Vaja 13 – 7 segmentni prikazovalnik z STM32L1 Discovery

1. **Cilj naloge**: S pomočjo programskega okolja **STM32CubeIDE** in HAL knjižnicami sprogramirajte STM32L1 mikroprocesor tako, da bo krmilil 7-segmentni LED prikazovalnik. Prikazovalnik bo štel desetiška števila od 0 do 9 v razmaku 1 sekunde.

1. **Postopek inicializacije periferije**.
   1. Zaženite **STM32CubeIDE** in ustvarite nov STM32 projekt (pod zavihkom *information Center*). V zavihku *Board selector* s pomočjo filtrov Vendor, Type in MCU/MPU izberite ustrezno razvojno ploščo, kliknite *Next,* projekt poimenujte **vaja13\_7-segment** in kliknite *Finish* (na možnosti opcije za prenastavitev periferije izberite *Yes*, izbrana naj bo tudi opcija perspektive za STM32CubeMX).
   2. V *Pinout & Configuration* pogledu glede na vašo razvojno ploščico postavite 7 prvih prostih pinov **skupine A** na GPIO\_Output. Pine označite *(label)* s črkami segmenta: ***a, b, c, d, e, f*** in ***g***.

Zapišite izbrane (konfigurirane) pine:

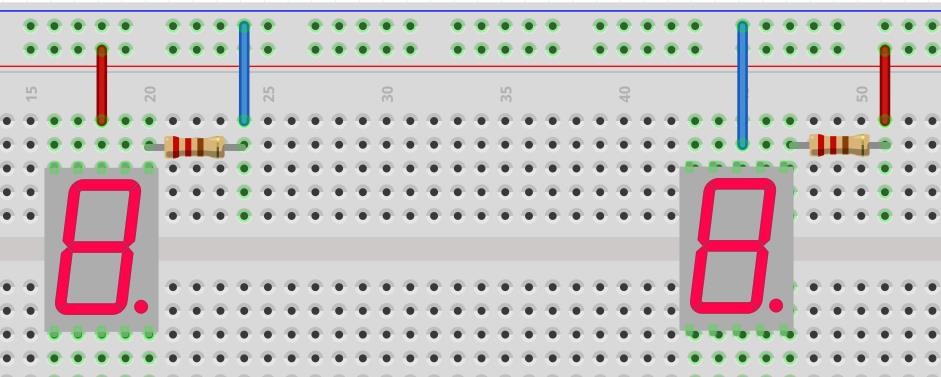
Segment **a**: \_\_\_\_\_

Segment **b**: \_\_\_\_\_ Segment **e**: \_\_\_\_\_

Segment **c**: \_\_\_\_\_ Segment **f**: \_\_\_\_\_

Segment **d**: \_\_\_\_\_ Segment **g**: \_\_\_\_\_

* 1. 7-segmentni display je lahko v izvedbi skupne **anode** ali skupne **katode.** Vaš prikazovalnik testirate tako, da ga zvežete po spodnji (levi ali desni) shemi in preizkusite, katera polariteta je ustrezna (rdeča 5V, modra GND):



Če zasveti

segment, je to

skupna ANODA

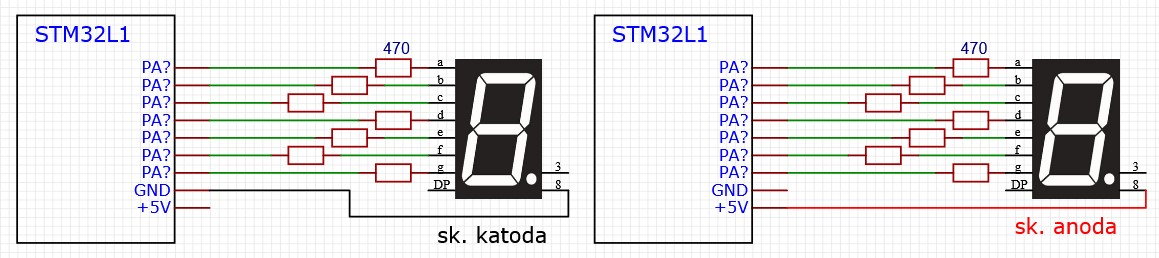
Če zasveti

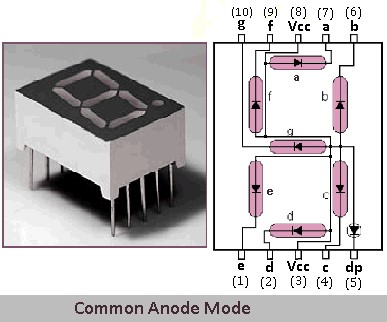
segment, je to

skupna KATODA

Vaš prikazovalnik je s skupno \_\_\_\_\_\_\_\_\_\_\_\_. Preverite delovanje še preostalih segmentov!

* 1. Ustrezno povežite pine STM32L1 ploščice preko uporov na prikazovalnik. Za napajanje prikazovalnika uporabite + 5 V (sk. anoda) ali GND (sk. katoda) same ploščice.

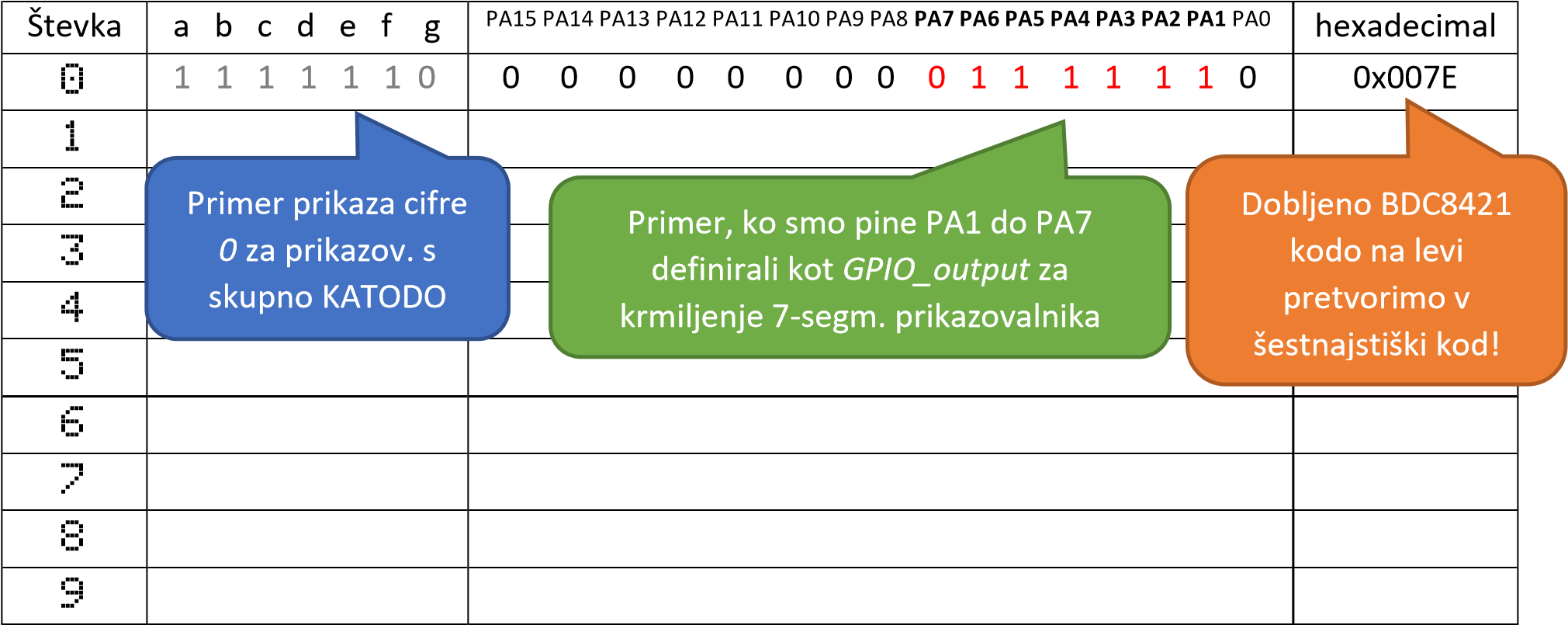




Na desni sliki vidimo postavitev pinov prikazovalnika z ustrezno oznako. Sredinski - (3) in (8) – sta pina za skupno anodo/katodo. Ko krmilimo segmente, **uporabimo VEDNO upore** (470 Ω), sicer uničimo segment (ter mikroprocesor na ploščici).

Primer krmiljenja – če želimo, da pri skupni ANODI zasveti števka 4, bomo na segmente **b, c, f** in **g** postavili *nizko logično stanje O (* (GND), na segmente **a**, **d** in **e** pa *visoko logično stanje 1* (+ 5 V). Skupna anoda (pina 8 ali 3) sta povezana na 5 V.

* 1. Glede na izvedbo vašega prikazovalnika (sk. anoda / sk. katoda) ustrezno **popravite** in **izpolnite** spodnjo pravilnostno tabelo za števke od 0 do 9:



* 1. Sedaj generirajte kodo tako, da enostavno kliknete ikono *Save* in po potrebi še enkrat potrdite generiranje kode.

1. **Programiranje v** **IDE:**
   1. V skrajno levem oknu *Project Explorer* poiščemo **main.c** datoteko pod *Core* → *Src* → *main.c* (dvokliknite na datoteko, odpre se tekstovni urejevalnik za main.c).
   2. V /\* USER CODE BEGIN PV \*/ definiramo *polje (ang. Array),* ki predstavlja za vsako desetiško števko ustrezno kombinacijo stanj na GPIO vratih skupine A, na katere imamo priključene segmente prikazovalnika:

**int** LEDS[] = {0x00BE, 0x??, 0x??, 0x??, 0x??, 0x??, 0x??, 0x??, 0x??, 0x??};

?? ustrezno zamenjajte s heksadecimalnim zapisom, ki ste ga zapisali v tabeli.

* 1. V /\* USER CODE BEGIN 1 \*/ definiramo spremenljivko za štetje *Count*:

**int** Count = 0;

* 1. V neskončni zanki /\* USER CODE BEGIN 3 \*/zapišite naslednjo kodo:

GPIOA -> ODR = LEDS[Count]; Count++; **if** (Count == 10) Count = 0;

HAL\_Delay(1000);

* 1. Kaj dela ukaz GPIOA -> ODR ?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Naložitev kode (*Run*) in opazovanje delovanja na STM32L1.**
2. Kodo preverite s tipko ***Build*** (ikona za kladivce - *Debug*). Ko je preverjanje končano lahko preverimo, če smo med pisnjem kode naredili kakšno napako sintakse, sicer se pod kodo v oknu *Console* izpiše 0 errors.
3. Priklopite STM32L1 na vaš računalnik preko USB kabla.
4. S tipko ***Run***(zelena ikona za play – *Run as STM…*)prenesete program na STM32L1.
5. V oknu *Edit Configuration* kliknemo OK. Nekaj sekund bo na ploščici STM izmenično utripala zelena in rdeča LED, ko je program naložen, sveti LED rdeče.
6. Opazujte delovanje 7-segmentega prikazovalnika. Kaj se zgodi, ko pritisnemo črno tipko na STM32L1?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Delovanje posnamite s telefonom.

5. **Vaš projekt** (datoteko *main.c* [25%], slikovni izrezek *Pinout* mikroporcesorjaiz CubeMX [15%]*,* kratek videposnetek delovanja [20%], **naložite v Github** kot nov *Repository* z imenom **Vaja13-7segment-STM32L1** [5%]**.** V *readme* datoteki zapišite vse **odgovore** [30%] na vprašanja v *Readme* ter **komentar** [5%] na delovanje. **Oceno** pridobite iz Github dokumentacije!