

## Kalkulaator

Realiseerida treeningplaadi peal lihtne kalkulaator, kasutades sisendiks 8't lülitit, viite nuppu ja väljunditeks LED'e, 7-segmendi ekraani ja LCD ekraani. Kalkulaatoril peavad olema vähemalt need 4 funktsiooni: liitmine, lahutamine, korrutamine ja jagamine. LCD ekraanil peab olema kuvatud kogu vajalik informatsioon. Iga arvutuse vastus peab olema salvestatud mällu ja kasutatav järgmises arvutuses. Eraldi *reset*'i ei ole vaja teha, sisse ehitatud *reset* nupuga saab alustada arvutamist nullist.

Alustuseks tuleb alla tõmmata projekt:

<http://edu.pld.ttu.ee/~trkal/mikrokontrollerid/Exercise3.zip>

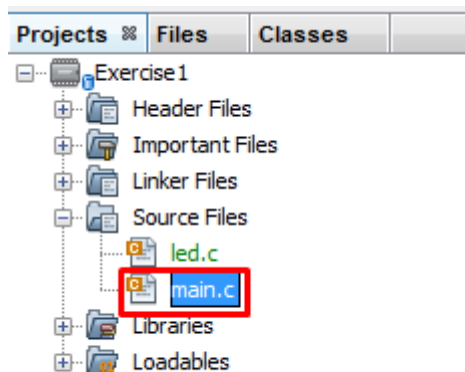
Teiseks, eralda(*extract*) allatõmmatud arhiiv *Desktop*'le.

Kolmandaks, ava terminaal MPLAB, kasutades käsku *cad* ja *mplab*.

*Mplab*'s projekti avamiseks kasutage *Open Project*:

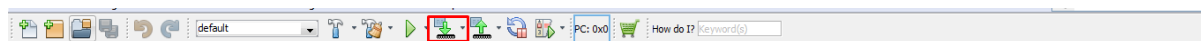


Otsige paremalt aknast *Source Files* alt *main.c* ja tehke topeltklõps selle peal:



Kogu kood kirjutage *main* funktsiooni, lõpmatusse tsüklisse. Ärge unustage komponente initsialiseerida.

Väga tähtis, et kood tööle saada on jätta meelde, et muuta tohib AINULT *main.c* faili ja ainult *main* funktsiooni. Vajadusel võib juurde teha enda funktsioone või lisada vajalikke teeke.



Koodi laadimiseks plaadile kasutage nuppu *Make and Program Device Main Project*.

## Funktsioonide näiteid:

Et initsialiseerida 8't lülilit, tuleb kasutada `SWT_Init()` ;

Teistel komponentidel on teised nimed funktsioonile, kõik lühendid leiab *Header files* alt.

Et lugeda lüliteid, saab kasutada kas `SWT_GetValue(int x)`, kus `x` näitab soovitava lüliti numbrit, või `SWT_GetGroupValue()`, see funktsioon annab kõigi lülitite väärtused.

*LED*'ide väärtustamiseks saab kasutada `LED_SetValue(int x, int y)`, kus `x` näitab soovitava *LED*'i väärtust ja `y` annab oleku(1 või 0 ehk põleb või ei põle). Et väärtustada kõiki *LED*'d korraga, näiteks lülitite väärtuste järgi, siis saab kasutada `LED_SetGroupValue(int x)`, kus siis `x` näitab ära kõikide *LED*'de oleku.

Nupu oleku saab kätte funktsiooniga `BTN_GetValue(char c)`, kus `c` on soovitud nupu täht (näiteks keskmine nupp(*center*) puhul on `c` väärtus `'c'`, vasaku(*left*) nupu puhul aga `'l'`. Funktsioon tagastab väärtuse 1, kui nupp on alla vajutatud ja väärtuse 0, kui nupp on üleval.

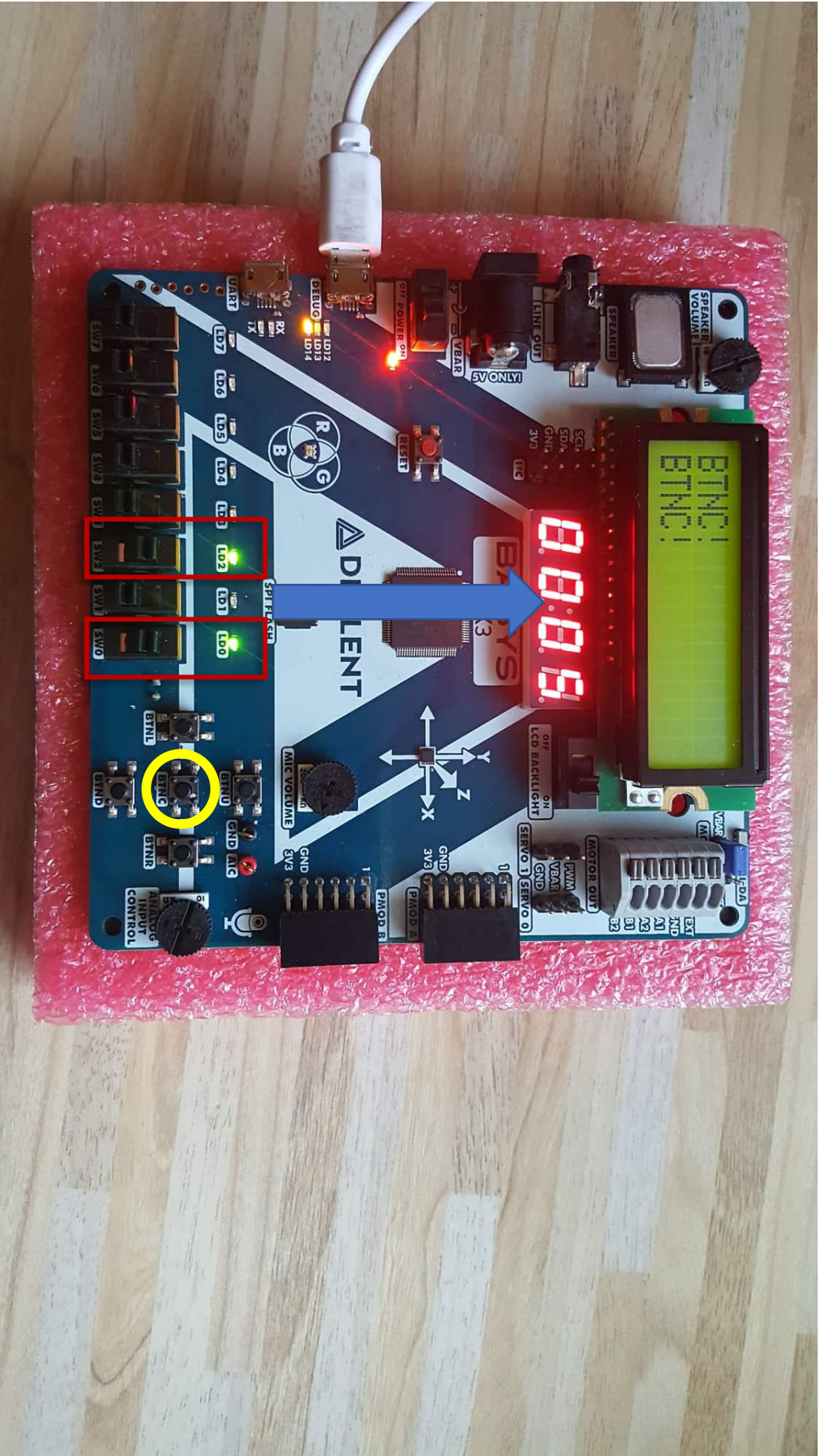
7-segmendiline ekraan annab väärtuse 16nd kujul (ehk 0000-FFFF). Ekraanile kirjutamiseks saab kasutada funktsiooni `SSD_WriteDigitsGrouped(int x, int y)`, kus `x` on kirjutatav arv ja `y` on komakohtade seadistus(ei ole vaja kasutada selles ülesandes, asendage lihtsalt 0'ga).

LCD-ekraanile kirjutamiseks on funktsioon `LCD_WriteStringAtPos(char* str, int x, int y)`, mis võtab 3 parameetrit. Esimeseks on string, mis kirjutatakse ekraanile(saab olla ainult *string*'i või *char* kujul), teiseks on arv, mis näitab kui kaugel vasakust äärest hakatakse ekraanile kirjutama ja kolmandaks on arv, mis saab olla 0 või 1. 0 kirjutab ülemisele reale ja 1 alumisele. Ekraanile mahub ühte ritta 16 sümbolit.

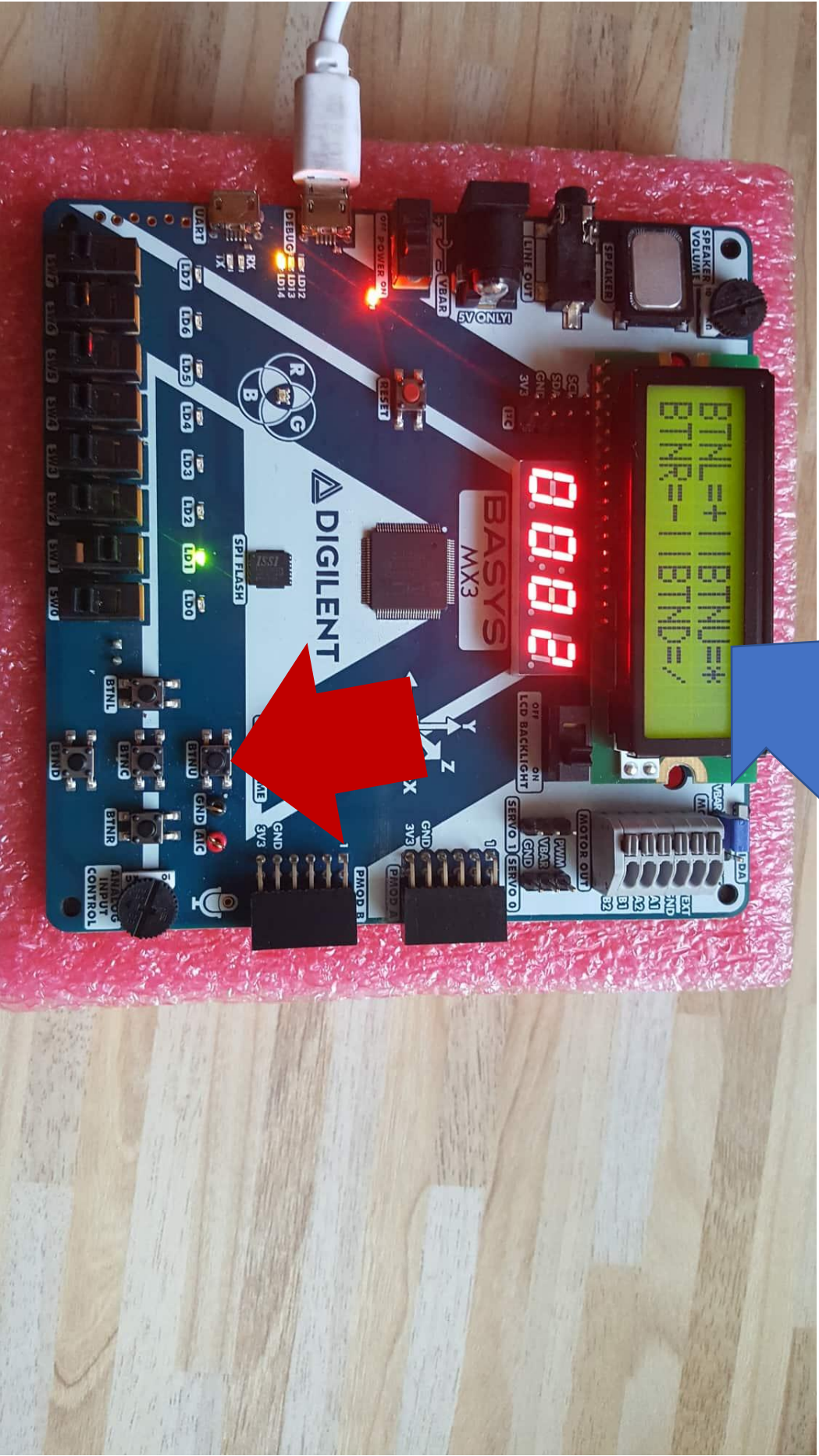
Rohkem infot erinevate funktsioonide kohta saab projektis sisalduvatest teekidest, aga mitte ühelgi juhul ei tohi neid muuta.

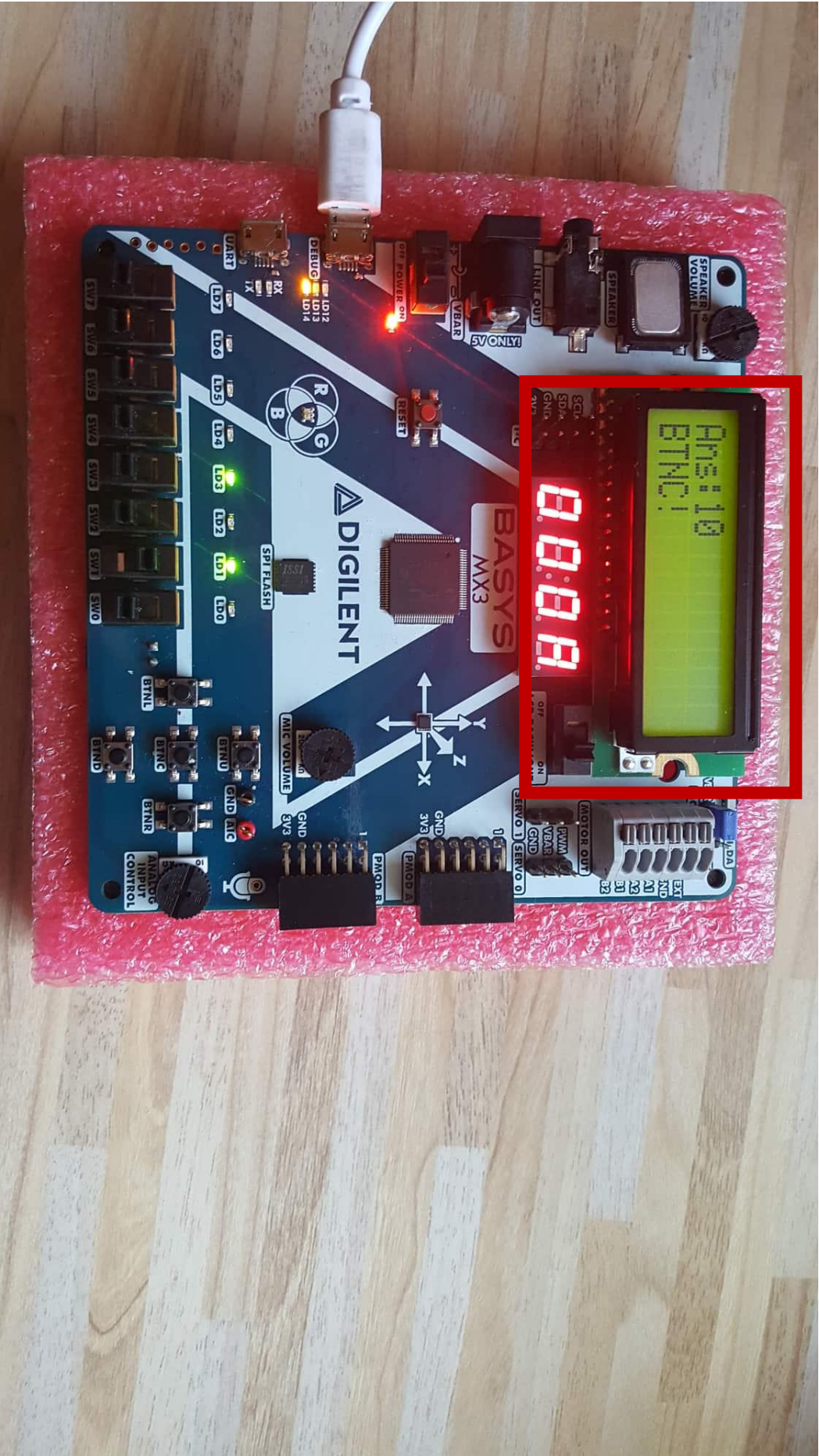
## Järgnevatel lehtedel on pilte näidised.

Esimesel pildil sisestatakse operand 5. Teisel pildil valitakse operand 2. Kolmandal pildil on tehtud korrutus tehe (BNTU) ja vastus on saadud 10(16nd süsteemis A). Neljandal pildil valitakse uus operand, 3. Viiendal valitakse liitmis tehe ja kuuendal on uus operand liidetud meie eelmisele vastusele(10) ja saadud 13(vastus kuvatakse ka LED'dele). Näidisel kasutatakse keskmist nuppu, et liikuda operaatorite menüüsse peale esimese operandi valimist.









ANS:10  
BTNC:  
0:0:0:0



