

Osnove mikroprocesorske elektronike

Priprava 5: LED skupina

Namen te priprave je, da si v sklopu projekta pripravite vse potrebno, da se nato lahko lotimo dejanske implementacije *sistemskih* funkcij, ki bodo upravljale z Miškotovimi LEDicami.

Priprava na vajo

Pomembno je, da spodnje stvari pripravite že doma, saj bomo tako imeli na vajah *na voljo več časa za bolj pomembne stvari*: kako implementirati funkcije, ki bodo upravljale z LEDicami tako, kot da so le-te urejene v nekakšno skupino. Kar pa tudi so – če pogledate na tiskanino.

Pripravite torej sledeče stvari:

1. ***ustvarite nov projekt z imenom*** `VAJA_05-LEDs`.

2. ***Poskrbite, da je sistemska ura mikrokrmilnika nastavljena ustrezno.***

Uro nastavite na enak način, kot ste to storili pri 3. vaji. Obljubimo, da jo tokrat nastavljate zadnjič.

3. ***Podučite se, kako se STM Cube projektu doda nova mapa, v kateri hranite izvorno kodo.***

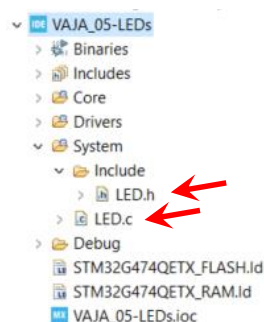
Znotraj projektne mape sicer lahko ročno ustvarite novo mapo, a STM Cube te mape in njene vsebine ne bo upošteval, vse dokler mape ne dodate v tako imenovano "iskalno pot" (angl. path) za `*.c` datoteke oziroma za `*.h` datoteke (v žargonu imenovane "header datoteke").

4. ***Projektu dodajte mapo "System" ter znotraj te mape še mapo "Include".***

V mapi "System" bomo hranili `*.c` datoteke modulov *sistemskega nivoja*, v mapi "Include" pa pripadajoče `*.h` datoteke.

5. ***Projektu dodajte modul LED*** (torej `LED.c` in `LED.h` datoteki, ki jih najdete v mapi "predloga").

Dodajte jih seveda v podmapo "System" na ustrezno mesto.



6. Preučite, kateri pini mikrokrmilnika upravljajo z Miškotovimi LEDicami.

Pomagajte si seveda z [električno shemo Miškota](#).

7. V grafičnem vmesniku CubeMX nastavite te pine kot digitalne izhode s sledečimi lastnostmi:

1. vrsta izhoda: simetrični izhod ("push-pull" oz. "totem pole")
2. onemogočeni zgornji upor ("pull-up") in spodnji upor ("pull-down")
3. hitrost izhoda: nizka

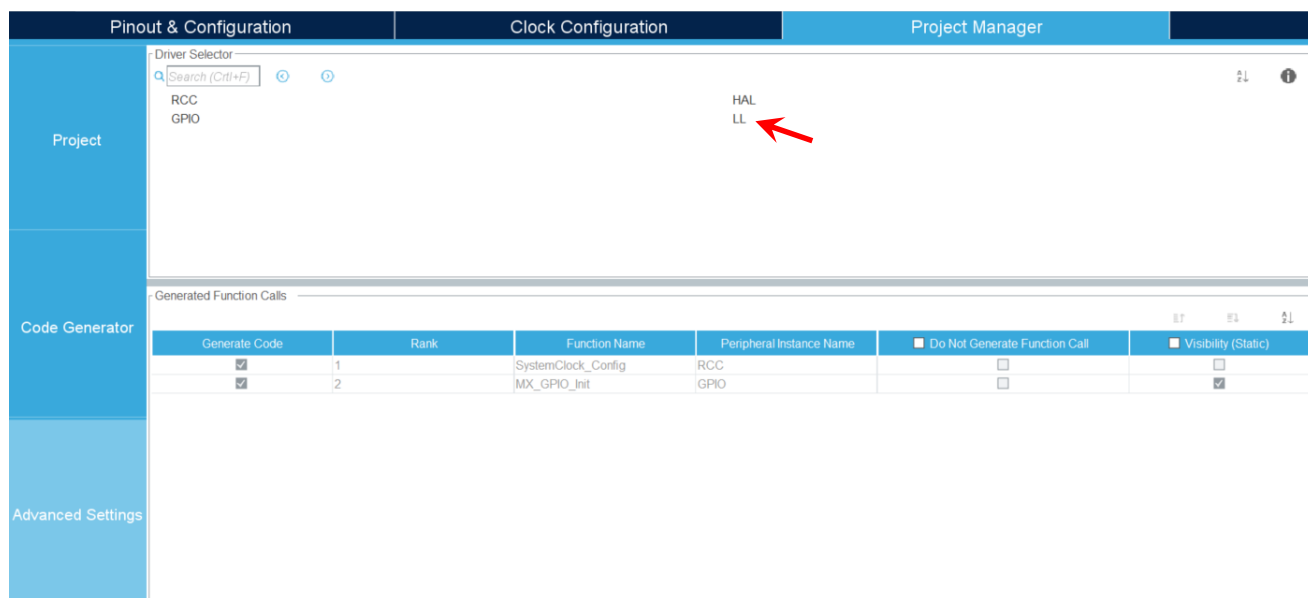
Namig: pri iskanju pinov znotraj CubeMX-a vam lahko močno pomaga iskalnik desno spodaj.

Namig: lastnosti uporabljenih GPIO pinov nastavljate znotraj rubrike "Pinout & Configuration → System Core → GPIO → Configuration". Več v poglavju spodaj.

8. Poskrbite, da se bo za GPIO modul uporabljala knjižnica "LL" (angl. low-level) in ne knjižnica HAL.

V večini primerov bomo za delo na strojnem nivoju uporabljali "LL" knjižnico, saj je implementacija njenih funkcij bolj enostavna in nazorna.

Za posamezno strojno opremo lahko izberete vrsto knjižnice znotraj CubeMX znotraj podokna "Project Manager → Advanced Settings → Driver Selector" tako, da kliknete na mesto, ki ga prikazuje puščica na spodnji sliki.



9. Shranite nastavitve v CubeMX in tako zgenerirajte avtomatsko kodo za vaš projekt.

S tem zadnjim korakom ste naredili nekaj zelo pomembnega:

poskrbeli ste, da se bo *dodala programska koda za inicializacijo vseh digitalnih izhodov*, ki jih bomo uporabljali za krmiljenje LEDic. Poskrbeli ste torej za *inicializacijo strojne opreme* in tako je vse pripravljeno, da pričnemo implementirati *funkcije za upravljanje LEDic* na višjem, *sistemskem nivoju*!

10. Preučite, katere nizko-nivojske funkcije bomo potrebovali za upravljanje LEDic.

Pri implementaciji *sistemskih* funkcij za upravljanje skupine LEDic si bomo še vedno pomagali z *nizko-nivojskimi* funkcijami (angl. low-level, LL), ki bodo poskrbele za spreminjanje stanja digitalnega izhoda, ki krmili posamezno LEDico.

Da bo implementacija sistemskih funkcij hitreje stekla, se je smiselno vnaprej podučiti, katere nizko-nivojske funkcije nam bodo prišle prav. S pomočjo [uradne dokumentacije za "LL knjižnico"](#) torej ugotovite, katere funkcijo bomo potrebovali za

- prižiganje LEDice,
- ugašanje LEDice,
- invertiranje stanja LEDice (angl. toggle),
- ugotavljanje trenutnega stanja LEDice.

Namig: potrebujete funkcije, ki upravljajo z digitalnimi izhodi, torej z "GPIO pini". In ne pozabite: uporabljamo LL knjižnico (in ne HAL knjižnice).

Ugotovitve bi bilo smiselno zbrati v tabeli sledeče oblike:

Kaj želimo doseči?	Katero LL funkcijo uporabimo?	Katere vhodne parametre funkcija potrebuje? Kakšnega tipa so?
prižig LEDice		
ugasnitev LEDice		
invertiranje stanja LEDice		
ugotavljanje trenutnega stanja LEDice	<code>LL_GPIO_IsOutputPinSet()</code>	

NE POZABITE DOMA PRIPRAVLJENI PROJEKT PRINESTI S SEBOJ NA VAJO!

To storite tako, da celotno projektno mapo stisnete v arhiv (npr. zip) in prinesete s seboj ta arhiv. Na vajah pa bomo pripravljene projekt le še uvozili.

Namigi

Iskalnik, ki vam pomaga najti pine na mikrokrmilniku



Nastavljanje lastnosti uporabljenih GPIO pinov

