

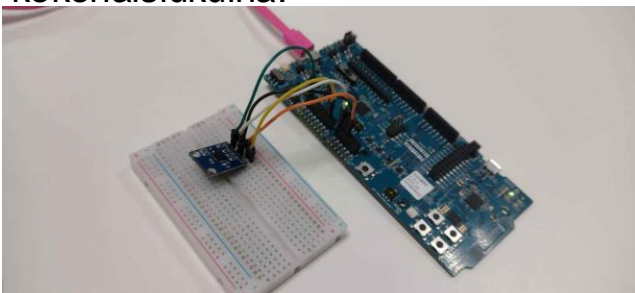
Tietoliikenteensovellusprojekti

Juuso Toratti/Teemu Niemi

Tietotekniikan tutkinto-ohjelma, ohjelmistokehitys / laite- ja tuotesuunnittelu

Projektin tarkoitus

Projektin perimmäisenä tarkoituksena oli oppia lisää Linux-palvelimista, Bluetoothista, tietokantojen muokkaamisesta, Python-neuroverkoista ja mikrokontrollereista. Projektissa käytettiin NRF5340DK-mikrokontrolleria, joka oli kytketty kiihtyvyysanturiin. Kiihtyvyysanturi mittaa kiihtyvyyttä x-, y- ja z-akseleiden suunnassa ja palauttaa näiden kiihtyvyysarvot kokonaislukuina.



Kuva 1. Kiihtyvyysanturi ja nRF5340DK mikrokontrolleri

Viikot 1-7

Projekti käynnistettiin luomalla GitHub-repositorio, arkkitehtuurikaavio ja alustamalla Kanban-taulu projektinhallintaa varten. Raspberry Pi -laitteen käyttöönotto suoritettiin onnistuneesti ilman merkittäviä ongelmia. Toisella viikolla projekti eteni siihen vaiheeseen, että kiihtyvyysanturi konfiguroitiin lähettämään XYZ-dataa kehitysalustan Bluetooth-toiminnallisuuden avulla älypuhelimeen ladattuun nRF Connect- sovellukseen. Kolmannella viikolla Linux-palvelimelle asennettiin ja otettiin käyttöön Netfilter-palomuurisovellus esimerkiskriptin avulla.

Raspberry Pi -laitteelle luotiin Python-ohjelma nano-tekstieditorilla. Ohjelma vastaanottaa dataa kehitysalustalta Bluetoothin välityksellä, hyödyntäen Bleak-kirjastoa. Vastaanotettu data lähetettiin MySQL-tietokantaan, joka oli valmiiksi määritelty 172.20.241.9-palvelimella. Neljäs viikko: Visual Studio Codeen asennettiin Thunder Client -lisäosa REST-API:en testaamiseen. Ilmatieteen laitoksen rajapinnasta haettiin Oulun säätiedot, ja HTTP GET-kyselyjen toimivuutta testattiin sisältäen tarkistuksia, kuten vastauksen JSON-muotoisuus ja tiettyjen merkkijonojen esiintyminen. Viides viikko: Luotiin neuroverkkoluokittelija, joka ottaa inputtina kolme arvoa: x, y, z. Neuroverkko suunniteltiin luokittelemaan syöte kuuteen eri luokkaan. Neuroverkon pohjana käytettiin Kerasin tarjoamaa esimerkkiä MNIST-konvoluutiomallista.

Lopputuotteen arviointi

Sensori lähetti kiihtyvyysarvoja ja suuntatietoja Linux-palvelimella sijaitsevaan tietokantaan onnistuneesti Python-koodin avulla. Tietokannasta haettiin oman ryhmän tunnusluvun alla olevia tietoja toisen ohjelman avulla, ja nämä tiedot tallennettiin CSV-tiedostoon. Tätä tiedostoa käytettiin apuna K-means-algoritmin kuvantamisessa mittauksista.

```
x = 1497, y = 1551, z = 1236
Direction: 5 -> Closest CP[5], Min Distance: 789
ADC reading:
x = 1497, y = 1552, z = 1237
Direction: 5 -> Closest CP[5], Min Distance: 789
Confusion matrix =
      cp0 cp1 cp2 cp3 cp4 cp5
cp0 100   0   0   0   0   0
cp1  0  100   0   0   0   0
cp2  0   0  100   0   0   0
cp3  0   0   0  100   0   0
cp4  0   0   0   0  100   0
cp5  0   0   0   0   0  100
```

Kuva 3. Hämmennysmatriisi

K-means-algoritmin avulla saatiin kuusi keskipistettä, jotka tallennettiin tekstitiedostoon. Tästä tekstitiedostosta saatiin arvot, joita käytettiin C-kielen ohjelmassa, joka tulosti hämmennysmatriisin. Kiihtyvyysanturilla tehtiin 100 mittausta eri suunnista, ja ohjelma laski, mihin keskipisteeseen oli lähin matka 3D-tilassa mitatusta pisteestä. Jos suunta vastasi keskipistettä, niin matriisi tulostui oikein.

Lopputulos

Päätehtävä onnistui hyvin, ja haasteista selvittiin. Lisätehtävät lisäsivät Linux osaamista. Harjaantumista lisättiin GitHubin ammattimaiseen käyttöön ja dokumentoitiiin. Työskentelyalustoina toimi VScode, nRF5340DK alusta, Google Colab, Raspberry Pi ja MySQL-palvelin.