

Osnove mikroprocesorske elektronike

Vaja 3: CubeIDE, knjižnica HAL in projekt "blinky"

Namen te vaje je, da se spoznate z *razvojnim orodjem STM32CubeIDE*, ki ga bomo tekom vaj uporabljali za programiranje MiŠKot. Spoznali boste tudi *HAL knjižnico*, ki vsebuje *funkcije za delo s strojno opremo* mikrokrmilnika [STM32G474QE](#), na katerem temelji vaš MiŠKo. Vaša naloga bo, da pripravite projekt "blinky", kjer boste s pomočjo HAL knjižnice sprogramirali MiŠKotovo LEDico tako, da bo utripala.

Priprava na vajo

1. Na domači računalnik si namestite razvojno okolje [STM32CubeIDE](#).

Razvojno okolje boste potrebovali na svojem računalniku, da boste lahko po potrebi laboratorijske vaje dokončali doma oziroma da se boste na laboratorijsko vajo delno pripravili že doma.

2. Poglejte si [video lekcijo](#) o "blinky" projektu na Nucleo razvojni ploščici.

V video lekciji je predstavljeno razvojno okolje STM32CubeIDE ter osnovno delo v tem okolju, ki je grobem sestavljeno iz:

- a) urejevalnika programske kode ter
- b) grafičnega vmesnika za urejanje nastavitev mikrokrmilnika (a.k.a. CubeMX).

Lekcija tudi na kratko predstavi idejo HAL knjižnice in koncept delitve programske kode na dve skupini:

- a) avtomatično generirana HAL programska koda ter
- b) uporabniška koda (angl. user code).

Naloge vaje

1. Ustvarite nov prazen projekt "blinky" za mikrokrmilnik STM32G474QE.

Pri tem pazite, da ne ustvarite projekta za *specifično* razvojno ploščico, pač pa zgolj za mikrokrmilnik STM32G474QE.

2. Poskrbite za ustrezno nastavitve sistemske ure mikrokrmilnika (angl. clock tree configuration).

Zahteve glede nastavitve ure so sledeče:

- a) *osnovni vir ure* je zunanji signal frekvence 8 MHz (pazite: zunanji *signal* in torej ne zunanji oscillator)
- b) *vir sistemske ure* (angl. system clock) naj bo izhod PLLCLK internega vezja s fazno sklenjeno zanko (angl. phase-locked loop, PLL)
- c) *vir ure za USB vodilo CK48* je izhod PLLQ vezja s fazno sklenjeno zanko
- d) frekvenca ure za USB vodila CK48 mora biti enaka 48 MHz
- e) *faktorje množenja in deljenja ure* nastavite tako, da bo frekvenca sistemske ure SYSCLK ter ostalih urinih signalov za notranje periferne enote *največja mogoča*

Namig: vir *zunanjega* urinega signala nastavite v grafičnem vmesniku CubeMX znotraj kategorije "System Core → RCC". Več o tem lahko preberete v poglavju z namigi.

Namig: da boste sploh lahko spreminjali nastavitve ure za USB vodilo, je potrebno v projektu začasno omogočiti USB periferno enoto. Več v poglavju z namigi.

3. Poskrbite za ustrezno nastavitve splošno-namenskega digitalnega izhoda (angl. general purpose input/output, GPIO) tako, da boste lahko krmilili LEDico LED0.

Pomagajte si [shemo električnega vezja MiŠKota](#), da ugotovite, na katero nožico je povezana LED0.

Nastavitve izhoda seveda opravite s pomočjo grafičnega vmesnika CubeMX.

4. Implementirajte programsko kodo za utripanje LEDice LED0.

Pri implementaciji uporabite ustrezne *funkcije HAL knjižnice* za delo z digitalnimi vhodi/izhodi (angl. GPIOs).

Pomagate si lahko z [uradno dokumentacijo HAL knjižnice](#), ki vsebuje seznam in opis vseh funkcij, ki so na voljo za upravljanje strojne opreme mikrokrmilnikov serije STM32G4.

Namigi

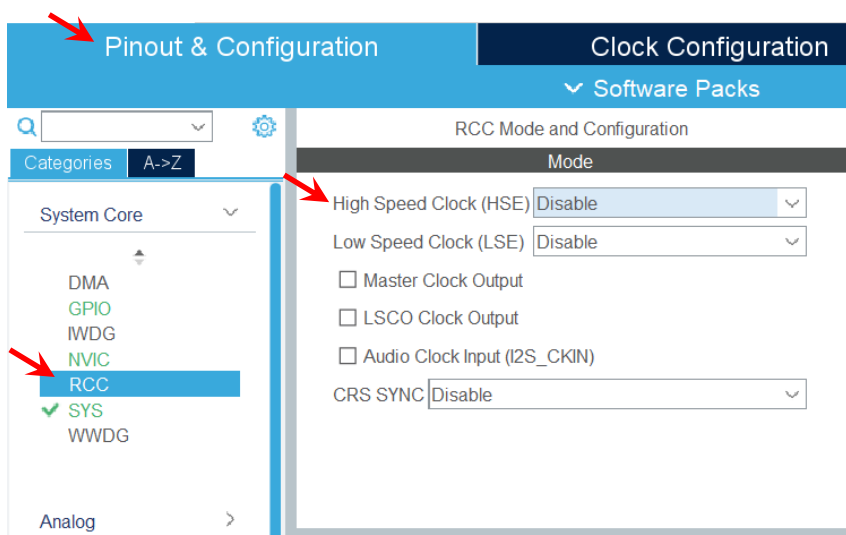
Nastavitev vira zunanjega urinega signala

Nastavitve, ki se tičejo *zunanjega* urinega signala (angl. external), urejate znotraj zavihka **"Pinout & Configuration"**.

To je smiselno, saj zunanji urin signal vstopa v mikrokrmilnik preko priključka (angl. pin, v žargonu pina).

Ker je urin signal ključen za delovanje jedra mikrokrmilnika, spadajo nastavitve ure v kategorijo **"System Core"**.

In ker se del periferije, ki upravlja z urinimi signali, imenuje **"Reset and Clock Control"**, najdete nastavitve ure znotraj sklopa RCC. Glejte spodaj.

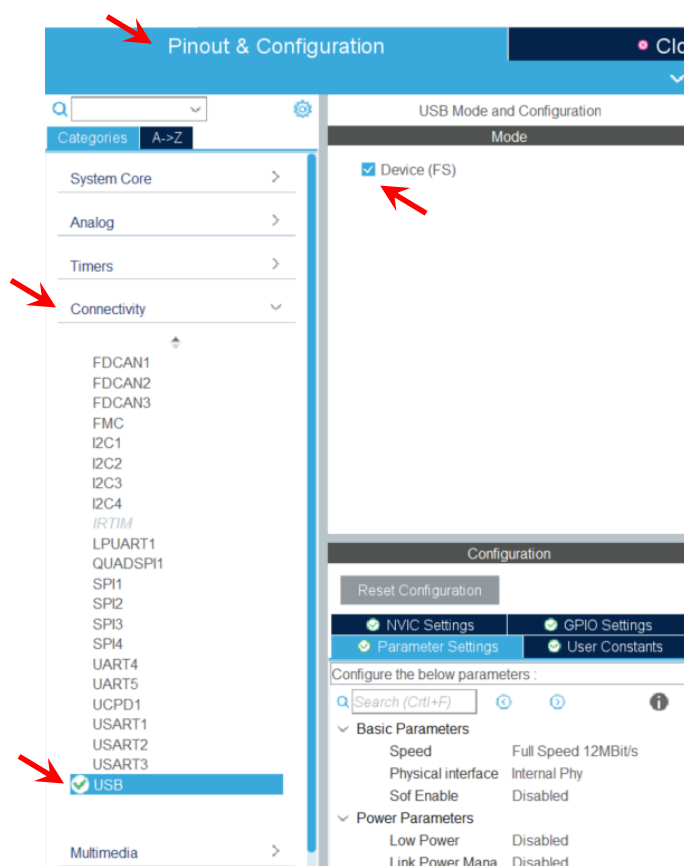


Začasen vklop USB periferne enote

Da boste lahko spreminjali nastavitve v zvezi z uro USB vodila, je potrebno v projektu začasno omogočiti USB periferno enoto. To storite s pomočjo CubeMX grafičnega vmesnika v zavihku

"Pinout & Configuration"

v kategoriji "Connectivity". Glejte spodaj.



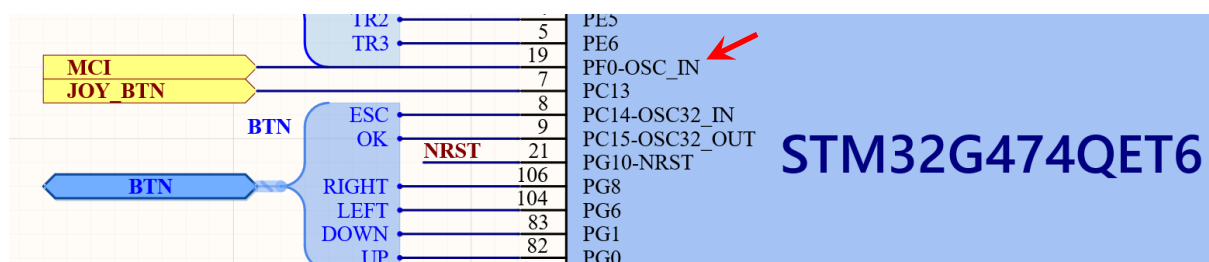
Slika 1 - začasno vklopimo USB enoto, da lahko spreminjamo nastavitve ure USB vodila

Ko boste uspešno nastavili generiranje vseh urinih signalov, pa USB periferno enoto spet onemogočite. USB enoto bi načeloma lahko pustili tudi omogočeno, vendar bi nam potem CubeMX avtomatično generiral programsko kodo za inicializacijo USB modula, ki pa je pri prvi vaji ne potrebujemo.

Dodatna pojasnila

Od kje pride 8 MHz zunanji vir ure?

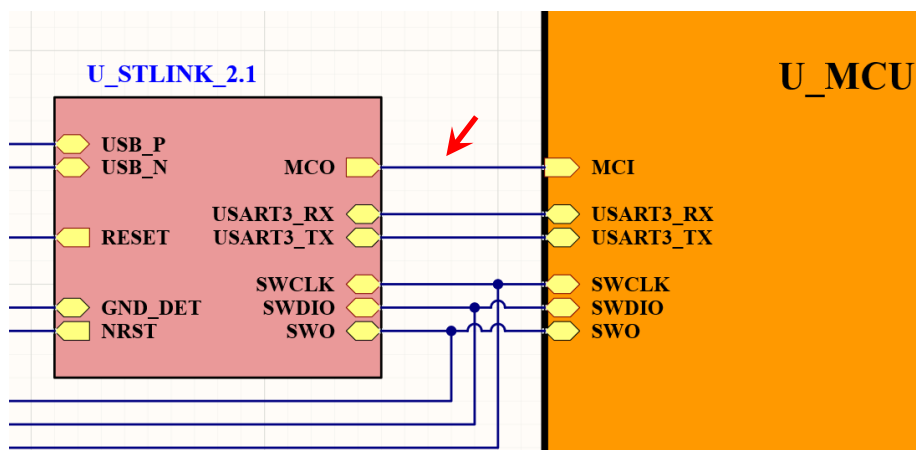
Če pogledate v električno shemo MiŠKota in se osredotočite na nožice mikrokrmilnika, ki so namenjene zunanjemu viru ure, boste našli sledečo sliko:



Slika 2 - zunanji urin signal je pripeljan na priključek PF0

Vidite, da je na nožico PF0-OSC_IN pripeljan zunanji signal MCI.

In če sledite poteku tega signala MCI, ugotovite, da ta signal pride iz mikrokrmilnika, ki je odgovoren za ST-LINK vmesnik.

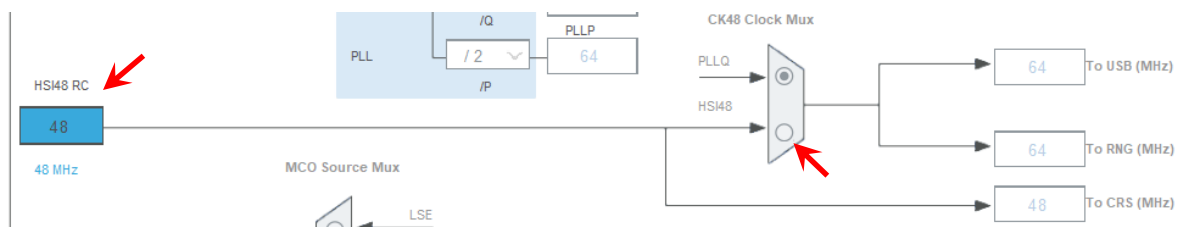


Slika 3 - ST-LINK modul poskrbi za zunanji vir urinega signala našega mikrokrmilnika

Na ta način dosežete, da vaš sistem potrebuje le en sam kristalni oscilator, katerega signal nato vodite v preostale dele sistema, ki potrebujejo stabilen vir urinega signala.

Zakaj za vir urinega signala USB vodila ne uporabimo notranjega oscilatorja?

Zakaj zapletamo nastavitve urinih signalov in zahtevamo, da se za vir urinega signala USB vodila uporabi signal fazno sklenjenega vezja PLLQ? Zakaj ne bi enostavneje uporabili kar notranjega hitrega RC oscilatorja HSI48 (glejte sliko spodaj), ki že zagotavlja 48 MHz urin signal, ki ga potrebuje USB vodilo?



Slika 4 - zakaj ne bi za uro USB vodila uporabili kar notranji hitri 48 MHz RC oscilator?

Razlog je v tem: veliko preprostejših mikrokrmilnikov nima na voljo takega hitrega notranjega RC oscilatorja. In zato je potrebno pri takih mikrokrmilnikih uporabiti kot vir ure USB vodila signal PLL vezja. In zato je prav, da se naučite nastaviti uro sistema tudi v takih primerih.