

TARTU ÜLIKOOL
LOODUS- JA TEHNOLOOGIA TEADUSKOND

Füüsika Instituut

Madis Nõmme

TEGUMITE HALDAMINE SafeRTOS
OPERATSIOONISÜSTEEMIGA KAARDIL DK-
LM3S9B96

Praktikumi juhend aines Reaalajasüsteemid

Juhendaja: Aivo Reinart

Tartu
2011

1 TÖÖ EESMÄRGID

Praktikumi eesmärgiks on etteantud koodinäidiseid täiendades luua kasutatavale Stellaris DK-LM3S9B96 *development kit*’ile (edaspidi seade) töötav programm, mis sisaldaks endas:

- vähemalt kahte tegumit
- tegumid täidavad mingit ülesannet
- toimub tegumitevaheline andmevahetus

Üheks võimaluseks oleks kasutada ja täiendada praktikumi juhendiga kaasas olevat tegumi koodi, mis väljastab infot ekraanile. Ekraanile väljastatav info tuleks teisest tegumist.

2 ALGTINGIMUSED

2.1 Seadme kirjeldus

Praktikumi töö on koostatud seadmele mudeli koodiga DK-LM3S9B96. Seadmel on:

1. 8 MB SDRAM moodul
2. MicroSD kaardi liides
3. Audio liides (*Integrated Interchip Sound (I2S)*)
4. Ethernet liides (RJ-45)
5. CAN (*Controller Area Network*) liides
6. 1 Potentsiomeeter
7. 3.5 tolline puuteekraan
8. 1 kasutaja poolt kontrollitav lüliti

2.2 Kasutatav tarkvara

Juhend on koostatud töötama IAR Embedded Workbench keskkonnas. Kasutatav tarkvara on alla laetav lehelt <http://www.luminarymicro.com/products/dk-lm3s9b96.html>.

2.3 Projektifaili ülesehitus

Kuna IAR Embedded Workbench IDE'ga otse ei õnnestunud seadmel käivitavat tühja projekti luua, siis minimaalse konfiguratsiooni leidmiseks võtsin ühe kaasasoleva näiteprojekti ja eemaldas sellelt kõik üleliigse selliselt, et algne funktsionaalsus alles jääks. Seadmega kaasasolevad näited on mõeldud töötama ka teiste tootjate tööriistadega ja kompilaatoritega (Keil, gcc, TI Code Composer Studio) ja sisaldavad seetõttu neile vahenditele vajalikke faile.

Fail	Ülesanne
./blink.c	Sisaldab sisenemispunkti (<i>entry point</i>) programmile. Peab sisaldama main() funktsiooni.
./startup_ewarm.c	Programmikoodi ülesseadmine (__iar_program_start(void)) vigade käsitlemise (NmISR(void), FaultISR(void), IntDefaultHandler(void),) definitsioonid, süsteemi pinu reserveerimine.
blink.dep	Projekti korrektseks tööks vajalike sõltuvuste ja nõutavate failide definitsioonid. Sealhulgas kompileerimise tulemid.
blink.ld	Instruktsioonid linkerile
blink.ewp	IAR Embedded Workbench IDE projektifail
blink.ewd	Veel Embedded Workbench projektiga seotud seadeid.

Töö käigus tekitatakse projekti kataloogi settings ja ewarm nimelised kataloogid. Esimeses neist on IDE väljanägemisega seotud seadistused, teises kompileerimise ja linkimisega seotud tulemid.

3 SafeRTOS KASUTAMINE JA API TUTVUSTUS

Programm, mis kasutab SafeRTOS'I peab pakkuma kolm *callback* funktsiooni:

- **vApplicationErrorHook(...)** – ei tohi tagasi pöörduda (sisaldada *return*). Võtab kolm parameetrit xCurrentTask(*task*, millest tema poole pöörduti),

pcErrorString(viga seletav sõne), xErrorCode(üks kolmest konstandist: errINVALID_TICK_VALUE, errINVALID_TASK_SELECTED, errTASK_STACK_OVERFLOW)

- **vApplicationDeleteHook(...)** – kutsutakse välja, kui *task* kustutatakse. *Task* võib ka ise ennast kustutada.
- **vApplicationIdleHook(...)** – kutsutakse välja *idle* olekus. võimaldab mitte midagi tegemise asemel madala prioriteediga rakendusega seotud tegevusi teha. Kood selles funktsioonis ei tohi kutsuda välja API funktsioone, mis võiksid ta blokeeritud seisu viia.

Kui kasutaja kood käivitatakse seadmel, siis esimene API funktsioon, mida kutsutakse peab olema vTaskInitializeScheduler(). Kuna kogu kasutaja funktsionaalsus tuleks realiseerida *task*'ides, siis peab *scheduler* olema kättesaadav.

3.1 Task'ide vaheline suhtlus

Task'ide vaheliseks suhtluseks pakub SafeRTOS järjekorra mehhanismi (FIFO puhver). Info saatmiseks ühelt *task*'ilt teisele peab üks kirjutama puhvrissi ja teine sealt lugema. Järjekorrast kirjutamine käib xQueueSend() ja lugemine xQueueReceive() funktsioonide abil. Üksteisele infot saavad *task*'id peavad olema samal arusaamisel info tähenduse kohta. Infot liigutatakse puhvrissi ja sealt välja kopeerimise teel. Seetõttu kui on vaja saata suurem hulk andmeid teisele tegumile on andmete asemel soovitatav saata viit neile.

4 TÖÖKÄIGU KIRJELDUS

4.1 Ettevalmistus

1. Veenduda, et arvutisse on paigaldatud on IAR Embedded Workbench. Selle puudumisel laadida see alla aadressilt <http://www.luminarymicro.com/products/dk-lm3s9b96.html> (DK-LM3S9B96-CD-631 alt) ja paigaldada. Praktikumi *workspace* on seadistatud töötama selliselt, et vajalikud teegid oleksid C:\StellarisWare kataloogis.

2. Tõmmata arvutisse praktikumi juurde kuuluv *workspace* fail (koos projektifaili ja koodifailidega).

4.2 SafeRTOS API'ga tutvumine

Edasist tööd lihtsustab oluliselt, kui kasvõi lugeda SafeRTOS kasutusjuhendit. See on praktikumi failidega kaasas failis SafeRTOS_UM.pdf. Kuna manuaal on varustatud paljude koodinäidetega, läheb lugemine kiiresti. Tuleks leida vastused küsimustele:

1. Millised osad on SafeRTOS'il? (lk 15-25)
2. Mis funktsiooni kasutatakse *scheduler*'i käivitamiseks. Millised nõuded sellele on? (lk 18-20)
3. Milliste funktsioonide abil tekitatakse uusi tegumeid (*tasks*)? (lk 29-39)

4.3 Töö koodiga

5 TULEMUSED

Tulemuseks peab projekt kompileeruma ja seadmel käivituma. Peale käivitamist kuvatakse ekraani ülemises osas nuppu [RUN <teie *task*'i nimi>]. Ekraani alumisse ossa pealkirjaga *Result* peaks ilmuma teie koostatud ülesande väljund.

6 LISAD

6.1 Kood

6.2 Kasutatud materjalid

LM3S9B96 Development Kit User's Manual,
(<http://www.luminarymicro.com/products/dk-lm3s9b96.html>)

SafeRTOS User's Manual (<http://www.luminarymicro.com/products/dk-lm3s9b96.html>)

TI E2E Community Support Forum

(http://e2e.ti.com/support/microcontrollers/stellaris_arm_cortex-m3_microcontroller/f/)

LM3S9B96 Forum (<http://highintegritysystems.zendesk.com/forums/110575-lm3s9b96>)