Osnove mikroprocesorske elektronike

Vaja 4: Projekt "blinky" z registri

Namen te vaje je, da preizkusite, kako je mogoče *upravljati mikrokrmilnik neposredno preko registrov*. Pri tej vaji še vedno ostajamo pri projektu "blinky", le da bomo tokrat utripanje LEDice sprogramirali *brez pomoči funkcij HAL knjižnice*. Tako boste hkrati tudi dobili boljši občutek o tem, kakšno vlogo ima HAL knjižnica pri programiranju mikrokrmilnikov.

Priprava na vajo

1. ***Obnovite znanje o uporabi bitnih operacij in maskiranja.***

   To znanje bose namreč potrebovali, ko bo potrebno *manipulirati s posameznimi biti v registrih* mikrokrmilnika.

   Pomagajte si z gradivom "Bitni in logični operatorji v jeziku C". Za rešitev te vaje je ključno gradivo, ki je podano *na straneh od 5 do 29*.

2. ***Preučite, katere registre mikrokrmilnika bo potrebno uporabiti***, če želimo priključek (tj. pin), kamor je priklopljena LEDica `LED0` uporabiti kot izhod. Preučite tudi, *katere bite teh registrov* bo potrebno nastavljati ter *kakšne morajo biti njihove vrednosti*.

   Vse potrebne informacije si zabeležite v tabelo, da vam bodo prišle prav pri implementaciji potrebne programske kode na vajah.

Kaj želimo nastaviti?	Kateri register uporabimo?	Kateri bit je potrebno nastaviti?	Kakšna mora biti vrednost tega bita/bitov?
omogočiti uro portu `GPIOF`			
nastaviti vrsto digitalnega izhoda na "push-pull" izhod			
onemogočiti "pull-up/pull-down" upore			
nastaviti hitrost izhoda na "low"			
nastaviti pin kot digitalni izhod			
nastaviti stanje dig. izhoda na 1			
nastaviti stanje dig. izhoda na 0			
invertirati stanje dig. izhoda (angl. toggle)			

Namig: če želite invertirati stanje digitalnega izhoda, morate poznati *trenutno* stanje digitalnega izhoda. Kateri register hrani *trenutno* stanje digitalnega izhoda? In še: ali lahko taisti register uporabite tudi za določitev *novega invertiranega* stanja?

Namig: invertiranje posameznega bita v registru lahko elegantno dosežete z bitno XOR operacijo.

Naloge vaje

1. ***Pripravite nov projekt "blinky", kjer pa onemogočite avtomatsko generiranje HAL kode*** za digitalne vhode/izhode.

   Pri tem "blinky" projektu bomo preprečili, da bi CubeMX avtomatsko generiral programsko kodo za inicializacijo digitalnega izhoda s HAL funkcijo `MX_GPIO_Init()`, saj želimo to kodo implementirati sami.

   Namig: v grafičnem vmesniku CubeMX pojdite pod zavihek "Project Manager" in izberite rubriko "Advanced Settings", kjer boste v podoknu "Generated Function Calls" našli možnost za izklop generiranja kode ("Generate code") za funkcijo `MX_GPIO_Init()`. Več v poglavju z namigi.

2. ***Definirajte prototipe funkcij***, ki jih nameravamo implementirati za upravljanje `LED0`:

   ```
   void LED0_GPIO_init(void);
   void LED0_on(void);
   void LED0_off(void);
   void LED0_toggle(void);
   ```

   Prototipe definirajte znotraj `main.c` datoteke znotraj sekcije

   ```
   Private function prototypes
   ```

3. ***Implementirajte funkcijo*** **`LED0_GPIO_init()`,** kjer poskrbite za naslednje stvari:

   1. omogočite uro za port `GPIOF`
   2. nastavite začetno stanje digitalnega izhoda za krmiljenje `LED0` na nizko stanje (tj. `LED0` bo na začetku po inicializaciji ugasnjena)
   3. nastavite vrsto digitalnega izhoda na "push-pull" izhod
   4. za digitalni izhod onemogočite "pull-up" in "pull-down" upore
   5. nastavite hitrost digitalnega izhoda na "nizko" (angl. low)
   6. nastavite `LED0` priključek kot digitalni izhod

   Programsko kodo umestite v sekcijo

   ```
   Private user code
   ```

   Namig: zgornje *korake, ki si potrebni za inicializacijo* GPIO pina, *uporabite kot komentarje* znotraj funkcije `LED0_GPIO_init()`. Pod vsakim takim komentarjem nato spišite programsko kodo, ki je nujna za implementacijo posameznega koraka. Tako hitro zagotovite nujno potrebno dokumentacijo vaše programske kode, hkrati pa imate vodilo pri programiranju.

Implementirajte še funkcije, ki bodo *upravljale s stanjem* LEDice `LED0`.

4. ***Implementirajte funkcijo*** **`LED0_on()`**, kjer poskrbite za *vklop* LEDice `LED0`.

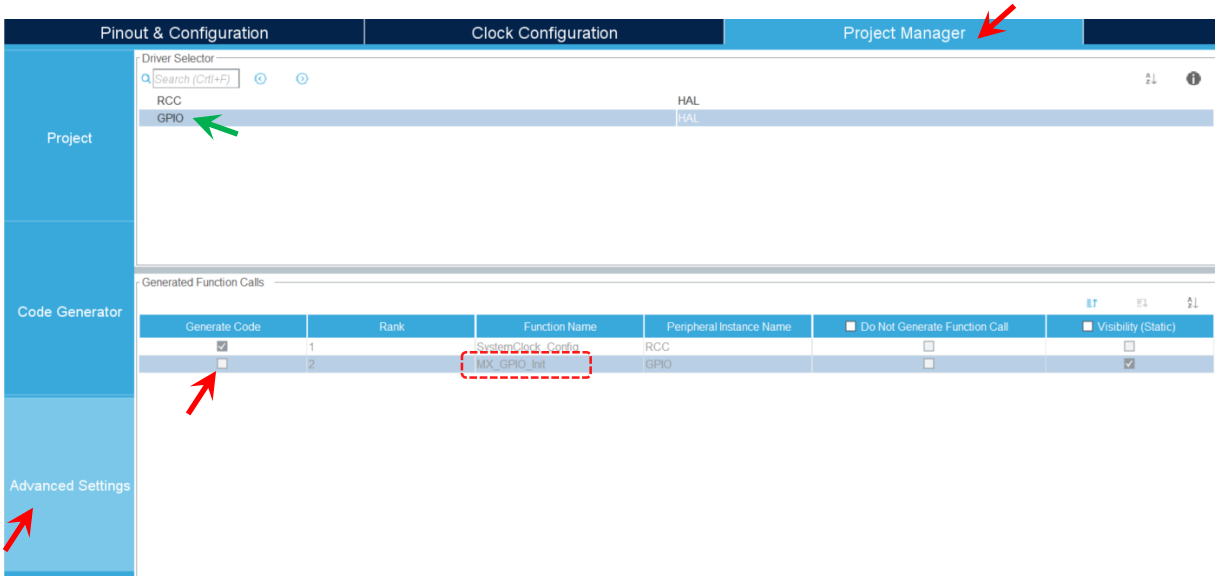
5. ***Implementirajte funkcijo*** **`LED0_off()`**, kjer poskrbite za *izklop* LEDice `LED0`.

6. ***Implementirajte funkcijo*** **`LED0_on()`**, kjer poskrbite *spremembo stanja* LEDice `LED0`.

Namigi

Izklop avtomatskega generiranja programske kode s HAL knjižnico

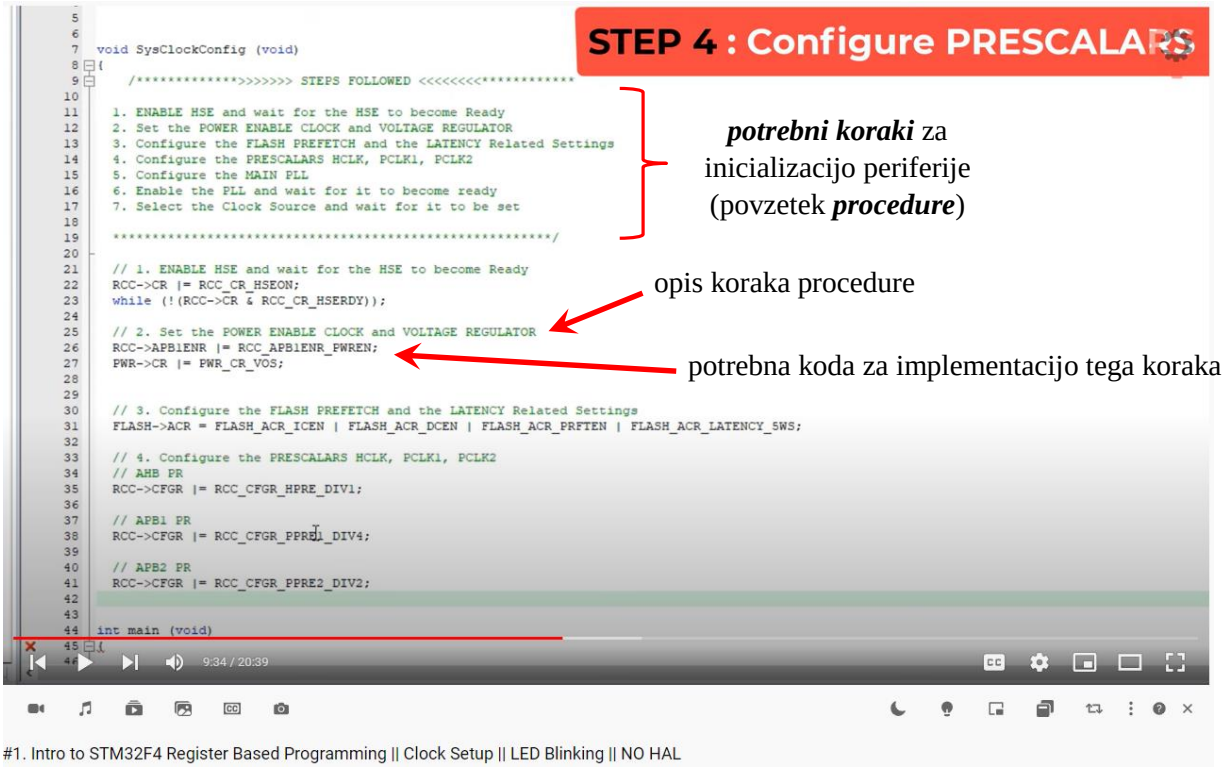
V grafičnem vmesniku CubeMX pojdite pod zavihek "Project Manager" in izberite rubriko "Advanced Settings", kjer boste v podoknu "Generated Function Calls" našli možnost za izklop generiranja kode ("Generate code") za funkcijo `MX_GPIO_Init()`. Poskrbite še, da imate izbrani GPIO modul (zelena puščica).



Slika 1 - podrobnejši napotki za izklopitev avtomatskega generiranja kode za funkcijo `MX_GPIO_Init()`

Opis ključnih korakov procedure naj postane komentar, ki vas vodi

Zelo ugodno je, če proceduro, ki jo programirate, na kratko povzamete s kratkimi opisi ključnih korakov. Opisi teh korakov nato lahko postanejo komentarji, ki vas vodijo pri implementaciji celotne procedure. Poglejte idejo spodaj. Tako dobite komentarje, ki vas po eni strani vodijo pri implementaciji, po drugi strani pa služijo kot opis in dokumentacija vaše procedure.



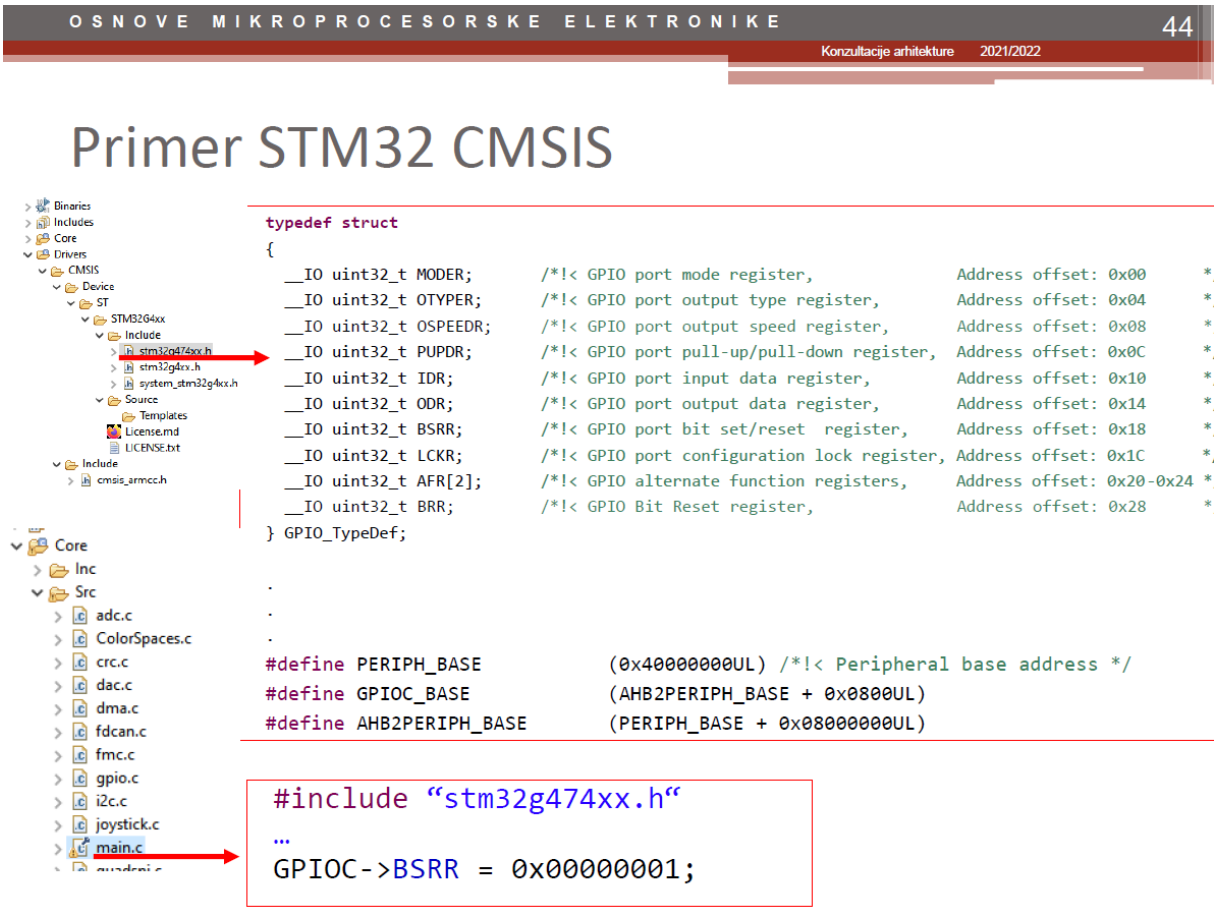
Slika 2 - povzetke ključnih korakov pri inicializaciji uporabite kot komentarje, pod katerimi nato implementirate potrebno programsko kodo (vir: youtube)

Dodatna pojasnila

Še vedno si pomagamo z ARM knjižnicami

S tem, ko smo izklopili avtomatsko generiranje kode za funkcijo `MX_GPIO_Init()`, smo se odločili, da *za delo z digitalnimi vhodi in izhodni ne bomo uporabili funkcij iz HAL knjižnice*. Vendar pa to ne pomeni, da si pri programiranju ne bomo pomagali z že obstoječimi knjižnicami!

Čeprav bomo digitalni izhod *upravljali neposredno preko registrov mikrokrmilnika*, nam bodo pri tem še vedno pomagale *CMSIS knjižnice za delo z mikrokrmilnikom STM32G474QE*. Namreč, poskrbele bodo, da bomo lahko *zelo elegantno dostopali do registrov*, ki upravljajo z GPIO periferijo. Za osvežitev koncepta si poglejte sliko s konzultacij spodaj.



Slika 3 –CMSIS knjižnice za delo z mikrokrmilnikom STM32G474QE bodo poskrbele za elegantno dostopanje do registrov mikrokrmilnika v smislu dela s strukturami.