Prototype 6 Data weergave verplaatsen van meetkast naar de centrale unit

hanlogo.png

Data weergave verplaatsen van meetkast naar de centrale unit - IoT Project - HBO ICT - 2024

Klas: ITN-IOT-A-f
Groepsnaam: Elektronische Levens Hulp (ELH)
Course: IoT Project
Versienummer: 1.0
Locatie: Nijmegen
Docent(en):
Eddy Luursema (Opdrachtgever)
Nils Bijleveld (Professional Skills)
Chris van Uffelen (Procesbegeleiding)
Student(en):
J.J.A. Visch (2104400)
N.T. Hoang (2112106)
R. Jurrius (631340)
L.P.W van Duijnhoven (2103948)
C.G.M van Kempen (2107890)
Datum: 24 april 2024

Inhoudsopgave

Inleiding

Wij willen dat het weergeven van data wordt verplaatst. Voorheen toonde de meetkast de data van de patiënt, maar na een demonstratie met onze opdrachtgever is er naar boven gekomen dat deze actie eigenlijk uitgevoerd moet worden op de centrale unit. Dit is logisch, aangezien dit ook ook is hoe het in de werkelijkheid zou gaan. In het echt zou de meetkast namelijk de data doorsturen naar de centrale unit en die zou pas data moeten tonen. Om dit te realiseren in ons project moeten er een paar aanpassingen gedaan worden.

Scenario

Er is nog steeds één meetkast en één centrale unit. Echter zijn er veranderingen gemaakt in de code. Naast de veranderingen in code die zijn gemaakt vanuit prototype 5 zijn er nog een paar wijzigingen gemaakt.
Het printen van de ontvangen data wordt niet meer gedaan in de meetkast. Dit is verplaatst naar de centrale unit. Daarnaast is er ook code geschreven om de ontvangen CAN data op de centrale unit uit te lezen, op te slaan en uiteindelijk weer te printen zoals voorheen werd gedaan in de meetkast. Door dit te printen kunnen wij met ons plotter programma de data tonen als grafieken.

Hieronder bevinden zich foto's van hoe het prototype er uit zien als opstelling en aansluitschema. In de eerste foto is te zien hoe de data verloopt door middel van lijnen.
De tweede foto representeert een beeld van hoe het prototype er fysiek uit ziet voor onze demo opstelling. Hierin zie je links de meetkast en helemaal rechts de centrale unit. Daar tussenin liggen twee CAN modules waar de data doorheen gaat. De groene tekst binnenin de blokken representeren de poorten. De groene lijnen representeren de kabels tussenin de hardware.

Opstelling_Prototype_6.jpg
p5opstelling.png

Concerns

1. Zal de code die de data print voor de plotter op de systeemkast ook werken op de centrale unit?
2. Kunnen wij de data van CAN fatsoenlijk omzetten in plotter data en moeten wij die mogelijk ook nog sorteren?

Fasering

Kleur	Betekenis

Rood	New
Oranje	In progress
Paars	Ready to review
Groen	Done

16-05-2024
Planningsgraaf_prototype_6_16-05-2024.png

17-05-2024
Planningsgraaf_prototype_6_17-05-2024.png

22-05-2024
Planningsgraaf_prototype_6_22-05-2024.png

24-05-2024
Planningsgraaf_prototype_6_24-05-2024.png

29-05-2024
Planningsgraaf_prototype_6_29-05-2024.png

Weergave verdeling code

Prototype_6_verdeling_code.jpg

Testplan

Algemene precondities

- De meetkast en de centrale units zijn verbonden volgens de weergave van de verdeling van de code.
- De Arduino en ESP32 kunnen code runnen en zijn verbonden met een laptop door middel van een USB kabel.
- Er wordt vanuit de meetkast een CAN bericht gestuurd naar de centrale unit door middel van onze CAN tabel.
- De debug variabele in debug.cpp moet op false staan.
- Op de Arduino moet de code staan van MeetKastConfig
- Op de ESP32 moet de code staan van CU_Config_Send

1. CAN data uitlezen

Preconditie

Er worden CAN berichten ontvangen die uit te lezen zijn door middel van onze CAN tabel.

Stappen

1. Sluit een Arduino en ESP32 aan volgens volgens de weergave van de verdeling van de code.
2. Sluit de Arduino en ESP32 met een USB kabel aan op een laptop voor stroom.
3. Run de code van MeetKastConfig op de Arduino.
4. Run de code van CU_Config_Send op de ESP32.

Verwacht resultaat

Als een bericht ontvangen wordt wordt er langs alle rijen van de CAN tabel afgegaan. Per rij wordt er gekeken naar het CAN ID. Uiteindelijk wordt er een rij gevonden waarin het CAN ID overeenkomt met het ID uit de ontvangen data. Als dit punt is bereikt weten we dat we in de juiste rij zitten en wordt de data opgeslagen.

2. Patiënt data opslaan

Preconditie

Data van CAN bericht is uitgelezen.

Stappen

1. Sluit een Arduino en ESP32 aan volgens volgens de weergave van de verdeling van de code.

2. Sluit de Arduino en ESP32 met een USB kabel aan op een laptop voor stroom.
3. Run de code van MeetKastConfig op de Arduino.
4. Run de code van CU_Config_Send op de ESP32.

Verwacht resultaat

Er is nu een multidimensionale array aangemaakt die gevuld is met minstens twee rijen (patiënten) en zeven kolommen (meetdata).

3. Patiënt data printen

Preconditie

Er is een multidimensionale array gevuld met een patiënt per rij en een meetwaarde per kolom.

Stappen

1. Open Realterm (Windows) of Coolterm (Linux). Kies de correcte port en kies baudrate 115200.
2. Druk vervolgens op "Open". Druk daarna op het tabje "Capture" en druk op "Start overwrite".
3. Open het programma Kst en open daarin het bestand Setup-Plotter-Prototype3-v4.kst

Verwacht resultaat

Er worden 7 meetdata's geprint per patiënt achter elkaar.

Testrapport

Scenario	Werkelijk resultaat	Conclusie
1. CAN data uitlezen	Als het ontvangen CAN Id ook in de config tabel staat, wordt de data succesvol opgeslagen. Zo niet, dan gebeurt er niks.	Geslaagd. Er moet dus wel een juist CAN Id ontvangen worden om succesvol te plotten. Ook kunnen we de conclusie trekken dat de patiënt zelf gegenereerd wordt door de centrale unit.
2. Patiënt data opslaan	Het CAN bericht wordt succesvol uitgelezen en de data wordt opgeslagen in een array.	Geslaagd. De data wordt opgeslagen, maar het is niet meer nodig dat we dit per patiënt doen. Voor de demo slaan we het nog wel op per patiënt.
3. Patiënt data printen	De 7 meetwaardes worden per patiënt geprint.	Geslaagd. De print code die wij voorheen gebruikte is nog steeds van toepassing.

Logboek

Main issue: https://sasscm.han.nl/redmine/issues/13463

Subtaken:

- Print code verwijderen meetkast #13468
- Code schrijven op CU om ontvangende data op te slaan en te printen #13469
- Ontvangende CAN data van meetkast uitlezen #13481
- Testplan schrijven #13470

Conclusie

We zullen in dit hoofdstuk een conclusie trekken over het nut van dit prototype en de resultaten daarvan. Ook gaan we de eerder benoemde concerns langs.

Concerns

Zal de code die de data print voor de plotter op de systeemkast ook werken op de centrale unit?

Ja, de code die is geschreven voor de plotter werkt nog steeds op de centrale unit. Dit komt omdat we op de centrale unit de ontvangende data opslaan in een twee dimensionale array zoals voorheen ook is gedaan. Toen ik dit concern schreef was ik

eigenlijk meer bezorgd over het feit of ik de data wel kon opslaan als twee dimensionale array. Eigenlijk had ik mijn concern dus anders moeten verwoorden.

Kunnen wij de data van CAN fatsoenlijk omzetten in plotter data en moeten wij die mogelijk ook nog sorteren?

Het omzetten van het CAN bericht naar plotter data is goed verlopen. Het was een kwestie van het CAN bericht uitlezen met de in Prototype 5 gemaakte CAN config tabel. In deze tabel moest nog een kleine aanpassing gedaan worden, namelijk het toevoegen van de machines aan de tabel. Door dit te doen kon ik de data aanwijzen naar een machine en kan de plotter het fatsoenlijk tonen.

Issues

Print code verwijderen meetkast

Achteraf gezien was dit niet groot genoeg om een issue op zich te zijn. Mijn bedoeling was eerst om de code op de meetkast aan te passen zodat de meetkast niet meer zou printen, maar dat de centrale unit dat zou dan. Achteraf is er in Prototype 5 een hoop aangepast aan de meetkast waardoor het verwijderen van deze print code ook is gedaan. Dit issue blijkt achteraf dan ook onnodig te zijn geweest.

Code schrijven op CU om ontvangende data op te slaan en te printen

Dit issue is succesvol uitgevoerd. De ontvangende data wordt nu fatsoenlijk geprint voor de plotter. Wat hier wel anders is gegaan dan verwacht, is het toewijzen van de opgeslagen data aan een patiënt. Na een demonstratie met onze opdrachtgever is het duidelijk geworden dat het niet uitmaakt welke mockdata van welke patiënt is. Alleen voor de demo is dit mooi om te zien. Puur voor de demo heb ik dan ook in mijn code geschreven dat de ontvangende data wordt aangewezen aan een willekeurige patiënt.

Ontvangende CAN data van meetkast uitlezen

Ook dit issue is succesvol verlopen. In het begin was het wel een beetje uitzoeken hoe de CAN config tabel werkte, maar nadat dit duidelijk was verliep dit prima. Er wordt door alle rijen van de CAN tabel langsgegaan en per rij wordt er gekeken of het CAN id van die rij gelijk is aan de CAN id van het ontvangen bericht. Als dit klopt, dan wordt er in deze loop data toegevoegd aan een tweedimensionale array. Bij deze array representeert elke rij een patiënt en elke kolom hierin een meetdata van deze patiënt. In de CAN tabel kijken we naar het machine ID die bij de ontvangende data hoort en gebruiken we dit ID vervolgens om de kolom in de array te vullen.

Hieronder staat de code van deze twee dimensionale array:

Opbouw array:

```
const uint8_t DIFFERENTPATIENTAMOUNT = 2;
const uint8_t DIFFERENTMACHINEAMOUNT = 7;
long patientData[DIFFERENTPATIENTAMOUNT][DIFFERENTMACHINEAMOUNT];
```

Opslaan data in array:

```
int tempPatientId = rand() % 2; // Data toewijzen aan patiënt voor demo
for (int i = 0; i < ROWS; i++) {
  if(configTable[i].canId == canId)
  {
    patientData[tempPatientId][configTable[i].machine]= data;
  }
}
```

Array printen:

```
for (int patientId = 0; patientId < DIFFERENTPATIENTAMOUNT; ++patientId) {
  for (int machineId = 0; machineId < DIFFERENTMACHINEAMOUNT; ++machineId) {
    Serial.print(patientData[patientId][machineId]);
    Serial.print(" ");
  }
```

```
}
Serial.println();
```

Testplan schrijven

Er was wat moeite met het testplan schrijven, aangezien er een misverstand was met wat de stappen waren daarin. Na feedback van andere groepsgenoten is het testplan nu af en is het gemaakt volgens onze eigen geschreven richtlijnen hiervoor.

Link van het testplan:
https://sasscm.han.nl/redmine/projects/intoft25_2023_intoft25_p4_01/wiki/Prototype_6_Data_weergave_verplaatsen_van_meetkast_naar_de_centrale_unit#Testplan