

2. Ülesanne

Ülesanne on taimer. Tuleb realiseerida taimer, kasutades sisendina 8't lülitit, millega antakse binaarkoodis ette sekundid. 7-segmen dilise ekraani pealt peaks olema väärtus nähtav kuueteistkümnendkoodis. Peale sisestust tuleb 7-segmen dilisel ekraanil näidata allesjäänud sekundeid. Lisa punkte saadakse, kui aja nulli jõudmisel hakkab vilkuma punane LED tuluke. Kõikide arvsüsteemide konversiooniga tegelevad funktsioonid ise.

Lisaks eelmises ülesandes kasutatud nuppudele ja LED' dele, on taimeri realiseerimiseks vaja ka 7segmen dilist ekraani(*Seven Segment Display* ehk SSD) ja RGB-LED'i ning lüliteid.

SSD teek: `ssd.h`

RGB-LED'i teek: `rgbled.h`

Lülite teek: `swt.h`

Sarnaselt teistele plaadi komponentidele, tuleb ka SSD, lülid ja RGB-LED initialiseerida enne kasutamist ning project peab sisaldama nende vastavaid *Source* ja *Header* file.

Selleks kasutame "`SSD_Init()`", "`SWT_Init()`" ja "`RGBLED_Init()`" käske.

`RGBLED_Init()` käsuga peab olema ettevaatlik, kuna see aeglustab kontrolleri tööd. Seega see on ainuke `Init` käsk mida peaks kasutama viivitamatult enne RGB-LED funktsiooni kasutamist ning koheselt pärast funktsioonide kasutust peaks selle sulgema käsuga `RGBLED_Close()`.

SSD ekraanile kuvamiseks on vaja kasutada käsku:

```
SSD_WriteDigitsGrouped(int x, int y)
```

x on kuvatav täisarv ja y on komakohad (siin ülesandes mitte vajalik ja võib lihtsalt sisestada 0'i).

Lülite lugemiseks saab kasutada `SWT_GetValue(int x)` ning `SWT_GetGroupValue()`

Nende kahe vahe on see, et esimene annab ainult ühe lüliti staatuse, teine annab kõikide 8-bitises formaadis (0 tähendab kõik üleval, ehk väljalülitatud, ja 255, et kõik all, ehk sisselülitatud).

RGB-LED kasutamiseks on käsk: `RGBLED_SetValue(int r, int g, int b)`

Iga täisarv näitab kaheksa bitiga ühe värvi osa LED's. Näiteks roheline värvi jaoks, tuleb sisestada:

0, 255, 0

Ja lilla värvi jaoks tuleb sisestada:

255, 0, 255