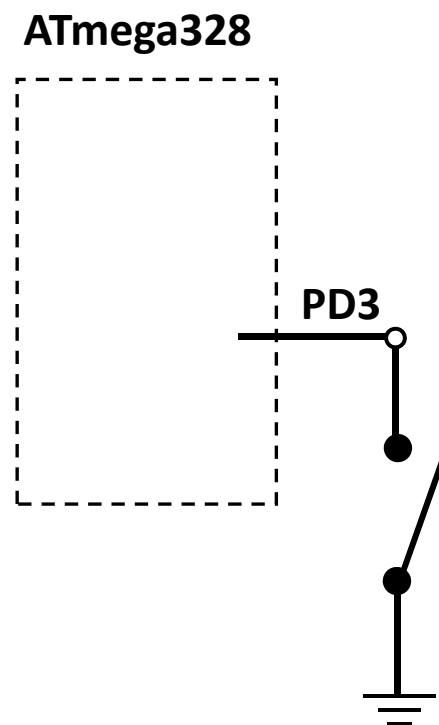


Ulazni portovi

Digitalni

Ulazni digitalni pin – Primjer 1

- ‘Očitavanje ulaznog pina’
 - Napisati ćemo nekoliko C linija koda za Arduino u cilju definisanja načina djelovanja kada je pojas vozača u autu vezan (prekidač zatvoren).
 - Ako je pojas vezan, omogućeno je uključenje auta kroz poziv funkcije `start_enable()`.
 - Ako pojas nije vezan omogućeno je uključenje auta kroz poziv funkcije `start_disable()`
 - Napisaćemo najprije psudokod!



Ulazni digitalni pin – Primjer 1

- ‘Očitavanja pina’

- Pseudokod:

- Postaviti PD3 kao ulazni

- Uključiti PD3 pull-up otpornik

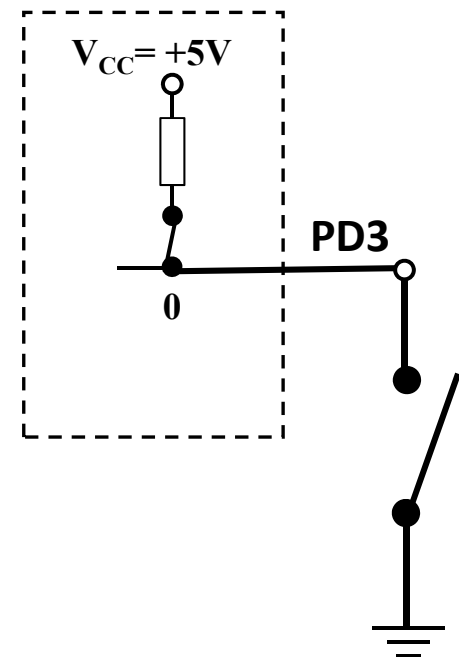
- Očitati napon sa Arduino pin 3 (PIN_D3)

- IF PIN_D3 napon je LOW (vezan), THEN
pozovi funkciju start_enable()

- ELSE

- pozovi start_disable()

ATmega328



Ulazni digitalni pin – Primjer 1

- ‘Očitavanja pina’

- Pseudokod:

- Postaviti PD3 kao ulazni

- Uključiti PD3 pull-up otpornik

- Očitati napon sa Arduino pin 3 (PIN_D3)

- IF PIN_D3 napon je LOW (vezan), THEN

- pozovi funkciju start_enable()

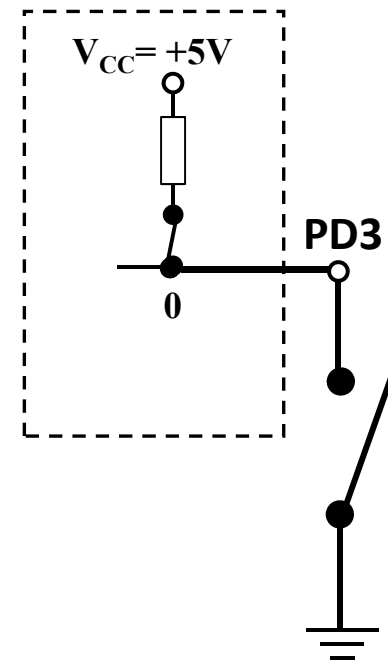
- ELSE

- pozovi start_disable()

Fragment. Nije cijeli program.

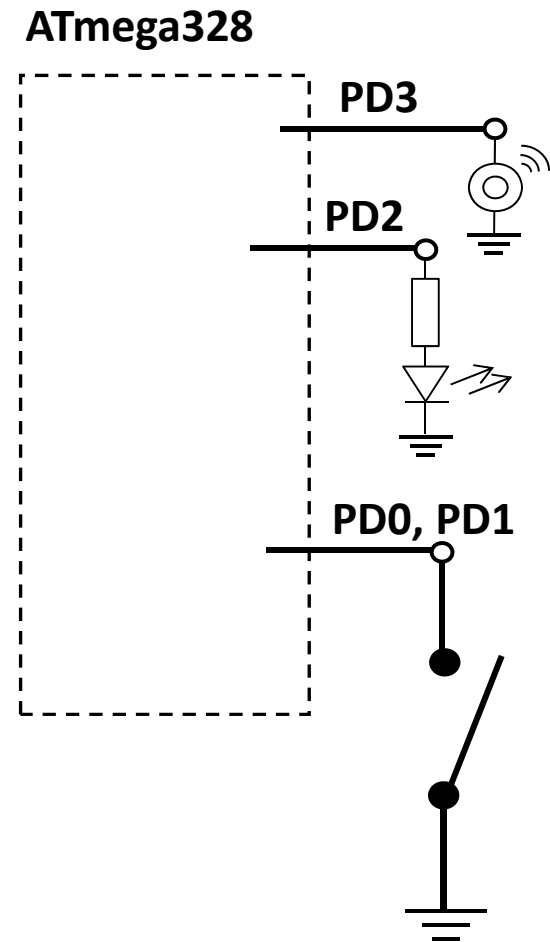
```
#define PIN_SWITCH 3
#define LATCHED LOW
pinMode(PIN_SWITCH, INPUT_PULLUP);
belt_state = digitalRead(PIN_SWITCH);
if (belt_state == LATCHED)
{ ig_enable(); }
else
{ ig_disabled(); }
```

ATmega328



Ulazni digitalni pin – Primjer 2

- Čitanje sa pina i upisivanje na pin
 - Napisaćemo nekoliko linija C koda za Arduino, s ciljem uključenja LED (PD2) i zvučnog signala (PD3) ako je ključ u bravi (PD0 zatvoren), ali pojas vozača nije vezan (PD1 otvoren)
 - Najprije pseudokod



Ulazni digitalni pin – Primjer 2

- Pseudokod:

Postavljanje toka podataka za pinove

Postaviti PD0 i PD1 kao ulaze

Uključiti pull-up otpornike za PD0 i PD1

Postaviti PD2 i PD3 kao izlaze

Beskonačna petlja

IF je ključ u bravi THEN

IF ako je pojas vezan, THEN

Isključi zvučni signal

Isključi LED

ELSE

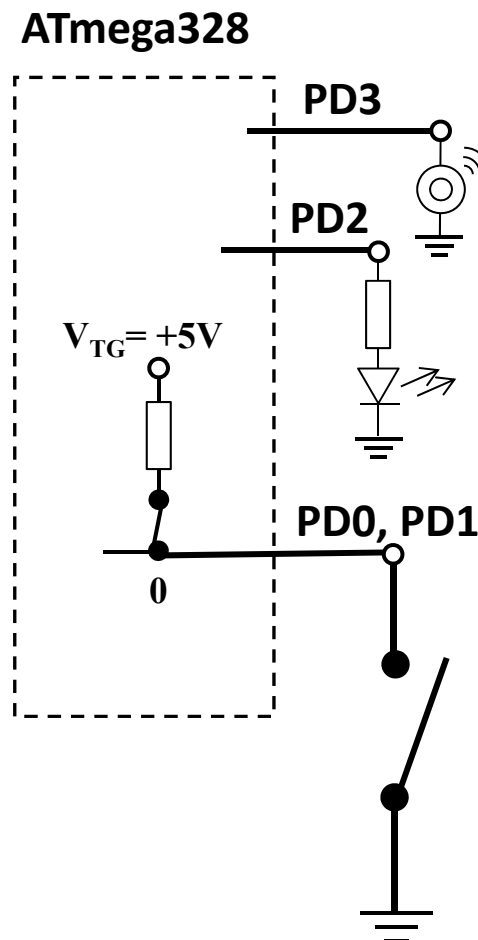
Uključi LED

Uključi zvučni signal

ELSE

Isključi zvučni signal

Isključi LED

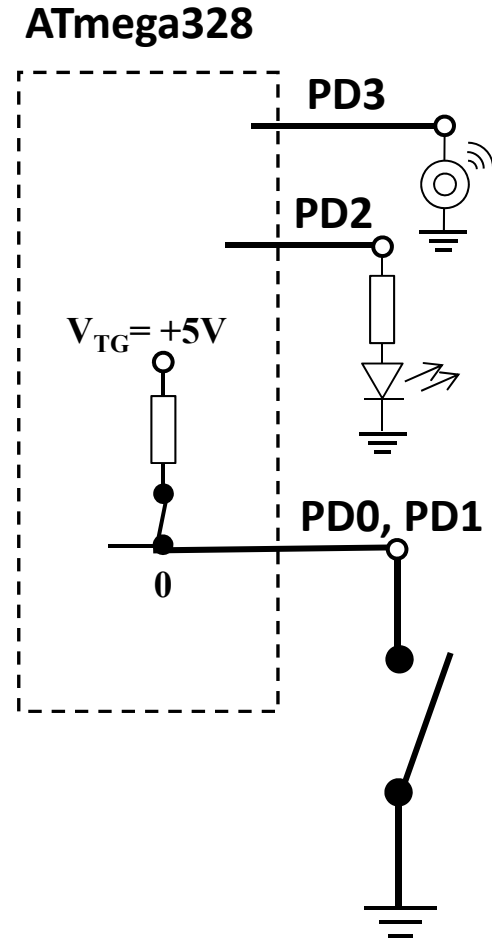


Ulazni digitalni pin – Primjer 2 (Arduino kod)

```
#define PIN_IGNITION 0
#define PIN_SEATBELT 1
#define PIN_LED 2
#define PIN_BUZZER 3
#define SEATBELT_LATCHED LOW
#define KEY_IN_IGNITION LOW
#define LED_ON HIGH
#define LED_OFF LOW
#define BUZZER_ON HIGH
#define BUZZER_OFF LOW

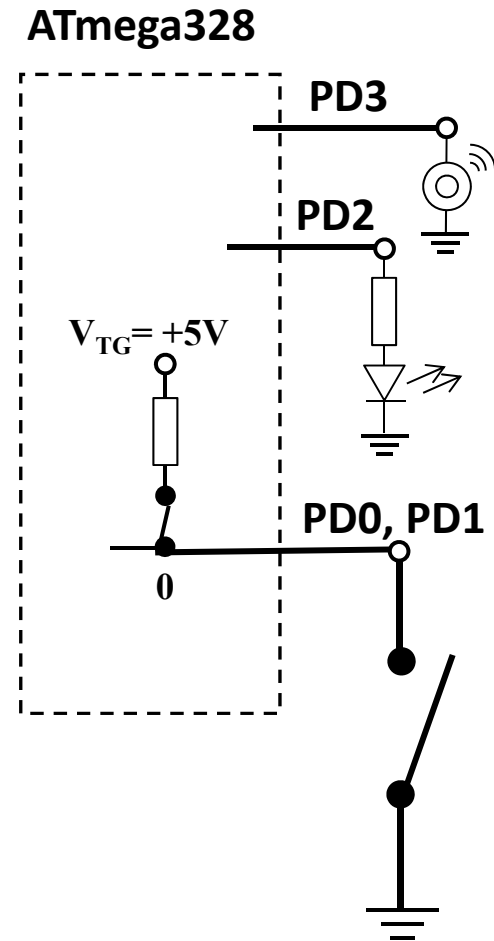
void setup()
{
  pinMode(PIN_IGNITION, INPUT_PULLUP); // key switch
  pinMode(PIN_SEATBELT, INPUT_PULLUP); // belt latch switch
  pinMode(PIN_LED, OUTPUT); // lamp
  pinMode(PIN_BUZZER, OUTPUT); // buzzer
}
```

/ see next page for code */*



Ulazni digitalni pin – Primjer 2 (Arduino kod)

```
/* see previous page for code before loop() */  
void loop()  
{  
  int key_state = digitalRead(PIN_IGNITION);  
  int belt_state = digitalRead(PIN_SEATBELT);  
  if (key_state == KEY_IN_IGNITION)  
  {  
    if (belt_state == SEATBELT_LATCHED)  
    {  
      digitalWrite(PIN_BUZZER, BUZZER_OFF);  
      digitalWrite(PIN_LED, LED_OFF);  
    }  
    else // key is in ignition, but seatbelt NOT latched  
    {  
      digitalWrite(PIN_BUZZER, BUZZER_ON);  
      digitalWrite(PIN_LED, LED_ON);  
    }  
  }  
  else // key is NOT in ignition  
  {  
    digitalWrite(PIN_BUZZER, BUZZER_OFF);  
    digitalWrite(PIN_LED, LED_OFF);  
  }  
}
```



Ulazni digitalni pin – Primjer 2 (Alternativni kod)

/* NOTE: #defines use predefined PORT pin numbers for ATmega328 */

#define PIN_IGNITION PD0

#define PIN_SEATBELT PD1

#define PIN_LED PD2

#define PIN_BUZZER PD3

#define SEATBELT_LATCHED LOW

#define KEY_IN_IGNITION LOW

#define LED_ON HIGH

#define LED_OFF LOW

#define BUZZER_ON HIGH

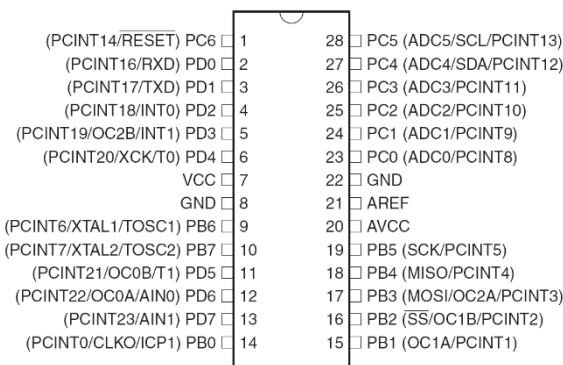
#define BUZZER_OFF LOW

#define _BIT_MASK(bit) (1 << (bit)) // same as _BV(bit)

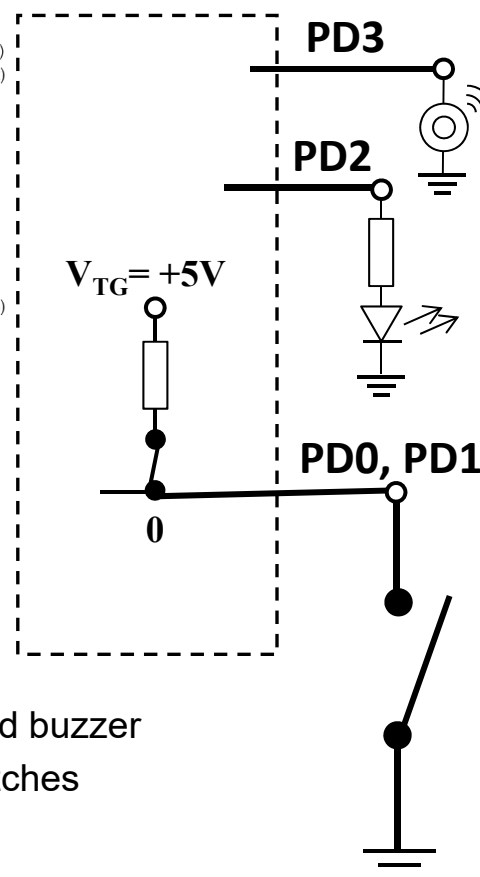
void setup()

```
{
  PORTD = 0; // all PORTD pullups off
  DDRD = _BIT_MASK(PIN_LED) | _BIT_MASK(PIN_BUZZER); // LED and buzzer
  PORTD |= _BV(PIN_IGNITION) | _BV(PIN_SEATBELT); // pullups for switches
}
```

/* See next page for loop() code */

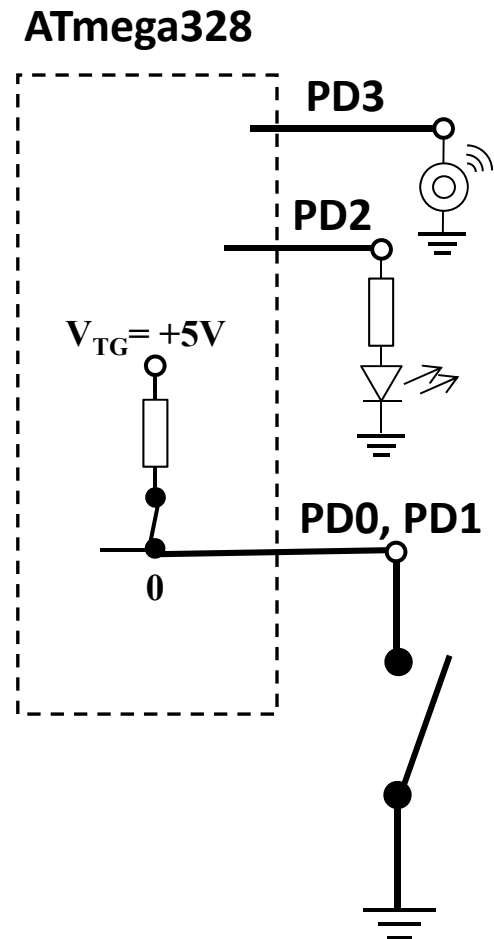


ATmega328



Ulazni digitalni pin – Primjer 2 (Alternativni kod)

```
/* see previous page for setup() code */  
void loop()  
{  
  uint8_t current_PORTD_state, key_state, belt_state;  
  current_PORTD_state = PIND; // snapshot of PORTD pins  
  key_state = current_PORTD_state & _BV(PIN_IGNITION);  
  belt_state = current_PORTD_state & _BV(PIN_SEATBELT);  
  if (key_state == KEY_IN_IGNITION)  
  {  
    if (belt_state == SEATBELT_LATCHED)  
    {  
      PORTD &= ~(_BV(PIN_LED) | _BV(PIN_BUZZER) );  
    }  
    else  
    {  
      PORTD |= ( _BV(PIN_LED) | _BV(PIN_BUZZER) );  
    }  
  }  
  else  
  {  
    PORTD &= ~(_BV(PIN_LED) | _BV(PIN_BUZZER) );  
  }  
}
```



“Konkurencija” Arduinu

- Alternative Arduino platformi
 - Pinguino – PIC kontroler
 - MSP430 – Texas Instruments;
 - Drugi: korisnički, Teensy, itd.
- Netduino
- Računari
 - Raspberry Pi
 - BeagleBones – TI; ima računar i kontroler

Netduino

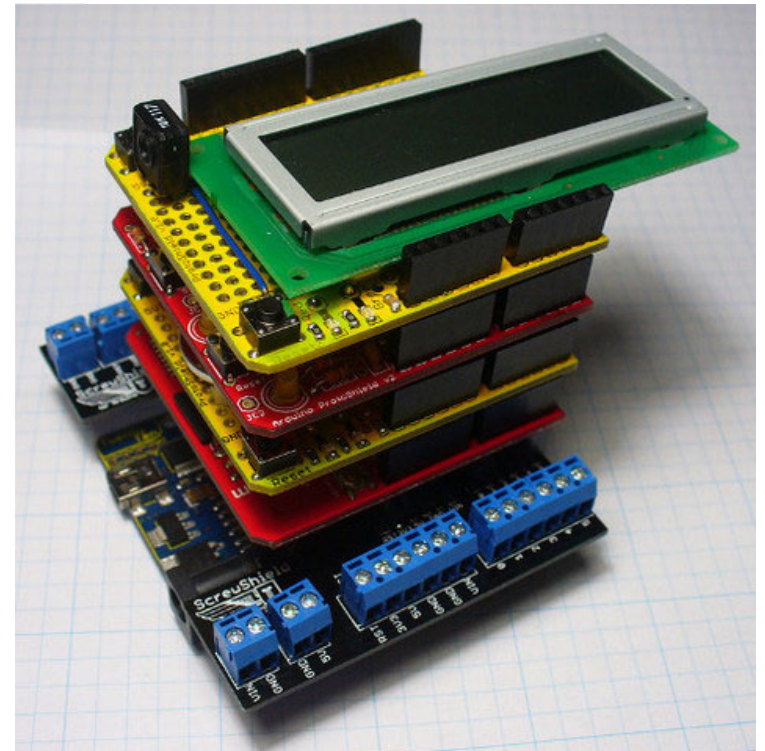
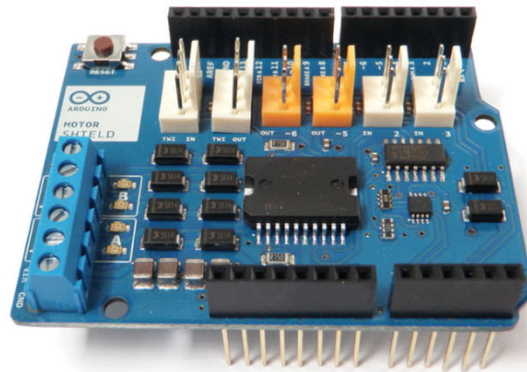
- Mikrokontroler i razvojno okruženje kreirano od strane Microsoft za rad sa .NET Micro Framework.
- DALEKO moćnije razvojno okruženje.
- Razlike
 - Pinovi na Netduino su 3.3V (ne 5V).
 - Netduino ima mnogo brži procesor
 - 60K RAM-a (naspram Uno-vih 2K).
- U velikoj mjeri kompatibilan sa Arduino-om, ali nije potpuno izmjenjiv.

Raspberry Pi

- Pojednostavljeni računar, ne kontroler
- Koristi Debian Linux
 - Arch Linux ARM, Fedora, FreeBSD, Slackware...
- Programiran sa Python-om
 - BBC BASIC, C, Perl
- Kako je u pitanju računar (ime operativni sistem), ima drugačiju ulogu u projektima.
- Hijerarhija: računar upravlja kontrolerima, kontroleri upravljaju hardverom.

Šildovi (Dodaci)

- Šildovi su ploče koje se dodaju na Arduino ploču.
- Oni proširuju mogućnosti Arduina.
- Primjeri:
 - Ethernet
 - GPS
 - Motor
 - Prototip
- shieldlist.org

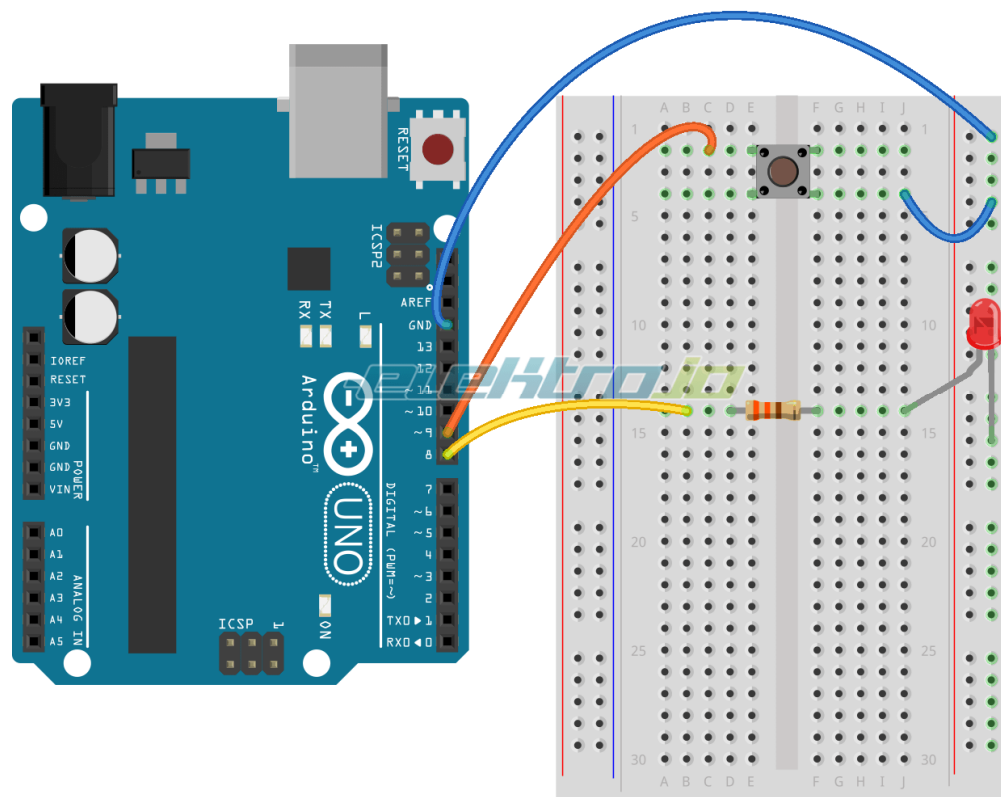


Zaključak

- Arduino platforma predstavlja jeftin način da se uđe u svijet robotike.
- Arduino ima:
 - Brojne korisnike
 - Bogatu online biblioteku kodova i projekata

Zadaci za vježbu 2

1. Upotrijebiti taster za uključenje i isključenje LED. Dva pritiska uključuju LED, treći pritisak isključuje i tako u krug. **(2 poena)**



Made with  Fritzing.org

2. Trčeće svjetlo sa 4 LED. Smjer se određuje prekidačem. Prekidač otvoren - jedan smjer, prekidač zatvoren - drugi smjer. Zaustavlja se pritiskom na taster. **(3 poena)**



3. Brojiti koliko je puta, od startovanja program, pritisnut taster. Informaciju o broju pritiskanja tatera ispisati na jednocifarskom sedmosegmentnom displeju. Kada broj pritiskanja premaši 9, ispisati 0 i tako u krug. **(4 poena)**



4. Nakon pritiska tastera LED trepne tri puta, nakon čega ostaje isključena do otpuštanja, kada trepne dva puta i ostaje isključena do ponovnog pritiskanja tastera. (4 poena)



5. Nivo tečnosti u bazenu. Kao gornji i donji senzor nivoa upotrijebiti obične kratkospojnike, a kao bazen čašu i nešto vode u njoj. Informaciju o nivou tečnosti signalizirati na jednocifarkom sedmosegmentnom displeju, LED diodama i zvučno, na sljedeći način: Visok nivo - slovo 'H' i uključena crvena LED, nizak nivo - slovo 'L' i uključena žuta LED, OK nivo - slovo 'O' i uključena zelena LED, neispravnost - slovo 'E', isključene sve LED i isprekidani zvučni signal. **(6 poena)**

