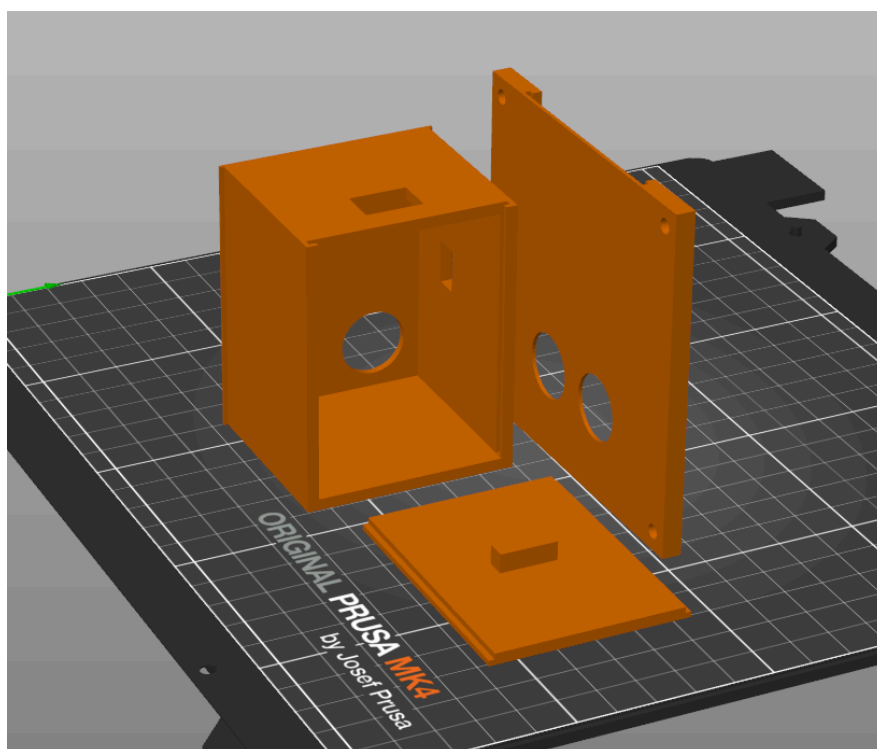
Låda: <https://a360.co/3DJRku8>

Lock: <https://a360.co/4fMhRV2>

### Lösen:

vattensensor2024

**Obs! När du printar är det viktigt att lådan och plattan står på högkant. Detta är för att underlätta relingarna och minska svåra overhangs. Kan också rekommendera organic supports.**



*Figur 5 - exempel på hur det ska se ut när du printar*