**FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE**

**DEPARTAMENTUL CALCULATOARE**

Logo, icon

Description automatically generated

**DOCUMENTAȚIE**

**PROIECT SEMESTRIAL**

**Disciplina**: Structura sistemelor de calcul

**Titlu:** Simulator protocol de comunicare in rețele de tip CAN

**Nume:** Pucani Liviu-Cătălin

**Grupa**: 30232

**An academic:** 2021 – 2022

**CUPRINS**

1. Introducere......................................................................................pag.2
2. Obiective.........................................................................................pag.2
3. Studiu bibliografic..........................................................................pag.2
4. Design si implementare..................................................................pag.3
5. Testare............................................................................................pag.4
6. Concluzii.........................................................................................pag.5
7. Bibliografie.....................................................................................pag.6

**1.Introducere**

CAN (Controller Area Network – Retea de control zonal) este un protocol de transmisie proiectat initial pentru electronica auto, dar in prezent este utilizat si in alte domenii cum ar fi: controlul industrial, automatizari sau retele de aparate medicale.

Principalul motiv pentru proiectarea unui astfel de protocol de transmisie a fost nevoia unui mecanism care sa foloseasca cat mai putine fire in automobilele care au ajuns sa foloseasca tot mai multi senzori si module, care initial erau conectate la un singur ECU (Electronic Control Unit). Strategia CAN a fost folosirea unei magistrale pe care mai multe noduri puteau transmite date.

In CAN nu se folosesc adrese, astfel ca un mesaj transmis de catre un nod va putea fi vizibil de toate celelalte noduri, iar fiecare nod in parte il va putea lua in considerare sau il va putea ignora. Arbitrarea magistralei se face pe un set de reguli stabilit la momentul proiectarii sistemului, pe baza id-ului nodurilor, așa ca in cele din urma unele mesaje vor avea o mai mare prioritate decat celelalte.

**2. Obiective**

Obiectivul proiectului este acela de a dezvolta o aplicatie care sa simuleze modalitatile de comunicare intre mai multe noduri ce fac parte dintr-o retea de tip CAN. Utilizatorul va avea posibilitatea sa trimita mesaje dintr-un anumit nod si poate vizualiza toate mesajele de pe magistrala CAN la un moment dat, iar cu ajutorul unui buton STEP, va putea sa parcurga toate mesajele de pe magistrala, astfel incat fiecare mesaj va fi interceptat de catre fiecare nod in parte, iar acesta din urma va hotari daca mesajul care este in acel moment pe magistrala ii este de folos sau nu.

**3. Studiu bibliografic**

**Principiu de funcționare**

CAN bus a fost proiectat pentru a reduce cantitatea de fire cerute în rețeaua de senzori auto • Fizic, mediul de transmisie fir, constă în două fire torsadate pe care se transmite semnal diferențial. Este proiectată pentru medii ce lucrează uneori în condiții extreme și poate 6 pentru medii ce lucrează uneori în condiții extreme și poate opera chiar dacă: – un fir este întrerupt – unul din fire este scurtcircuitat la masă – unul din fire este scurtcircuitat la tensiunea de alimentare – cele două fire sunt întrerupte în același loc. În acest caz cele două capete vor funcționa ca rețele separate.

Viteza relativ redusă pentru că durata unui bit trebuie să fie cel puţin dublul întârzierii de propagare pe bus – 1 Mb/s până la 25 metri – 800 kb/s până la 100 metri – 125 kb/s până la 500 metri 7 – 125 kb/s până la 500 metri – 50 kb/s până la 1000 metri • Magistrala CAN utilizează transmisia serială pe bit • Protocolul de magistrală permite mai mulți masteri pe magistrală • Fiecare nod al magistralei CAN are propriile drivere și receivere electrice prin care se conectează nodul la magistrală într-o manieră SI - cablat.

Dacă un nod transmite: – mesajul e acceptat de un singur alt nod (caz frecvent) – mesajul e acceptat de mai multe noduri (caz rar) • De exemplu: – nod detectare a temperaturii care trimite date de temperatură acceptate doar de un nod afișor. – dacă nodul detectează o supra-temperatură, datele transmise pot fi acceptate de mai multe noduri ce utilizează informația

**Variante CAN**

• ISO 11898-2. CAN de mare viteză. Utilizează semnal diferențial transmis prin 2 fire. Este cel mai răspândit standard în aplicații de automobile, transport feroviar de marfă și aplicații de rețele de control industrial. • ISO 11898-3. CAN de mică viteză (tolerant la erori)

• TTCAN (ISO 11898-4). Time Triggered CAN - 60

• TTCAN (ISO 11898-4). Time Triggered CAN - Protocolul TTCAN alocă ferestre de timp pentru raportare din partea nodurilor, ferestre ce sunt planificate și administrate pe magistrală astfel încât să se asigure că nu se produce supraîncărcare (overload) pe magistrală chiar şi în cele mai dezavantajoase situații. Se bazează pe un protocol CAN la nivelul legăturii de date furnizând în plus un Clock sistem pentru planificarea mesajelor.

• CANopen. Este proiectat pentru aplicații de control în industrie. Limitează numărul de noduri la 127 şi le alocă câte un ID

• SAE J1939. O întreagă familie de standarde industriale (agricultură, marină, camioane, autobuze, etc.) construite pentru servicii de comunicare a datelor pe baza CAN - ISO 11898-2.

• SAE J2411. CAN pe un singur fir (Single-wire CAN = SWC). Single Wire CAN este un nivel fizic de mică putere și viteză redusă utilizat mai ales în aplicațiile auto 62

• MilCAN - e definit pentru a fi utilizat în vehicule militare terestre unde este necesar un protocol de diagnosticare. El definește un set de reguli de utilizare și un nivel de software peste nivelurile rețelei CAN convenționale. S-a introdus un nod master de sincronizare (pentru sincronizare pseudo-hardware) şi care transmite pe magistrală cadre de sincronizare ce includ și un număr al slotului de timp de sincronizare

**4. Design si implementare**

Pentru a dezvolta aceasta aplicație am folosit limbajul de programare JAVA, si un design pattern de tipul MVC (Model View Controller). La partea de proiectare am folosit un model arhitectural de tip Layer pentru funcționalitate si unul de tip MVC pe partea de interfață grafica. Folosirea acestui tip de pattern arhitectural de tip stratificat determina existenta unor nivele care au niște clase corespunzătoare.

**Cerințe funcționale**

Programul va trebui sa permită utilizatorului sa își aleagă noduri din rețea din care dorește sa aibă rolul de expeditori de mesaje, sa poate scrie anumite mesaje ce se vor a fi trimise pe magistrala, si in cele din urma sa poată parcurge pas cu pas fiecare mesaj din magistrala si sa vadă cum fiecare nod primește mesajele respective.

**Cerințe non-funcționale**

* Interfața simulatorului trebuie să fie prietenoasă și ușor de utilizat.
* Programul sa fie eficient din punct de vedere a memoriei si a timpului de executie
* Interfata grafica sa ofere un feedback vizual asupra mesajelor noi adaugate pe magistrala si pentru mesajele receptionate la nivelul fiecarui nod in parte.

**4. Testare**

**A screenshot of a computer

Description automatically generated with medium confidence** După cum se poate vedea in exemplu, daca trimite un mesaj de la nodul nr.4, acesta va fi trimis pe magistrala si poate fi văzut in tabelul din partea stângă.

A screenshot of a computer

Description automatically generated with medium confidence Daca trimitem un mesaj de la nodul nr.1, acesta având un cod de identificare mai mic decât cel al nodului cu nr.4, mesajul va fi prioritizat prin procedeul de arbitrare, asa ca va fi pus pe o poziție superioara in tabela cu mesaje.

In aceasta captura se poate vedea ca atunci când utilizatorul alege sa parcurgă pas cu pas fiecare mesaj din tabela de magistrala, la fiecare nod in câmpul rx va fi afișat mesajul de pe magistrala.

Graphical user interface, application

Description automatically generated

**5. Concluzie**

In concluzie, protocolul de comunicare CAN este unul foarte performant si fezabil in domeniul industrial, dar si in cel de mașini, având ca beneficii o viteza mare de transfer de date, dar in același timp este un protocol ce face economie de cabluri, nefiind necesara o conexiune directa de la fiecare nod la un nod principal, ci folosind o magistrala ce utilizează un mecanism de arbitrare bazat pe prioritizarea mesajelor ce provin de la noduri mai importante.

**7. Bibliografie**

<https://www.ti.com/lit/an/sloa101b/sloa101b.pdf?ts=1634278660612&ref_url=https%253A%252F%252Fwww.google.com%252F>

<https://dewesoft.com/daq/what-is-can-bus>

<http://etc.unitbv.ro/~romanca/EmbSys/Pentru%20anul%204%20EA/12-13-CAN-bus.pdf>

<http://stst.elia.pub.ro/RIC/Teme_RIC_2008_9/RazvanMocanu/RazvanMocanu_Retele%20industriale.pdf>

<https://www.javatpoint.com/can-protocol>

<https://www.kvaser.com/about-can/the-can-protocol/can-messages-33/>