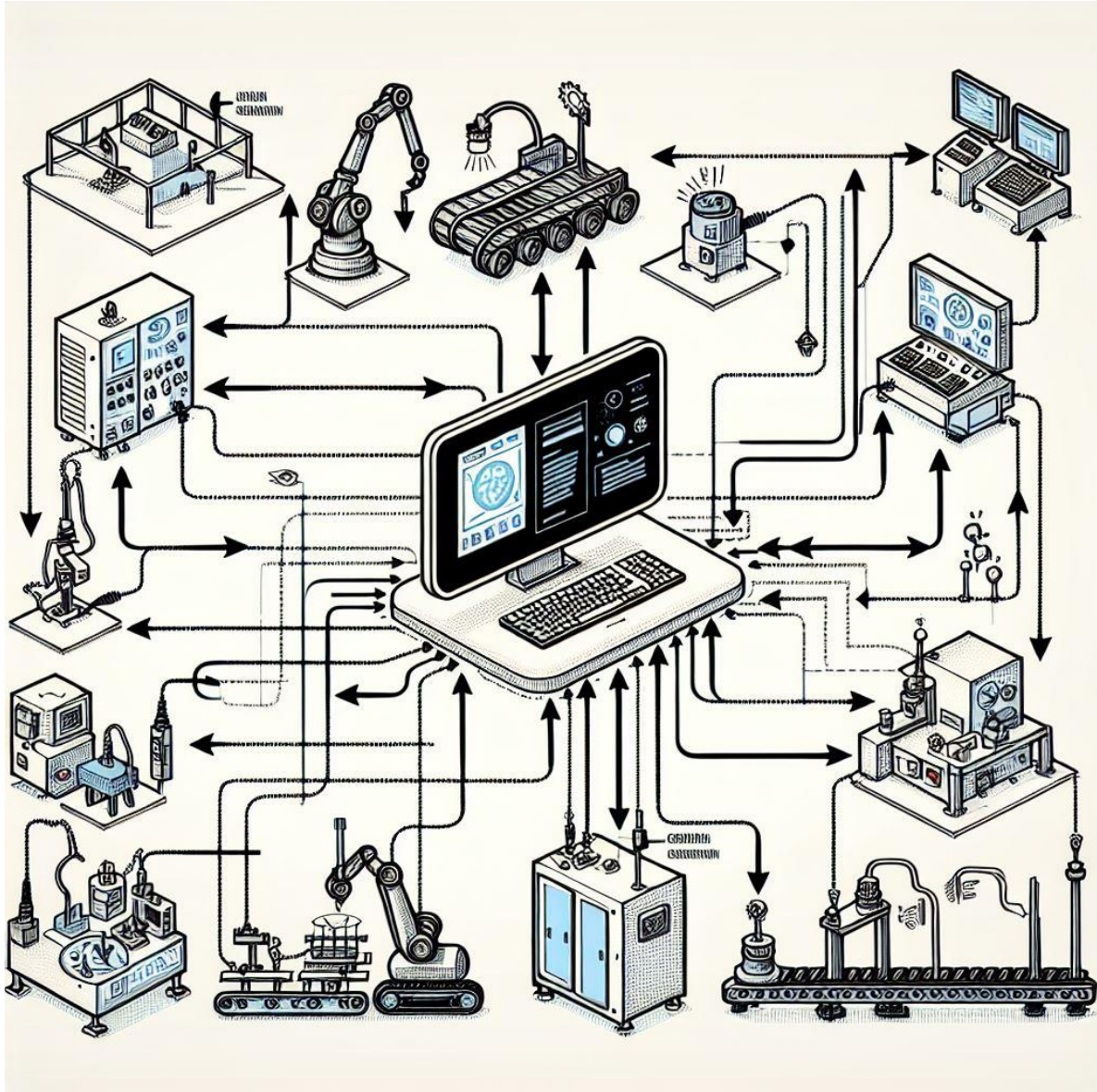


## Obligatorisk oppgave 3: Overvåking av Vibrasjon og Temperatur i Maskiner



### Introduksjon

Denne prosjektoppgaven gir dere muligheten til å designe og implementere et system for å overvåke vibrasjoner og temperatur i maskiner.

Ved hjelp av en Raspberry Pi Pico med tilkoblede TMP102 (temperatursensor) og ADXL345 (akselerometer), skal data samles inn, behandles, visualiseres og lagres i en MySQL-database. Hver gruppe får utdelt en ferdigprogrammert mikrokontroller med sensorer.

Det antas at KI benyttes for raskere programmering, men det forventes at dere setter dere inn i og forstår koden.

Dere kan jobbe individuelt eller i grupper på opptil 4 medlemmer.

## Mål

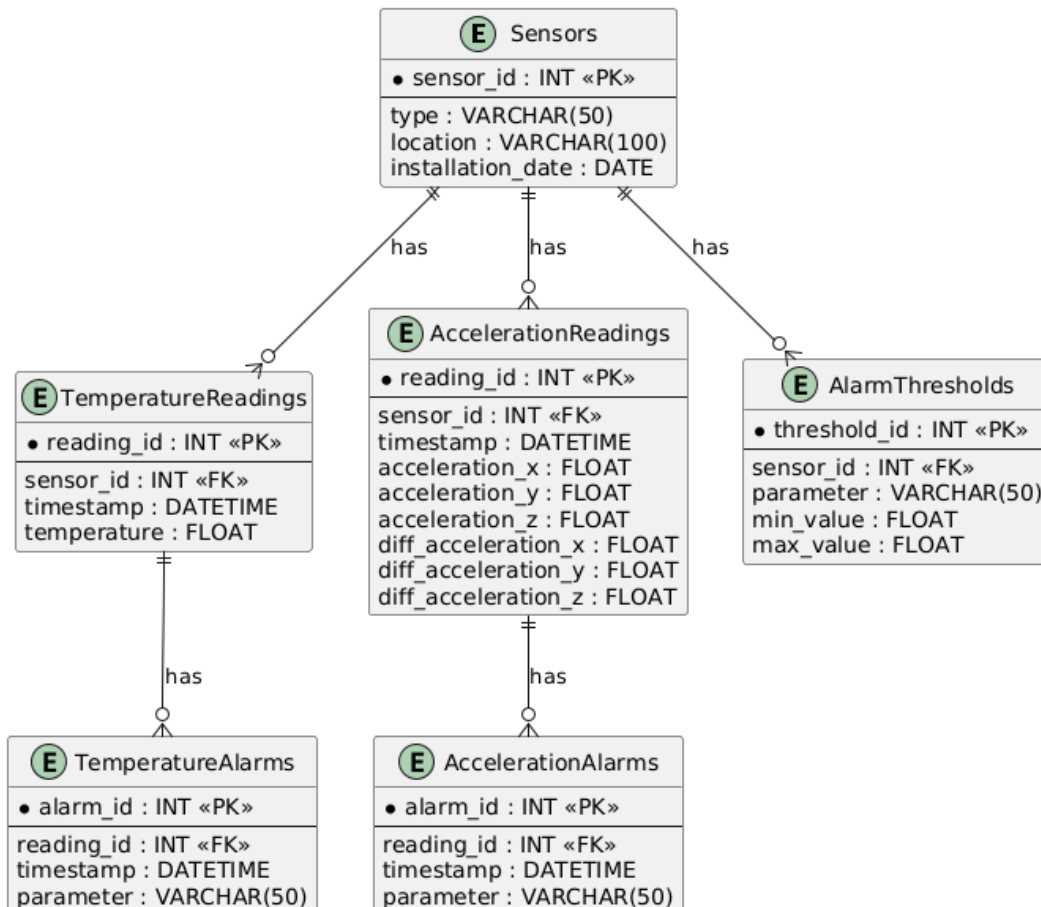
1. Datainnsamling og Lagring: Utvikle Python-programmer for å motta data fra Raspberry Pi Pico via seriell kommunikasjon, og lagre dataene i en MySQL-database.
2. Databehandling og Visualisering: Behandle og visualisere dataene i sanntid ved hjelp av Python med et grafisk brukergrensesnitt (GUI).
3. Konfigurasjon av Sensorer: Utvikle et GUI-program for å registrere nye sensorer og omkonfigurere eksisterende sensorer.
4. Alarmhåndtering: Implementere et alarmsystem som utløses når målte verdier overstiger definerte grenser, og vis det i GUI.

## Databasekrav

Dere skal opprette databasen, som må inneholde minimum følgende informasjon:

- Sensortyper
- Sensorlokasjoner
- Installasjonsdatoer
- Temperaturavlesninger
- Akselerasjonsavlesninger (x, y, z)
- Alarmgrenser
- Utløste alarmer

Dere kan bygge strukturen selv, men vi inkluderer et ER-diagram som et eksempel på en struktur.



## Oppgaver

### Del 1: Opprettelse av Database

#### Databaseoppsett:

- Planlegg og tegn et ER-diagram for deres database
- Opprett en MySQL-database med tabeller i henhold til ER -diagrammet.
- Sørg for at alle nødvendige felter og relasjoner er inkludert.

### Del 2: GUI-program for Sensorregistrering og Konfigurasjon

#### Programfunksjonalitet:

- Utvikle et GUI-program for å registrere nye sensorer og omkonfigurere eksisterende sensorer.
- Programmet må kunne laste opp informasjon til databasen, samt sende konfigurasjonsdata til mikrokontrolleren.
- Konfigurasjonsdata til mikrokontrolleren skal ha følgende format:

```
{
  "SensorConfiguration": {
    "TemperatureSensor": {
      "Type": "EksempelNavn1",
      "Sensor_id": id-nummer
    },
    "Accelerometer": {
      "Type": "EksempelNavn2",
      "Sensor_id": id-nummer
    }
  }
}
```

### Del 3: GUI-program for Datainnsamling og Visualisering

#### Datainnsamling:

- Utvikle et GUI-program som kommuniserer med mikrokontrolleren for å hente lesinger.
- Programmet skal kunne starte og stoppe avlesing av data på mikrokontrolleren ved å sende følgende kommandoer:

```
{"Command": "START"}
{"Command": "STOP"}
```

- Mikrokontrolleren sender data i følgende formater:

<pre>{   "temperature":   {     "temperature":25.4375,     "sensor_id":1   },   "acceleration":   {     "x":0.627626,     "y":-0.392266,     "z":8.943665,     "sensor_id":2   } }</pre>	<pre>{   "acceleration":   {     "x":0.745305,     "y":-0.392266,     "z":9.139798,     "sensor_id":2   } }</pre>
--	---

#### *Dataopplasting til Database:*

- Mottatte data fra mikrokontrolleren skal lastes opp til MySQL-databasen.
- Sørg for at hver avlesning registreres med korrekt sensor\_id og tidsstempel.

#### *Nåverdier og Alarmstatus:*

- GUI-programmet skal vise nåværende verdier for temperatur og akselerasjon.
- Programmet skal også vise om en sensor er i alarmtilstand basert på forhåndsdefinerte alarmgrenser.

#### *Visning av Historisk Data:*

- Implementer funksjonalitet for å vise historiske data for temperatur og akselerasjon.
- Bruk et grafisk bibliotek som matplotlib eller plotly for å visualisere dataene i grafer.

## Verktøy og Biblioteker

### GUI-bibliotek:

- Dere kan bruke enten TkInter eller PyQt for å utvikle GUI.
  - Merk: Ved bruk av PyQt må dere bruke PyMySQL i stedet for mysql-connector-python for databasetilkobling. Dette biblioteket har så å si samme syntax som mysql-connector-python, men er kompatibelt med pyQT.

## Rapporter og Video

### Rapport:

- En kort rapport som beskriver utførelsen av prosjektet, valg som er gjort, og metoder som er brukt.
- Rapporten skal også inneholde en refleksjon over utfordringer og hvordan de ble løst.
- Rapporten skal inkludere en seksjon som beskriver hvorfor innsamling av vibrasjons- og temperaturdata er nyttig.
- Pythonfiler legges ved i en zip-fil.

## Video:

- En kort video som demonstrerer applikasjonene i bruk.
- Videoen skal vise registrering og konfigurasjon av sensorer, samt innsamling og visning av data.

## Viktige Tips

- Test individuelle komponenter: Test hver del av koden individuelt før du integrerer dem for å sikre at hver komponent fungerer som forventet.
- Versjonskontroll: Bruk versjonskontroll (f.eks. Git) for å holde styr på endringer i koden. Dette vil gjøre det enklere å samarbeide i grupper og å gå tilbake til tidligere versjoner hvis noe går galt.
- Dokumentasjon: Sørg for å dokumentere koden godt, slik at andre kan forstå den. Det er ikke nødvendig å kommentere innlysende kode, som f.eks:

```
#Prints an error message to the user  
print("ERROR!!!")
```

- Bruk av Biblioteker: Velg riktig bibliotek for GUI og databasetilkobling avhengig av ditt valg (TkInter med mysql-connector-python eller PyQt med PyMySQL).
- Feilhåndtering: Implementer robust feilhåndtering for å sikre at programmet kan håndtere uventede situasjoner uten å krasje.
- Brukeropplevelse: Design GUI med tanke på brukeropplevelse. Sørg for at grensesnittet er intuitivt og enkelt å bruke.
- Mikrokontroller: Sørg for at sensorkortet er koblet riktig før mikrokontrolleren kobles til PCen. Husk at COM-porten vil variere fra PC til PC og mikrokontroller til mikrokontroller. Brukeren må derfor kunne velge hvilke COM-port programmet skal bruke. Baudrate = 9600.
- Det er mulig å endre hvor ofte data samles av mikrokontrolleren. Den aksepterer heltall fra 1-10 med følgende kommando:

```
{"GatherFreq" : N}
```

- Det er mulig å se hvilke data som er lagret på mikrokontrolleren med følgende kommando:

```
{"Command": "RETURN_DATA"}
```

## Oppsummering av Oppgaven

### Opprettelse av Database:

- Opprett en MySQL-database med nødvendige tabeller og relasjoner.

### GUI-program for Sensorregistrering og Konfigurasjon:

- Design og implementer et GUI-program for registrering og omkonfigurering av sensorer.
- Sørg for at programmet kan kommunisere med mikrokontrolleren for å sende konfigurasjonsdata.

## GUI-program for Datainnsamling og Visualisering:

- Design og implementer et GUI-program for datainnsamling og visualisering.
- Implementer funksjonalitet for å starte og stoppe datainnsamling, laste opp data til databasen, og visualisere sanntids- og historiske data.

## Rapport og Video:

- Skriv en rapport som beskriver prosjektet, metodene som er brukt, og hvorfor innsamling av vibrasjons- og temperaturdata er nyttig.
- Lag en kort video som demonstrerer applikasjonene i bruk.
- Legg ved python-filer
- Legg ved SQL-kode dersom dere ikke brukte python til å opprette databasen

## Eksempel på Rapportstruktur

- Introduksjon:
  - Beskrivelse av prosjektet og dets mål.
- Database Design:
  - Beskrivelse av databasestrukturen med ER-diagram.
  - Forklaring av tabeller og deres relasjoner.
- GUI-program for Sensorregistrering og Konfigurasjon:
  - Beskrivelse av programmet og dets funksjonalitet.
  - Skjermbilder av GUI.
- GUI-program for Datainnsamling og Visualisering:
  - Beskrivelse av programmet og dets funksjonalitet.
  - Skjermbilder av GUI.
  - Eksempel på data som blir samlet inn og visualisert.
- Teknologivalg:
  - Begrunnelse for valgte biblioteker og verktøy.
- Utfordringer og Løsninger:
  - Beskrivelse av eventuelle utfordringer som ble møtt underveis og hvordan de ble løst.
- Nytten av Datainnsamling:
  - Forklaring av hvorfor innsamling av vibrasjons- og temperaturdata er nyttig.
  - Eksempler på praktiske applikasjoner og fordeler ved å overvåke maskindata.
- Konklusjon:
  - Oppsummering av hva som ble oppnådd i prosjektet.
  - Refleksjon over læringsutbyttet.

## Eksempel på Videoinnhold

- Introduksjon:
  - Kort introduksjon
- Sensorregistrering og Konfigurasjon:
  - Demonstrasjon av hvordan nye sensorer registreres og konfigureres i GUI-programmet.
- Datainnsamling og Visualisering:
  - Demonstrasjon av hvordan data samles inn fra mikrokontrolleren, lastes opp til databasen, og visualiseres i GUI-programmet.
  - Visning av sanntidsdata og historiske data i grafer.
  - Demonstrasjon av alarmfunksjonalitet.