

Inventas

Inventas er et helnorsk selskap som har jobbet med produktutvikling og problemløsning i over 25 år. Gjennom å kombinere kunnskap fra erfarne fagpersoner og kreative medarbeidere, tilbyr vi tjenester som har som mål å støtte våre kunders behov. Vi har kontorer i ulike deler av landet, inkludert Oslo, Kristiansand, Trondheim, Molde, Bergen, Stavanger og Asker.

Vi arbeider med et bredt spekter av kunder, fra de som jobber med kommersialisering av forskning, til oppstartsselskaper og etablerte teknologiselskaper. Uavhengig av behovet, enten det er total produktutvikling eller spesifikk konsultasjon, søker vi å tilby relevant støtte. Blant våre tjenester finner man digital og fysisk produktutvikling, elektronikk, kybernetikk, strategisk rådgivning og med mer.

Ved våre kontorer i Oslo og Asker, har vi ansatte med kompetanse innen mekanikk, produktdesign, elektronikk, FPGA/ASIC og kybernetikk. Vår ASIC/FPGA-avdeling har et solid omdømme og bruker vårt eget testrammeverk, UVVM, som er i bruk av over 20% av alle FPGA-designere globalt. I tillegg har vi laget et verktøy kalt HDLRegression for effektiv HDL regresjonskjøring. For å møte den voksende etterspørselen etter IoT-løsninger, har vi også utviklet en egen plattform, Multiply, skreddersydd for prototyping av slike systemer. Samarbeidet vårt strekker seg til ulike sektorer, deriblant romfartsindustrien med aktører som ESA og NASA.

Multiply FPGA

Multiply-plattformen, utviklet av Inventas, fungerer som en "kickstarter" for trygg og sikker oppkobling av produkter. Den baserer seg på nRF9160-plattformen fra Nordic Semiconductors, og tilbyr integrert LTE-M/NB-IoT modem og GPS, i tillegg til en rekke grensesnitt for tilkobling av eksterne sensorer og enheter, inkludert 12-bit ADC, SPI, I2C og UART. Multiply er utstyrt med feather connector, og har muligheter for direkte tilkobling mot FPGA. Med et ekstremt lavt strømtrekk (50-100 μ A i dvalemodus) kan systemet støtte opptil 10 års batteridrift, avhengig av bruksmønster.



Oppgaver

Prediktivt vedlikehold:

I industrien har prediktivt vedlikehold, med hjelp av AI/ML, blitt interessant for å sikre effektiv drift. Ved å benytte AI/ML-metoder kan man forutsi vedlikeholdsbehov i sanntid, noe som bidrar til å forhindre unødvendig nedetid og redusere kostnader.

Problemstilling:

Detektere tidlige feil i kulelagre ved hjelp av vibrasjonsmåling. Implementasjon på FPGA

Aktuelle datasett:

- [HUST bearing: a practical dataset for ball bearing fault diagnosis](#)
- [NASA Bearing Dataset](#)
- [Case Western Reserve University Bearing Data Center Website](#)

Oppgaver:

1. Trene en ML-modell (Tensorflow/PyTorch) som kan gjenkjenne kulelagre med feil fra eksisterende datasett

2. Konvertere ML-modellen til et format som kan kjøres på en FPGA (hvilke kompromisser/optimaliseringer må gjøres i denne prosessen? Hvordan påvirker samplingsfrekvens og ADC-oppløsning ytelsen? Hvilke begrensninger har FPGA i forhold til en floating-point basert datamaskin?)
3. Sammenligne ytelse på original modell og FPGA-modell
4. Implementer ML-modell på et FPGA utviklingskort med ADC-kort (f.eks. type PMOD) for innsamling av vibrasjonssignal. Vurder ulike arkitekturer for implementering: Koble ADC data direkte til ML-modell i FPGA-logikk vs. la en CPU hente data fra ADC og så videresende data til ML-modell i FPGA-logikk.
5. Sett opp en praktisk test hvor en aksling med kulelager roteres av en motor. Vurder sensortype og plassering av sensor. Lag ulike defekter på kulelageret. Test og evaluer ML-modellen på sanntidsdata.
6. Samle inn nye data med det praktiske oppsettet, og optimaliser modellen ytterligere ved å trene den med ekstra data. Test og evaluer den oppdaterte ML-modellen på sanntidsdata.

Datakomprimering og -overføring:

I en tid hvor enhetenes effektivitet og kostnadsbesparelse er i fokus, blir teknikker for datakomprimering, spesifikt designet for Edge-enheter og IoT-applikasjoner, stadig mer relevante.

Problemstilling: Komprimere ECG data fra en IoT sensor som skal brukes til inferens i en skybasert ML-modell.

Aktuelle datasett:

- [MIT-BIH Arrhythmia Database](#)
- [Apnea-ECG Database](#)

Oppgave:

1. Tren en ML-modell (Tensorflow/PyTorch) med full oppløsning på ECG-data som klassifiserer data i henhold til det valgte datasettet
2. Hvilke komprimeringsteknikker reduserer best dataoverføringskostnader samtidig som kritisk informasjon bevares? Komprimeringsalgoritmen må kunne kjøre på en typisk IoT-chip (som nRF-serien fra Nordic Semiconductor). Sammenlign nøyaktighet til ML-modellen med/uten datakompresjon

3. Implementer kompresjonsalgoritmen på et IoT dev.kit. Mål effektforbruk/båndbredde med og uten kompresjon.
4. Utforsk data-transformasjon på ECG-data (f.eks. FFT, wavelet). Tren en ML-modell som tar transformerte data som input i stedet for rådata. Sammenlign nøyaktigheten med modellen fra punkt 1.
5. Implementer transformasjonsalgoritmen på IoT dev-kitet. Mål effektforbruk/båndbredde med og uten transformasjon. Sammenlign med punkt 3.

Forkunnskaper:

- Kjennskap til IoT/ML
- Rust/C programmering
- FPGA og VHDL