

VEŽBA 2

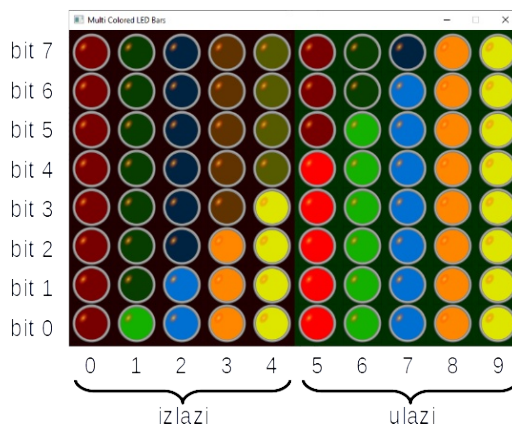
Osnovna namena mikrokontrolera je upravljanje uređajima i procesima. Da bi se ostvario pristup uređajima iz okruženja, mikrokontroleru je omogućeno da postavlja (upisuje) izlazne fizičke veličine, kao i da utvrđuje (čita) stanje ulaznih veličina. Ove fizičke veličine su uobičajeno ograničene na napone koji odgovaraju naponskim nivoima digitalnih veličina – jedinica i nula. Uređaji preko kojih se ovo ostvaruje nazivamo ulazno/izlaznim kapijama (*engl. input/output – I/O – ports*).

Simulirana svetla u boji

Grafička aplikacija za Windows platformu **LED_bars** omogućava simulaciju ulaznih i izlaznih digitalnih veličina. Svetla su predstavljena kao vertikalni stupci od 8 svetala u istoj boji, a moguće je simulirati do 10 stubaca od kojih svaki može biti ulaz ili izlaz. U ulaznim stupcima (prepoznaju se po tamno zelenoj podlozi) svetlima je moguće promeniti stanje levim ili desnim klikom pomoću miša. Svaki stubac može imati svetla u crvenoj (R/r), zelenoj (G/g), plavoj (B/b), žutoj (Y/y) ili narandžastoj (O/o) boji. Aplikacija se poziva komandom:

```
LED_bars RGOYrgboy
```

Svaki znak u stringu koji sledi iza komande određuje boju i tip odgovarajućeg stupca. Malo slovo odgovara ulaznom, a veliko slovo izlaznom stupcu. Zadana komanda će za rezultat imati sledeću konfiguraciju:



Dozvoljeno je zahtevati manje od 10 stubaca. Ako se ne navede ništa, rezultat će biti 5 stubaca različitih boja.

Biblioteka za upravljanje svetlima

Stupci su numerisani brojevima od 0 do 9. Svakim svetlom upravlja jedan bit (digitalna cifra 1 ili 0) unutar grupe od 8 bita (bajta). Iz programa se svetlima upravlja pozivanjem funkcija iz biblioteke `HW_access.lib` koja treba da bude uključena u proces prevođenja programa¹.

¹ U okruženju Visual Studio, svakom projektu treba pridružiti biblioteke `HW_access.lib` i `wsock32.lib`. Drugi fajl je deo razvojnog okruženja, dok se prvi mora ubaciti među fajlove projekta ručno, a može se naći kao arhiva na sajtu sa vežbama.

Funkcije za upravljanje svetlima

<code>int init_LED_comm();</code>	Ovu funkciju je neophodno pozvati pre bilo koje od ostalih. Ukoliko se to ne učini, poziv nijedne od funkcija za upravljanje svetlima neće imati efekta
<code>int close_LED_comm();</code>	Pre završetka rada programa potrebno je pozvati ovu funkciju. Nakon poziva ove funkcije, poziv nijedne od funkcija za upravljanje svetlima neće imati efekta
<code>int set_LED_BAR(uint8_t b, uint8_t d);</code>	Ova funkcija postavlja 8-bitnu vrednost d u registar koji kontroliše stubac broj b (od 0 do N-1, gde je N broj stubaca). Upis će imati efekta samo na stupce koji su podešeni kao izlazni (TYPE_IN)
<code>int get_LED_BAR(uint8_t b, uint8_t* d);</code>	Ova funkcija čita 8-bitnu vrednost iz registra koji kontroliše stubac broj b (od 0 do N-1, gde je N broj stubaca). Parametar d pokazuje na promenljivu tipa uint8_t gde će biti upisan pročitani sadržaj. Dejstvo je isto bez obzira na tip stupca (TYPE_OUT ili TYPE_IN)
<code>int set_LED_BAR_type(uint8_t b, uint8_t d);</code>	Pozivom ove funkcije, postavlja se tip izabranog stupca na izlaz ili ulaz u skladu sa parametrom b. Time se redefiniše postavka iz string argumenta simulatora. Parametar d ima i svoje definisane makroe TYPE_OUT – (0) i TYPE_IN – (1)
<code>void Delay_ms(uint32_t ms);</code>	Poziv ove funkcije izaziva pasivno čekanje u trajanju od ms (parametar funkcije) milisekundi. Strogo uzevši, ovo nije funkcija za upravljanje svetlima, ali je neophodna za vremensku sinhronizaciju programa.

Da bi se funkcije mogle pozivati iz programa, neophodno je uključiti heder fajl `HW_access.h` koja je deo arhive sa bibliotekom. Sve funkcije koriste celobrojne tipove podataka definisane pomoću heder fajla `stdint.h`. Broj uz svaki tip označava dužinu tipa u bitima.

Primer programa za upravljanje svetlima

Na početku svakog programa, pre nego što se obavi ijedan pristup svetlima, neophodno je pozvati funkciju `init_LED_comm`. Bez toga, pristup neće biti uspešan.

```
uint8_t br;

init_LED_comm(); // inicijalizacija veze sa svetlima

while(1)
{
    get_LED_BAR(0, &br);
    // ovde se obradjuje ocitana vrednost br
    // i odlucuje se o narednom izlazu
}
```

```
set_LED_BAR(1, br);
Delay_ms(100); // kratka pauza, brzi rad nije potreban
/* ovo je specificno za simulirano okruzenje
 * u realnim uslovima se iz
 * petlje ne izlazi nikad */
if(br == 0x0F)
{
    break;
}
}
close_LED_comm(); // veza se moze prekinuti na kraju
```

Zadaci

1. Napisati program koji neprestano očitava stanje na prvom stupcu (stubac 0, ulazni) i to isto stanje postavlja na drugom stupcu svetala (stubac 1, izlazni). Program treba da se završi (izađe iz glavne petlje) kada se uključe sva svetla u prvom stupcu. Testirati program sa odgovarajućim brojem simuliranih stubaca.
2. Proširiti prethodni program tako da u trećem stupcu postavlja svetla invertovano u odnosu na prvi (i drugi) stubac – upaljeno svetlo na mestu ugašenog i obratno. Testirati program sa odgovarajućim brojem odgovarajuće podešenih simuliranih stubaca.
3. Proširiti prethodni program tako da u četvrtom stupcu postavlja svetla u suprotnom redosledu u odnosu na prvi. Testirati program sa odgovarajućim brojem simuliranih stubaca.

Dodatni neobavezni zadatak:

4. Napisati program koji na 10 stubaca iscrtava (približnu) sinusoidu čija jedna perioda odgovara broju stubaca. Svakih 300 ms pomeriti sinusoidu u levu stranu tako da se dobije utisak kontinualnog talasa koji se pomera sa desne ka levoj strani. Iz glavne petlje se izlazi nakon 20 punih ciklusa.