

Merenje temperature motora u automobilu

Napisati softver za simulaciju sistema za merenje temperature u automobilu. Napraviti minimum 4 taska, jedan task za prijem podataka od PC-ja, jedan task za slanje podataka PC-ju, treći za merenje temperature, i četvrti za prikaz na displeju. Po potrebi dodati još taskova. Sihronizaciju između tajmera i taskova, kao i između taskova ako je potrebno, realizovati pomoću semafora (ili task notifikacija) ili mutexa, zavisno od potrebe. Podatke između taskova slati preko redova (*queue*).

Trenutne vrednosti temperature simulirati pomoću UniCom, simulatora “serijske” komunikacije. Radi bolje sigurnosti, implementirana su dva ista senzora od po 150 oma. Računati da se informacije o trenutnoj temperaturi dobijaju preko UniCom softvera svakih 200ms na kanalima 0 i 1. Temperatura se dobija kao vrednost otpornosti, maksimalne vrednosti 150 oma. Najniža temperatura od 0 stepeni odgovara otpornosti od 0 oma, maksimalna temperatura od 100 stepeni odgovara otpornosti od 150 oma. Karakteristika senzora je linearna. Komunikaciju sa PC-jem ostvariti isto preko simulatora serijske veze, ali na kanalu 2. Za simulaciju displeja koristiti Seg7Mux, a za simulaciju logičkih ulaza i izlaza koristiti LED bar.

1. Pratiti temperaturu. Posmatrati vrednosti koje se dobijaju iz UniCom softvera kao vrednosti otpornosti, koje imaju opseg od 0 do 150 oma. Uzimati zadnjih 5 očitavanja i usrednjavati ih. Temperatura motora je jednaka srednjoj vrednosti temperature sa dva senzora ($\text{temperatura} = (\text{temp_senzor_1} + \text{temp_senzor_2}) / 2$). Ne vršiti nikakve proračune u interapt funkciji, samo proslediti podatke tasku za merenje temperature. Treba da se primaju podaci sa dva senzora na dva odvojena kanala serijske komunikacije.

2. Ukoliko je razlika očitavanja temperature sa senzora 1 i senzora 2 veća od 5 stepeni celzijusa (npr. $\text{temp_senzor_1} = 40$, $\text{temp_senzor_2} = 47$), uključiti alarm za pogrešno očitavanje senzora – jedan stubac LED bara treba da blinka periodom od 1000ms.

3. Napraviti histerezis tako da ukoliko je temperatura viša od 90 stepeni celzijusa, uključuje se „ventilator“ (simulirati ventilator kao donju LED diodu drugog stupca LED bara). „Ventilator“ se isključuje kad se temperatura spusti ispod 85 stepeni celzijusa.

3. Realizovati komunikaciju sa simuliranim sistemom. Slati naredbe preko simulirane serijske komunikacije. Naredbe i poruke koje se salju preko serijske veze treba da sadrže samo ascii slova i brojeve, i trebaju se završavati sa *carriage return (CR)*, tj brojem 13 (decimalno), čime se detektuje kraj poruke. Naredbe su:

a. Automatsko obavestavanje o prevelikoj i premaloj temperaturi: Ukoliko je temperatura viša od 95 stepeni, poslati poruku upozorenja i uključiti treći stubac LED bara da blika periodom od 100ms.

b. Ukoliko je temperatura viša od 90 stepeni, poslati poruku da je „ventilator“ uključen. Ukoliko temperatura padne ispod 85 stepeni, poslati poruku da je „ventilator“ isključen.

c. Ako neki od senzora pošalje vrednost izvan opsega (od 0 do 150 oma), poslati poruku da je senzor neispravan.

3. Na LCD displeju prikazati trenutnu vrednost temperature motora, brzina osvežavanja podataka 100ms.

4. Misra pravila. Kod koji vi budete pisali mora poštovati MISRA pravila, ako je to moguće (ako nije dodati komentar zašto to nije moguće). Nije potrebno ispravljati kod koji niste pisali, tj. sve one biblioteke FreeRTOSa i simulatora hardvera.

5. Obavezno je projekat postaviti na GitHub i na gitu je obavezno da ima barem jedan issue i pull request. Takođe obavezan je Readme.md fajl kao i .gitignore fajl sa odgovarajućim sadržajima. Navesti u Readme fajlu kako testirati projekat.