

Prekidi, prekidni potprogrami i rutine

U ovoj vježbi upoznat ćemo se s prekidima (engl. *interrupt*): glavnim mehanizmom “istovremenog izvođenja” operacija na Arduinou. Prekidi funkcioniraju na način da se, po specificiranom događaju (istekom vremena, promjena na digitalnom ulazu i slično) poziva funkcija koju sami specificiramo. Da bi se funkcija pozvala, potrebno je **samo jednom** (hint: funkcija *setup*) specificirati na koji će događaj koja funkcija pozivati. Funkciju koju pozivamo **mora** imati navedeni prototip:

```
void proizvoljnoImeFunkcije();
```

što ujedno znači da funkcija ne može primiti nikakve parametre. Ukoliko želimo u takvu funkciju prenijeti kakav kontekst, potrebno je to obaviti globalnim varijablama (postoje i napredniji mehanizmi, ali je u svrhu radionica a i općenito Arduino projekata nepotrebno).

Kako bi si bolje predložili **što** rade prekidi, pogledajmo slijedeći isječak kôda:

```
void loop() {  
    digitalWrite(LEDICA, HIGH);  
    delay(500);  
    digitalWrite(LEDICA, LOW);  
    delay(500);  
}
```

Pomislili bi da jedino što program radi jest pali i gasi ledicu s periodom od oko sekunde. Međutim, ukoliko je u funkciji *setup* podešeno da se prekidna rutina pali na npr. pritisak gumba, čak i ako je naš glavni dio programa jednostavan, dodatna funkcionalnost se transparentno može ostvariti prekidima.

U ovoj vježbi fokus će biti na prekide inicirane ponašanjem na digitalnom ulazu. U slijedećim vježbama, obradit ćemo i prekide inicirane istekom vremena, periodične prekide. Bitno je napomenuti kako svaki mikrokontroler ima točno specificirane pinove na kojima može inicirati prekid. Googlanjem “arduino attachinterrupt” saznajte koje pinove i kako ih možete koristiti.

Također, bitno je napomenuti kako neke operacije nije moguće koristiti unutar funkcija pozivanim interruptom. Tokom vježbe spomenut ćemo nekoliko takvih funkcija, ali za početak proučite nekoliko bitnih funkcija koje ćete koristiti tokom ove vježbe.

1. `attachInterrupt()`
2. `detachInterrupt()`
3. `interrupts()`
4. `noInterrupts()`
5. `digitalPinToInterrupt()`

Za sva pitanja obratite se voditelju radionice. Kada svladate gore navedeno gradivo (obično je dovoljno proučiti službene arduino stranice) prebacite se na zadatke. Nadalje, prije nego se bacite na zadatke, uputno je ponoviti i gradivo o *pullup* i *pulldown*-u.

Zadatak 1: bez korištenja vanjskog otpornika, napravite program čija je prekidna rutina slijedeća:

```
void paliLED() {  
    digitalWrite(LED_BUILTIN, HIGH);  
    while(1);  
}  
  
// u setup je potrebno dodati i slijedeću liniju koda:  
// pinMode(LED_BUILTIN, OUTPUT);
```

Zadatak 2: umjesto trajne while petlje, u prekidnu rutinu dodajte funkciju *delay* s jednom sekundom.

Zadatak 3: pomoću jedne prekidne rutine, implementirajte program koji će na pritisak gumba paliti i gasiti LED diodu. Stanje LED-ice pohranjujte korištenjem globalne varijable. Pritom proučite ključnu riječ *volatile* i prikladno je iskoristite.

Zadatak 4: isti program implementirajte pomoću dvije prekidne rutine.

Zadatak 5: probajte u prekidnoj rutini ispisati ispis na serijski monitor.

Zadatak 6: Implementirajte igru s jedne od prošlih radionica:

Napravi program koji će, nakon paljenja programa, po isteku slučajnog broja sekundi između 3 i 9, upaliti LED na pinu 13. Po paljenju LED-ice na pinu 13, dva korisnika moraju stisnuti gumb čim je prije moguće (svaki korisnik ima svoj gumb). Program ispisuje pobjednika u igri. Ukoliko jedan od igrača pritisne dugme prije paljenja LED-ice, ispisuje se da je korisnik gubitnik i igra završava.

Po potrebi modificirajte program na način prikladan korištenju pomoću prekidnih rutina.

Zadatak 7: Komunikacijom globalnim varijablama implementirajte brojač klikova u sekundi.