

TEHNIČKA ŠKOLA ŽUPANJA

LABORATORIJSKE VJEŽBE
AUTOMATSKO VOĐENJE PROCESA

LABORATORIJSKA VJEŽBA

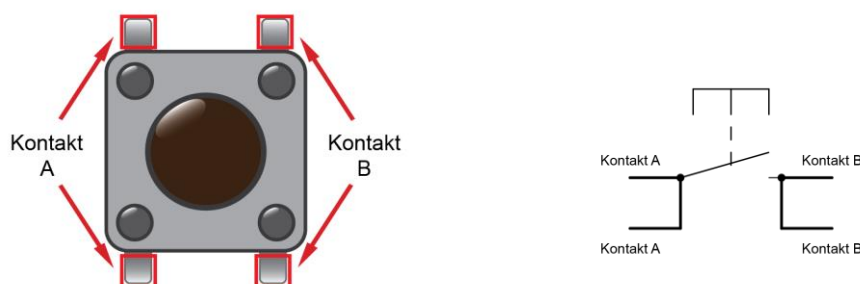
- TIPKALA I LED -

ŽUPANJA 2017

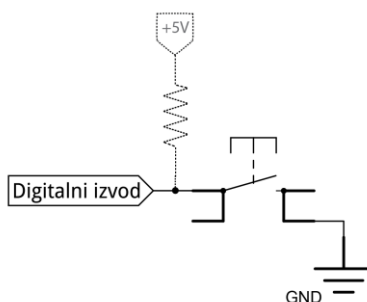
U ovoj vježbi naučit ćemo ponešto o najjednostavnijim elektroničkim komponentama koje će nam služiti za davanje ulaznih podataka na mikrokontroler – tipkalima. Za razliku od dosad upoznatih komponenata, tipkalima nećemo upravljati pomoću mikrokontrolera, već ćemo s njih očitavati podatke – je li tipkalo pritisnuto ili otpušteno.

Tipkala

Postoji puno raznih izvedbi tipkala. Na sljedećoj slici prikazana je jedna vrsta tipkala te njegov simbol u Fritzing programu.



Kada je tipkalo otpušteno ono ne provodi struju, a kada je tipkalo zatvoreno ono provodi struju. Osnovni način spajanja tipkala na mikrokontroler prikazan je na sljedećoj slici:



Otpornik prikazan na slici točkastom crtom naziva se pull-up otpornik i nalazi se u mikrokontroleru, a služi za definiranje stanja digitalnog izvoda kada je tipkalo otpušteno.

Kada je tipkalo otpušteno izvod mikrokontrolera spojen je preko otpornika na napon napajanja od 5V. Kako na tom otporniku nema pada napona na izvod mikrokontrolera je spojeno 5V i očitavamo visoko logičko stanje. Kada je tipkalo pritisnuto izvod mikrokontrolera spojen je direktno na masu napajanja i tada očitavamo nisko logičko stanje.

Znači:

Pritisnuto tipkalo – očitavamo stanje LOW

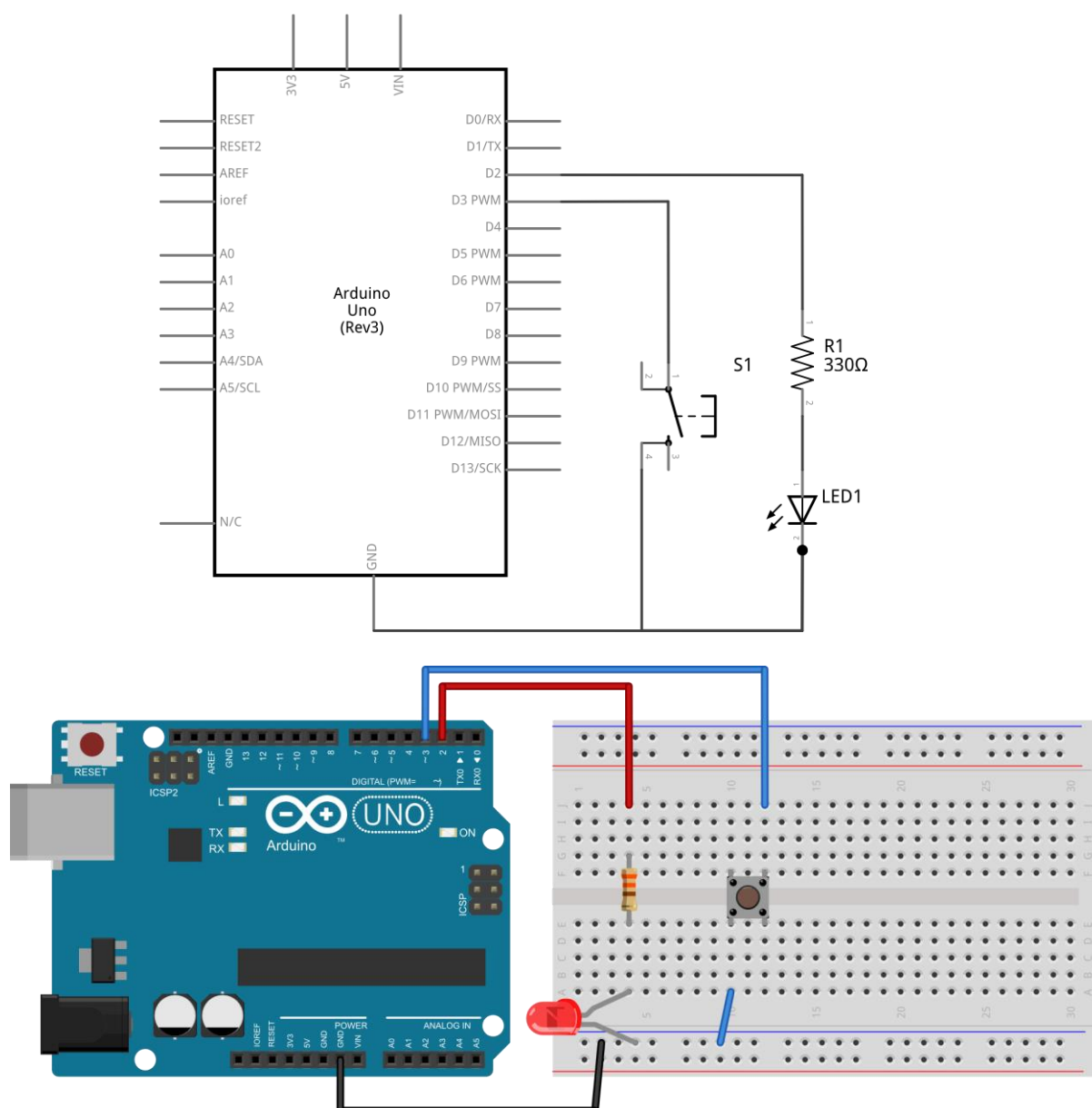
Otpušteno tipkalo – očitavamo stanje HIGH

Osnovni rad tipkala

U ovome ćemo zadatku prikazati osnovni način kako očitati tipkalo i djelovati u odnosu na očitavanje koje dobijemo.

Zadatak: Na mikrokontroler spojite jedno tipkalo i jednu svjetleću diodu. Napišite program koji će diodu uključivati i isključivati prema stanju tipkala. Ako je tipkalo pritisnuto dioda je uključena, a ako je tipkalo otpušteno dioda je isključena.

Slijedi prikaz spajanja:



Programski kôd rješenja:

```
int led = 2; //definiraj led = 2
int tipkalo = 3; //definiraj tipkalo = 3
int stanjeTipkala; //definiraj varijablu stanjeTipkala

void setup() {
    pinMode(led, OUTPUT); //postavi izvod led kao izlazni
    pinMode(tipkalo, INPUT_PULLUP); //postavi izvod tipkalo kao ulazni
    digitalWrite(led, LOW); //isključi led diodu -početno stanje
}

void loop() {
    stanjeTipkala=digitalRead(tipkalo); //očitaj stanje tipkala
    if(stanjeTipkala==HIGH){ //ako je tipkalo otpušteno
        digitalWrite(led, LOW); //isključi LED diodu
    } else { //inače (ako je tipkalo pritisnuto)
        digitalWrite(led, HIGH); //uključi LED diodu
    }
}
```

U prvoj vježbi govorili smo da izvodi mikrokontrolera mogu raditi na dva načina rada – ulazni i izlazni. U prethodnoj smo vježbi radili samo sa svjetlećim diodama i njima smo željeli upravljati pomoću mikrokontrolera pa smo uvijek koristili naredbu `pinMode(naziv izvoda, OUTPUT)`. Kod tipkala je situacija drugačija, s njih želimo očitavati stanje pa ćemo koristiti naredbu `pinMode(naziv izvoda, INPUT_PULLUP)`. Ovaj dodatak “_PULLUP” govori nam da koristimo unutarnji *pull-up* otpornik u mikrokontroleru kako smo pojasnili prije ovog zadatka.

Kako bismo očitali u kojem je stanju tipkalo u prvom smo dijelu programa napravili jednu varijablu koju smo nazvali *stanjeTipkala*. Unutar loop dijela programa u nju ćemo pomoću naredbe `digitalRead(naziv izvoda)` očitavati stanje tipkala. Funkcija za očitavanje tipkala vratit će nam jednu od dvije moguće vrijednosti – LOW ili HIGH. Kako bi provjerili u kojem se stanju nalazi tipkalo, tj. je li pritisnuto ili otpušteno iskoristit ćemo naredbu `if`. Ona ima sljedeću strukturu:

```
if (uvjet) { //ako je
    naredbe koje se izvršavaju ako je uvjet zadovoljen
} else { //inače
    naredbe koje se izvršavaju ako uvjet nije zadovoljen
}
```

U ovom zadatku željeli smo postići da svjetleća dioda svijetli dok je tipkalo pritisnuto. Stoga je postavljen uvjet koji uspoređuje je li vrijednost varijable jednaka HIGH i ako jest onda isključuje diodu, a ako nije (onda je LOW) i uključuje diodu. Vrlo je bitno primijetiti da je za usporedbu jednakosti vrijednosti neke varijable i vrijednosti LOW ili HIGH ili bilo koje druge vrijednosti bitno koristiti operator `==` koji se naziva operator usporedbe, a ne operator `=` jer je on operator pridruživanja.

Zadatak: Na mikrokontroler spojite dva tipkala (na dva zasebna izvoda) i jednu svjetleću diodu. Napišite program koji će omogućiti da prvim tipkalom uključujemo svjetleću diodu a drugim tipkalom isključujemo svjetleću diodu.

Programski kôd rješenja:

```
int led = 2; //definiraj led = 2
int tipkalo1 = 3; //definiraj tipkalo1 = 3
int tipkalo2 = 4; //definiraj tipkalo2 = 4

void setup() {
    pinMode(led, OUTPUT); //postavi izvod led kao izlazni
    pinMode(tipkalo1, INPUT_PULLUP); //postavi izvod tipkalo1 kao ulaz
    pinMode(tipkalo2, INPUT_PULLUP); //postavi izvod tipkalo2 kao ulaz
    digitalWrite(led, LOW); //isključi led diodu
}

void loop() {
    if(digitalRead(tipkalo1)==LOW){ //ako je tipkalo1 pritisnuto
        digitalWrite(led, HIGH); //uključi LED diodu
    }
    if(digitalRead(tipkalo2)==LOW){ //ako je tipkalo2 pritisnuto
        digitalWrite(led, LOW); //isključi LED diodu
    }
}
```

Očitavanje svakog tipkala izvedeno je u zasebnoj if naredbi. Kada je pritisnuto prvo tipkalo program će uključiti svjetleću diodu, a kada je pritisnuto drugo tipkalo, program će ju isključiti.

Pogledajte što se događa kada stisnemo oba tipkala zajedno. Što zaključujete?

Zadatak: Na mikrokontroler spojite dva tipkala i jednu svjetleću diodu. Napišite program tako da svjetleća dioda bude uključena dok god je pritisnuto bilo koje od dva tipkala. Kada su oba tipkala otpuštena LED dioda ne svijetli.

Programski kôd rješenja:

Unutar uvjeta if naredbe upisali smo dva uvjeta između kojih smo postavili operator `||`. Taj operator označava logičku funkciju ILI. Uvjet if petlje će biti zadovoljen onda kada su ili jedan ili drugi ili oba uvjeta zadovoljeni. Konkretno, u ovome primjeru, svjetleća dioda će se uključiti kada je pritisnuto bilo koje od tipkala.

Zadatak: Na mikrokontroler spojite dva tipkala i jednu svjetleću diodu. Napišite program koji će uključiti diodu kada su oba tipkala pritisnuta. U svim ostalim slučajevima LED dioda je isključena.

Programski kôd rješenja:

Unutar uvjeta if naredbe upisali smo dva uvjeta između kojih smo postavili operator `&&`. Taj operator označava logičku funkciju I. Uvjet if naredbe će biti zadovoljen samo onda kada su i jedan i drugi uvjet zadovoljeni. Konkretno, u ovome primjeru, svjetleća dioda će se uključiti kada su oba tipkala pritisnuta.

Zadatak: Na mikrokontroler spojite jedno tipkalo i jednu svjetleću diodu. Napišite program koji će svakim pritiskom tipkala invertirati stanje svjetleće diode – ako je dioda isključena, a korisnik stisne tipkalo, dioda će se uključiti i obrnuto.

Programski kôd rješenja:

Što primjećujete kada pritisnete tipkalo? Dogodi li se ponekad da pritisnete tipkalo a svjetleća dioda ne promijeni stanje?

To se događa zbog mehaničkog titranja tipkala. Naime, kada pritisnemo tipkalo ono zbog svojih mehaničkih svojstava nekoliko puta zatvori i otvori kontakt u vrlo kratkom vremenu. Zbog činjenice da se kôd u mikrokontroleru vrlo brzo izvršava, mikrokontroler ovo može detektirati i zamijeniti stanje svjetleće diode nekoliko puta, iako je korisnik tipkalo pritisnuo samo jednom. Postoji nekoliko rješenja ovoga problema, ali najjednostavnije rješenje je dodati kratku pauzu nakon očitavanja tipkala pomoću funkcije delay, trajanja oko 50 milisekundi.

Ovo rješenje nije uvijek najbolje jer se usporava izvođenje programa, ali za početak rada s mikrokontrolerima bit će zadovoljavajuće. Bolja rješenja nadograditi ćemo u nekoj od sljedećih vježbi. Slijedi loop dio gornjeg programskog kôda sa dodanom delay naredbom.

```
void loop() {  
  if(digitalRead(tipkalo)==LOW){ //ako je pritisnuto tipkalo  
    if(stanje==LOW){ //ako je stanje LOW  
      stanje=HIGH; //postavi stanje u HIGH  
    } else { //inače  
      stanje=LOW; //postavi stanje u HIGH  
    }  
    delay(50); //čekaj 50 ms  
    digitalWrite(led, stanje); //postavi izvod led u "stanje"  
  }  
}
```