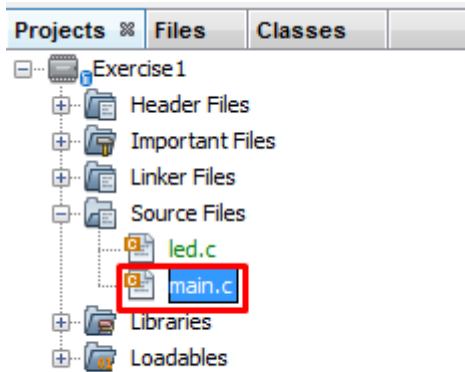


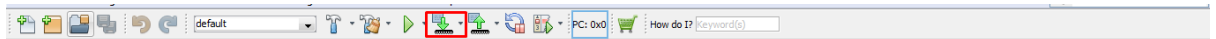
Iga ülesande jaoks antakse teekide ja osaliselt valmis kirjutatud main failiga projekt ette. Projekti avamiseks tuleb valida tööriista ribalt *Open Project* (joonisel ümbritsetud punase kastiga)



*Main* faili muutmiseks valida vasakult aknast *Source Files* kaust ja sealt *main.c*



*Main.c* failis tuleks muuta ainult lisatud teeke ja main funktsiooni.



Punase kastiga ümbritsetud nupuga saab programmi plaadile laadida, kui arvuti on plaadiga USB kaabliga ühendatud. Plaadil peab juhe olema *debug* pesas.

1. Ülesanne on KITT'i tuled. Tuleb realiseerida plaadi 8 LED'i peal KITT'i tuledega analoogne vilkumine. Lisa punktide jaoks tuleks panna tuled tööle nupust ning teha ka inverteeritud versioon. (Inverteeritud versioonis kõik tuled alguses põlevad ja siis kustuvad järjekorras)

Plaadi funktsioonidest on vajalikud LED'id ja nupud(*buttons*). LED'de kasutamiseks on vaja päises lisada(*include*) LED'i teek(*library*) ja nuppude puhul nuppude teek.

LED'i teek: `led.c`

Nuppude teek: `btn.c`

Samuti tuleb initsialiseerida mõlema komponendi algväärtustamise funktsioon.

LED'i puhul on selleks: `LED_Init()`

Nuppude puhul on selleks: `BTN_Init()`

Need funktsioonid tuleks kutsuda välja main funktsiooni alguses.

Peale deklaratsioone on main funktsioonis vaja teha lõpmatu tsükkel, kus toimub kogu plaadi töö. Sinna tsüklisse tuleks koostada algoritm, mis realiseerib KITT'i tuled.

Nuppu vajutuse lugemiseks tuleb kasutada funktsiooni `BTN_GetValue(char X)`, kus X näitab soovitu nuppu(nuppude tähte näeb plaadil iga nupu kõrval). Funktsioon tagastab 0 kui nupp on üleval ja 1 kui nupp on all.

LED'e saab väärtustada funktsiooniga `LED_SetValue(int x, int y)`, kus x näitab mitmendat LED'i väärtustatakse(iga LED'i numbrit näeb plaadilt) ja y on 0 kui LED tuleb kustutada ja 0'st erinev kui LED tuleb põlema panna.

`DelayApprox10Us(int x)` saab kasutada viivituse tekitamiseks, x asemele antakse viivituse aeg 10 mikrosekundi kaupa.