

# **Tekniskrapport**

## **Tilstand - Vår 2024**

Ai-Vy Hoang Pham

Asya Saygin

Iman Amara

Martin Nygard

Sarah Mehmedovic

Prosjektoppgave i IN1060 Institutt for Informatikk



**UNIVERSITETET  
I OSLO**

28.05.2024

Innholdsfortegnelse

Tekniskrapport.....	1
Tilstand - Vår 2024.....	1
Innholdsfortegnelse .....	1
1 Innledning.....	3
1.2 Mål og forklaring.....	3
2 Den tekniske løsningen .....	3
2.1 Komponenter .....	3
2.2 Sammensetting av løsningen.....	4
3 Arduino kode og forklaringer.....	6
3.1 Første Arduino kode.....	6
3.2 Oppdatert Arduino kode med knappfunksjon.....	7
4 Avslutning.....	8

# 1 Innledning

## 1.2 Mål og forklaring

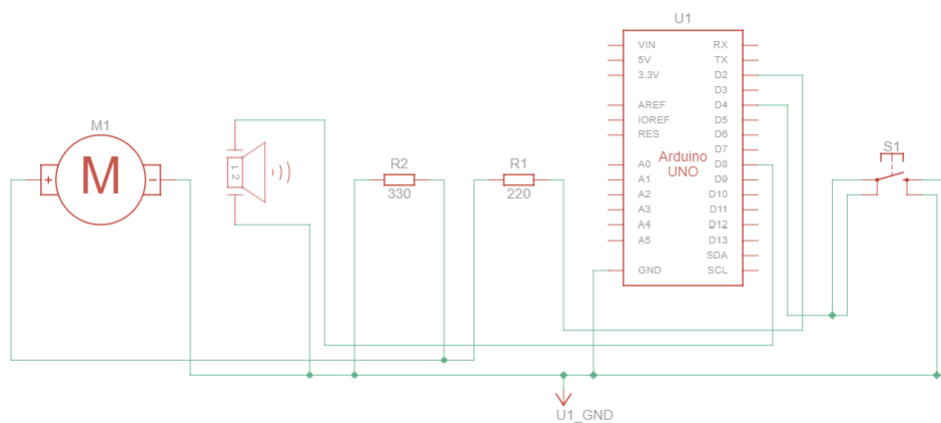
Grupped medlemmer: Martin, Aivy, Iman, Sarah og Asya

Målet med prosjektet vårt er at “beep-test armbåndet” skal vibrere og lage lyd samtidig, noen sekunder før neste intervall. Koden og implementeringen av Arduino må derfor følge tiden til en beep-test for hver runde, slik at vibrasjonen og lyden blir implementert riktig. Alt dette skal bidra til at brukeren har mulighet til å vite når hen skal gi sin siste innsjurt, eller om hen ikke rekker en til runde.

## 2 Den tekniske løsningen

### 2.1 Komponenter

Vi bestilte vibrasjonsmotor på nett. Under konstruksjonen av kretsen dukket det opp et problem. Vibrasjonsmotoren tåler bare 3V spenning. Det er et problem fordi digitale pins gir 5V. For å løse dette problemet prøvde vi oss frem med flere kretser med bruk av voltmeter. Voltmeteret ga i vårt første forsøk 4V som var for høyt. Vi kom til slutt fram til en krets som ikke gikk over 3V med bruk av 330 og 220 motstandere:



Komponentliste:

1x Piezo
1x Vibrasjonsmotor
1x 9V batteri

1x 220 ohms motstand
-------------------------

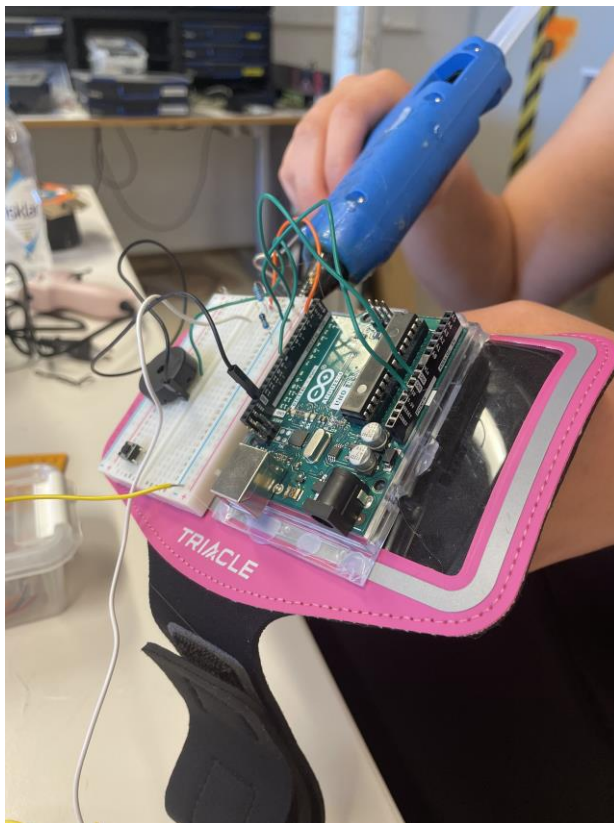
1x Arduino uno
----------------

1x 330 ohms motstand
-------------------------

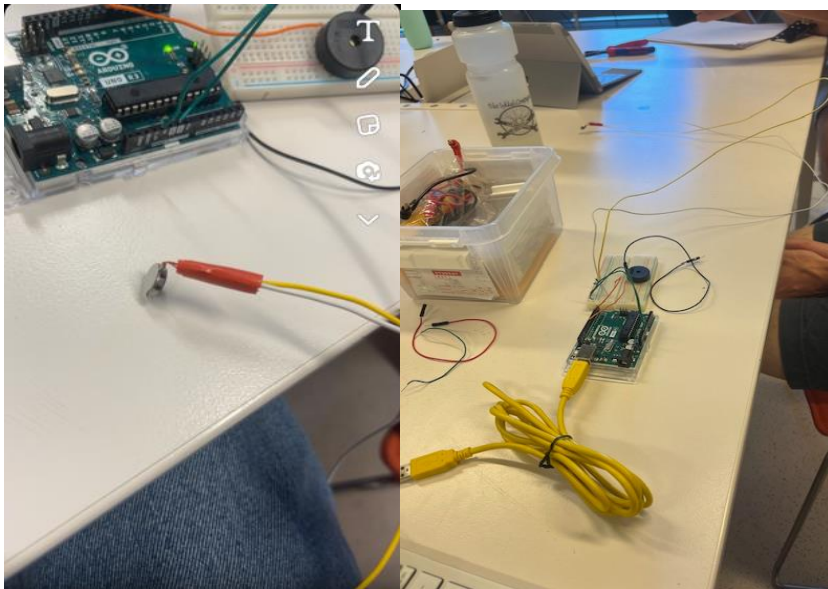
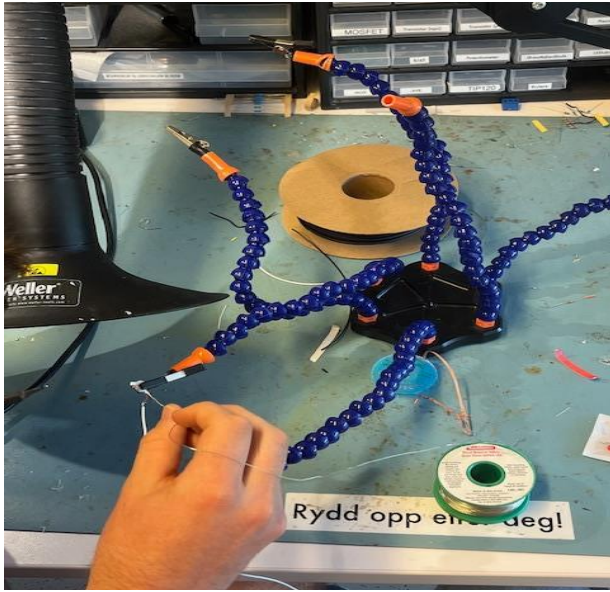
1x Knapp
----------

## 2.2 Sammensetting av løsningen

For ha på armbåndet og kunne teste det ut måtte vi sette Arduinoen nærme slik at alt setter seg godt sammen. Vi gjorde det ved å lime Arduinoen til en mobilholder som er egnet til armen. Ledningene måtte derfor strekke seg opptil håndleddet slik at brukeren kan løpe uten at den faller sammen. Vibrasjonsmotoren med ledningen skulle deretter bli limt til et armbånd:



Ledningene vi hadde var litt korte. Vi måtte derfor lodde dem sammen med lengre ledninger for at de skulle nå fram til håndleddet. Deretter satte vi på vibrasjonskomponenten til ledningen og armbåndet:



## 3 Arduino kode og forklaringer

### 3.1 Første Arduino kode

```
const int buzzerPin = 8;      // Pin for buzzer
const int vibrationPin = 2;   // Pin for vibrasjonsmotor
unsigned long previousMillis = 0; // Lagre tidspunktet for forrige handling
unsigned long interval = 15000; // Start med 15 sekunders intervall

void setup() {
  pinMode(buzzerPin, OUTPUT); // Sett buzzer-pinnen som utgang
  pinMode(vibrationPin, OUTPUT); // Sett vibrasjonsmotor-pinnen som utgang
}

void loop() {
  unsigned long currentMillis = millis(); // Hent nåværende tid

  // Sjekk om nåværende tid minus forrige tid er større enn eller lik intervallet
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis; // Lagre tidspunktet for forrige handling

    // Start vibrasjonsmotoren 4 sekunder før buzzer
    digitalWrite(vibrationPin, HIGH);
    delay(4000); // Vibrasjonens varighet (4 sekunder)
    digitalWrite(vibrationPin, LOW);

    // Aktiver buzzer
    tone(buzzerPin, 1000); // Send 1KHz lydsignal
    delay(1000);           // i 1 sekund
    noTone(buzzerPin);    // Stopp lyden

    // Reduser intervallet for neste pip
    if (interval > 500) {
      interval -= 500; // Reduser intervallet med 0,5 sekunder (500 millisekunder)
    } else {
      // Forhindre at intervallet blir null eller negativt
      interval = 500; // Sett et minimumsintervall på 0,5 sekunder
    }
  }
}
```

Vi trengte å lage en kode som kunne både styre vibrasjonskomponenten samtidig som piezo komponenten skulle pipe, og dette måtte da kunne skje sammenhengende. Vi lagde altså en kode som med millis piper hvert 15 sekund, og bruker 0,5 sekunder mindre for hver gang og vibrerer 4 sekunder før det skal pipe. Noen problemer vi møtte underveis var at piezo vil ikke pipe etter 4 sekunder med vibrasjon selvom koden egentlig skulle funke. Vi leste oss opp mer på hvordan bruke arduino buzzer og byttet heller om på koden at vi brukte tone(buzzer, 1000) som da sendte lyd signal og brukte etter det delay(1000) som gjorde at pipet varte i 1 sekund. Deretter brukte vi noTone(buzzer) slik at pipet skulle slutte. Dette fungerte da veldig bra og piezo pipet etter 4 sekunder med vibrasjon som vi ville.

Vi har også en if setning i koden som reduserer intervallet for hvert pip. Dette gjør slik at når brukeren har løpt en runde, vil det da bli mindre tid for dem før neste pip kommer og de må derfor være raskere. Vibrasjonen vil likevel vibrere 4 sekunder før hvert pip, som vil hjelpe brukeren å vite at de må løpe raskere fordi pipet kommer snart.

### 3.2 Oppdatert Arduino kode med knappfunksjon

```
const int piezoPin = 8; // Definerer pin-nummeret piezo-brikken er koblet til
const int motorPin = 2; // Definerer pin-nummeret vibrasjonsmotoren er koblet til
const int buttonPin = 4; // Definerer pin-nummeret knappen er koblet til

unsigned long previousMillis = 0; // Variabel for å lagre tiden sist beep ble spilt av
const long initialInterval = 15000; // Initialt interval mellom hver beep (i millisekunder)
long interval = initialInterval; // Variabel for å lagre intervallet, starter med det initielle intervallet
const long vibrationDuration = 3000; // Varigheten av vibrasjonen (i millisekunder)

bool programRunning = false; // Variabel for å holde rede på om programmet kjører

void setup() {
  pinMode(piezoPin, OUTPUT); // Setter pin-nummeret som en output for piezo
  pinMode(motorPin, OUTPUT); // Setter pin-nummeret som en output for vibrasjonsmotor
  pinMode(buttonPin, INPUT_PULLUP); // Setter pin-nummeret som en input for knappen med intern pullup-motstand
}

void loop() {
  // Sjekker om knappen er trykket
  if (digitalRead(buttonPin) == LOW) {
    delay(50); // Debouncing delay
    if (digitalRead(buttonPin) == LOW) { // Sjekker igjen etter debouncing
      programRunning = true; // Starter programmet
      previousMillis = millis(); // Nullstiller tiden riktig ved start
      interval = initialInterval; // Nullstiller intervallet
      while (digitalRead(buttonPin) == LOW) { // Venter til knappen slippes
        delay(10); // Liten forsinkelse for å unngå rask trykk/slipp
      }
    }
  }
  // Kjør programmet hvis det er aktivert
  if (programRunning) {
```

```

    unsigned long currentMillis = millis(); // Henter gjeldende tid
    if (currentMillis - previousMillis >= interval) { // Sjekker om det er tid
for en ny beep
        // Oppdaterer tiden sist beep ble spilt av
        previousMillis = currentMillis;

        // Starter vibrasjonen
        digitalWrite(motorPin, HIGH); // Start vibrasjonsmotoren
        delay(vibrationDuration); // Vibrasjonsvarighet
        digitalWrite(motorPin, LOW);

        // Spiller pipelyden
        tone(piezoPin, 1000); // Spiller en tone på 1000 Hz for beep
        delay(1000); // Vent en kort stund
        noTone(piezoPin); // Slutt å spille tonen
        // Reduserer intervallet med 0.3 sekunder for neste beep
        interval -= 300;

        // Sørger for at intervallet ikke blir mindre enn vibrasjonens varighet
+ beep
        if (interval < vibrationDuration + 1000) {
            interval = vibrationDuration + 1000;
        }
    }
}
}
}
}

```

#### 4 Avslutning

Gjennom prosessen har vi fått til å lage et armbånd som skal varsle/påminne brukeren noen sekunder før det neste “beepet” forekommer. Det har vært utfordrende underveis å sette sammen alle delene og å få alt til å fungere. Vi fikk til slutt til å implementere alle funksjonene slik at løsningen begynner å fungere når bruker trykker på “på-knappen”. Dette har vi fått til gjennom flere iterasjoner med flere tilbakemeldinger og korreksjoner.