**Buna ziua,**

Numele meu este Buzuriu George-Lucian si astazi am sa va vorbesc despre optimizarea timpului de reactie in sisteme TCU multicore.

**Ce este un TCU?**

* TCU sau unitatea de control a cutiilor cu transmisie automată, este un dispozitiv care controlează transmisiile electronice moderne automate. Un TCU foloseşte, în general, senzorii de la vehicul, precum şi datele furnizate de către unitatea de control a motorului pentru a calcula cum şi când să schimbe vitezele în vehicul pentru performanțe optime, economie de combustibil şi de calitate a schimbării treptelor.

**Ce se intelege prin timp de reactie?**

* Prin termenul de timp de reactie, m-am referit la perioada in care TCU-ul incepe sa comunice cu celelalte componente, adica cu unitatile de control electronice din masina. Deci, timpul de reactie este de fapt timpul de initializare al unitatii de control a transmisiei. Astfel, dupa cum reiese si din titlul lucrarii, imi doresc sa reduc acest timp de initializare.

**De ce este necesara imbunatatirea?**

* Dorinta imbunatatirii a iesit la iveala in momentul in care Hardware-ul proiectului s-a schimbat. De la un microcontroler cu un singur core, TC1784, s-a trecut la un microcontroler cu 3 core-uri, Aurix TC277. Bineinteles, odata cu schimbarea, s-au preluat in totalitatea functionalitatile dezvoltate pe generatia veche.

In continuare, o sa ilustrez o schema succinta de comunicare a TCU-ului cu celelalte componente electronice care la randul lor manipuleza: motorul, QSP,ESP... Acestea comunica pe FlexRay. Folosind Gateway-ul se transforma comunicare din FlexRay in CAN pentru serviciile de UDS care permit testerului sa verifice masina.

**Componentele de initializare ale TCU-ului**

Datorita faptului ca microcontrolerul are spatiul pentru data flash, Dflash, avem EEPROM emulat. Bineinteles, chipul de EEPROM este la locul lui. In plus, avem si functionalitatile de safety. Asdar, pentru ca un TCU sa se initializeze, este necesar sa citim datele din EEPROM-ul emulat, EEPROM-ul fizic si sa realizeze Switch off patch checkul.

Deoarece este o memorie din care doar putem citi, este clar ca avem nevoie de un element intermediar. Acesta il presupune oglinda RAM. Aici, ne vom salva datele de la ultima folosire a masinii, atat date din EEPROM fizic, cat si date din EEPROM emulat.

In continuare, prezint o mica schema care ilustreaza intreg procesul de stocare si folosire al datelor. : Astfel,

* Se actioneaza KL15, datele din EEPROM de la ultima functionare a masinii sunt necesare
* Atat cele din EEPROM fizic cat si cele din EEPROM emulat le stocam in oglinda RAM
* De aici, le putem accesa pentru folosirea locala
* Totodata, putem altera aceste date si rescrie valorile.
* La KL15 OFF, vom memora datele in EEPROM, deoarece informatiile din RAM se vor pierde.

Cand se scriu date in RAM:

* La KL15 ON
* Pe durata functionarii
* La momente de timp bine stabilite

Revenind la procesul de imbunatatire al timpului, prima idee logica si intuitiva a fost sa impart blocurile de memorie din EEPROM-ul fizic. Deoarece microcontrolerul comunica pe SPI cu EEPROM-ul extern mi-a fost aproape imposibila o asemenea solutie. Intervenea o problema la nivel de arbitrare. Pana la urma, solutia 1 a fost sa iau tot calupul de date din EEPROM-ul fizic si sa le pun pe core-ul 0, lasand operatiile aditionale pe core-ul1. Atunci am descoperit ca citirea blocurilor de memorie din EEPROM-ul fizic este realizata prin intermediul a 2 functii. Beneficiul nu a fost cel scontat, astfel ca am continuat cautarile.

Tinand cont de elementele initializarii TCU-ului, EEPROM fizic, EEPROM emulat si SOPC.