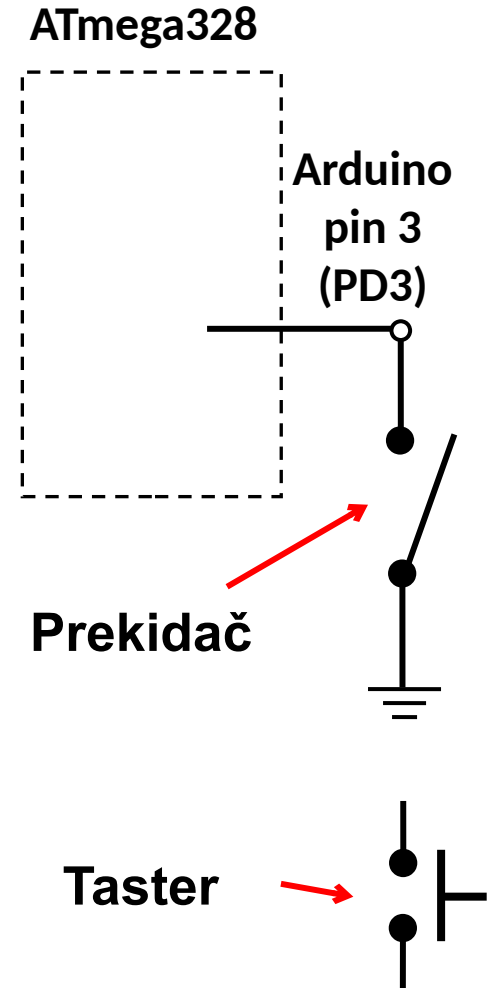


ULAZNI DIGITALNI PORTOVI



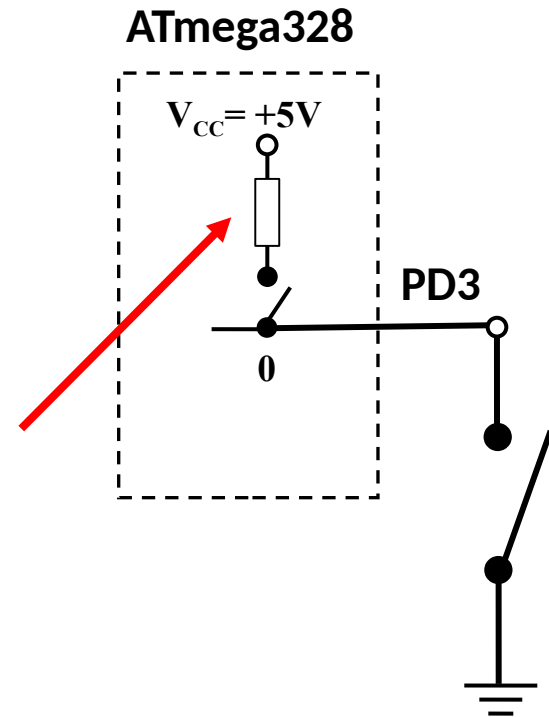
Pin kao ulazni + Pull-up otpornik

- Prekidač kao senzor
 - Pr. Senzor pojasa za sjedište u autu
 - Detekcija **stanja prekidača**
 - Koji tok podataka treba biti za Arduino pin 3 (PD3)?
 - `pinMode(3, INPUT);`
 - Koji će biti napon na PD3 kada je prekidač zatvoren?
 - Koji će biti napon na PD3 kada je prekidač otvoren?



Pin kao ulazni + Pull-up otpornik

- Prekidač kao senzor, nastavak.
 - Učinimo napon na pinu poznatim uključanjem pull-up otpornika za PD3
 - Neka je PD3 ulazni port:
 - `digitalWrite(3,HIGH);`
uključenje “pull-up” otpornika
 - `pinMode(3,INPUT_PULLUP);`
 - Koji će napon biti na PD3 kada je prekidač otvoren?
 - Koji će napon biti na PD3 kada je prekidač zatvoren?

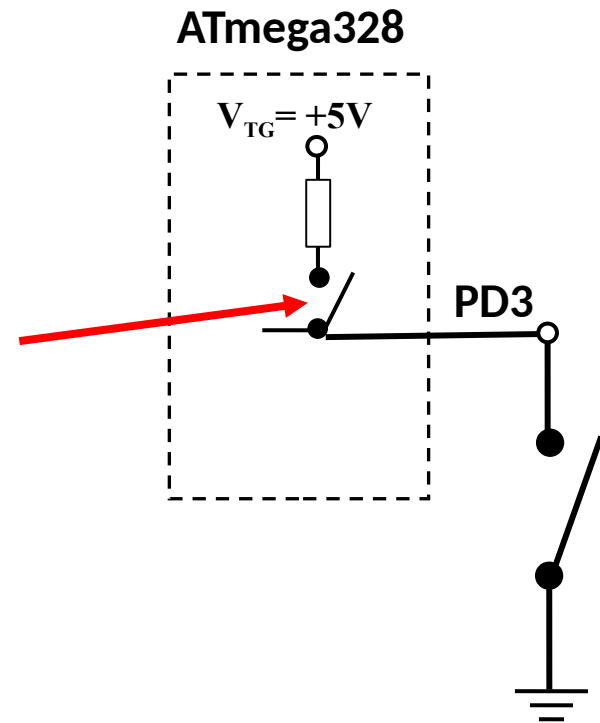


Pin kao ulazni + Pull-up otpornik

- Prekidač kao senzor, nastavak.
 - Za isključenje pull-up otpornika
 - Neka je PD0 ulazni port:

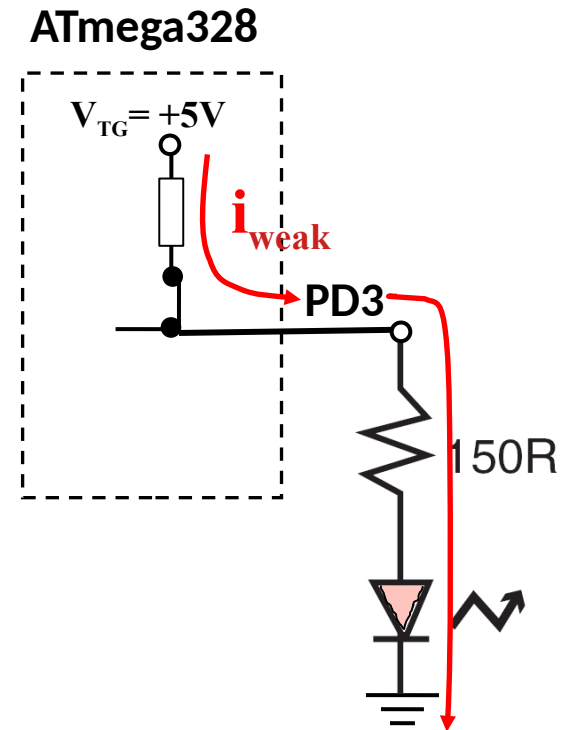
```
digitalWrite(3, LOW);
```

Isključuje "pull-up" otpornik



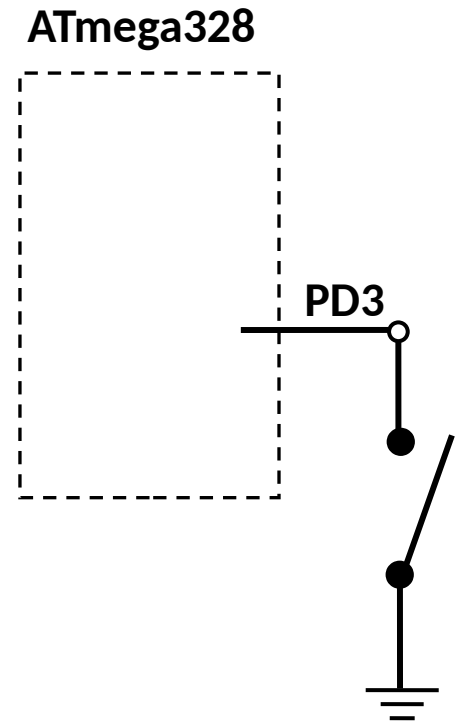
Pin kao ulazni + Pull-up otpornik

- Mogućnost 'slabog pogona' kada je pull-up otpornik uključen
 - Pin koji je postavljen kao ulazni sa uključenim pull-up otpornikom može dati malu struju.



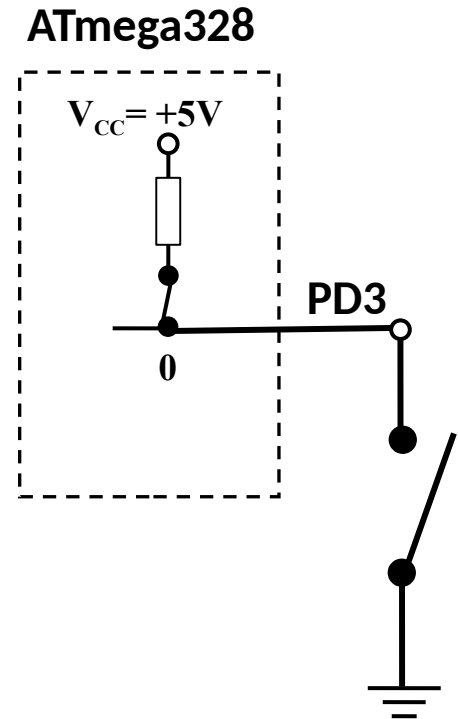
Ulazni digitalni pin – Primjer 1

- ‘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()



Primjer 2

- Postaviti Arduino pinove 2 i 3 (PD2 i PD3) kao ulazne, i uključiti pull-up otpornike

- Arduino pristup

```
pinMode(2, INPUT);  
pinMode(3, INPUT);  
digitalWrite(2, HIGH);  
digitalWrite(3, HIGH);
```

Ili ako je upotrijebljena me106.h:

```
pinMode(PIN_D2, INPUT);  
pinMode(PIN_D3, INPUT);  
digitalWrite(PIN_D2, HIGH);  
digitalWrite(PIN_D3, HIGH);
```

- Alternativni pristup

```
DDRD = 0; // all PORTD pins inputs  
PORTD = 0b00001100;  
ili  
PORTD = 0x0C;
```

Ili još bolje:

```
DDRD &= ~(1<<PD2 | 1<<PD3);  
PORTD |= (1<<PD2 | 1<<PD3);
```

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()

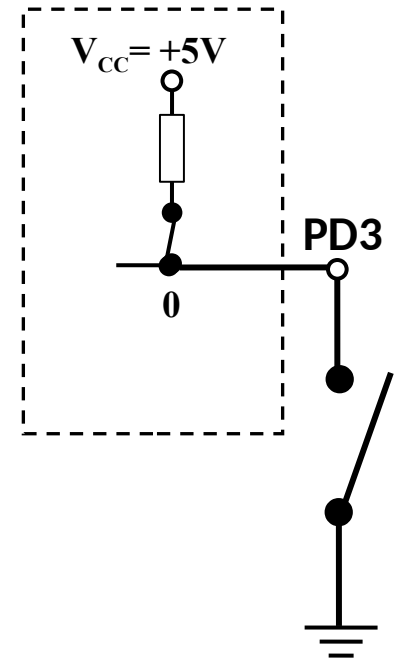
- ELSE

- pozovi start_disable()

Fragment. Nije cijeli skeč.

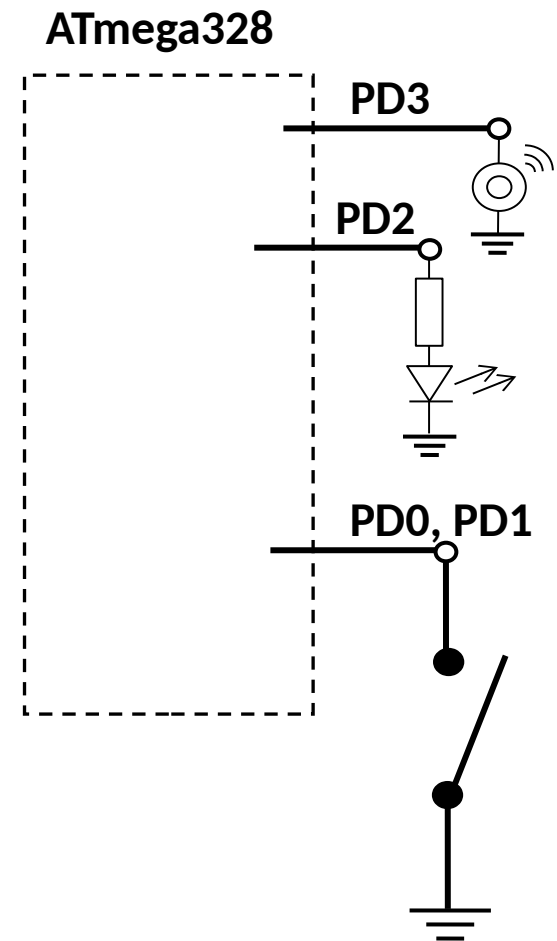
```
#define PIN_SWITCH 3
#define LATCHED LOW
pinMode(PIN_SWITCH, INPUT_PULLUP);
belt_state = digitalRead(PIN_SWITCH);
if (belt_state == LATCHED)
{ ig_enable(); }
else
{ ig_disabled(); }
```

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

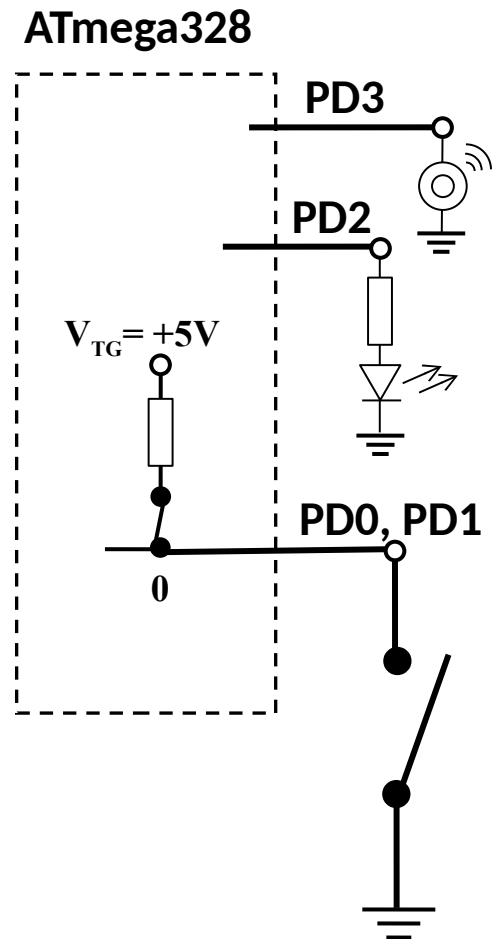
Uključi LED

Uključi zvučni signal

ELSE

Isključi zvučni signal

Isključi LED

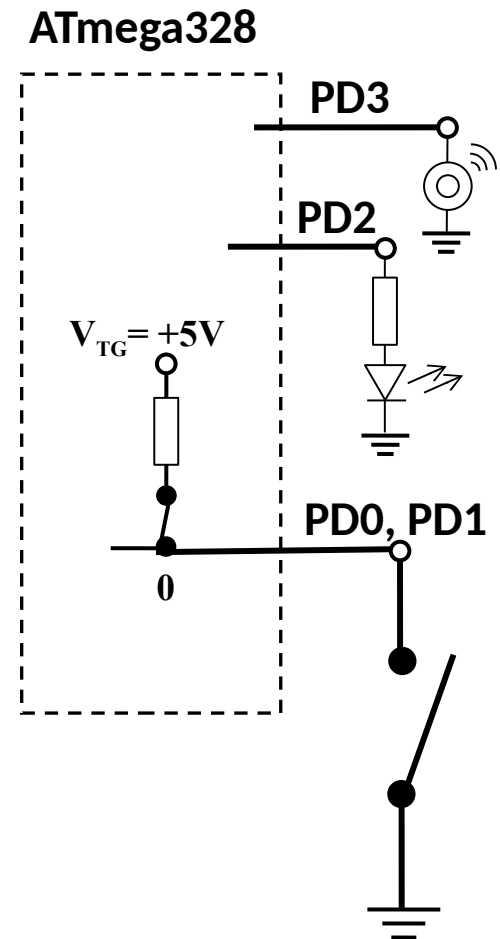


Ulazni digitalni pin – Primjer 2 (Arduino kod)

```
#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

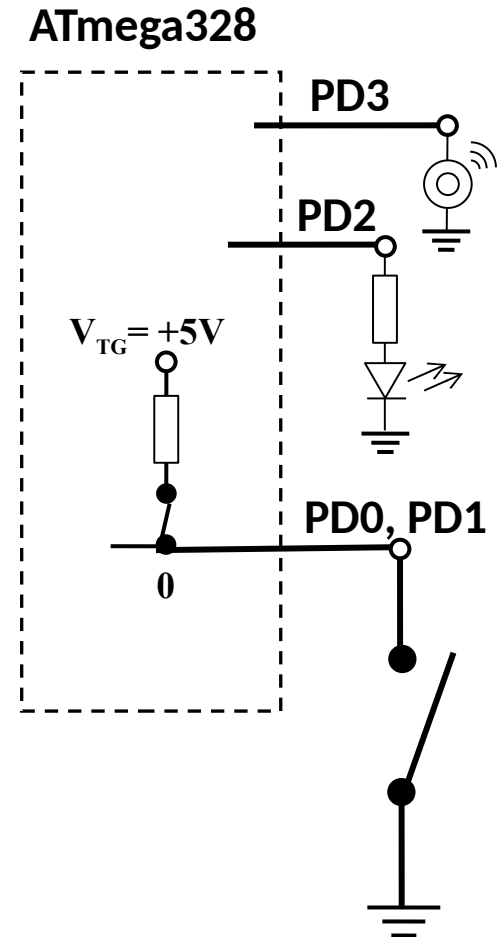
void setup()
{
  pinMode(PIN_IGNITION, INPUT_PULLUP); // key switch
  pinMode(PIN_SEATBELT, INPUT_PULLUP); // belt latch switch
  pinMode(PIN_LED, OUTPUT); // lamp
  pinMode(PIN_BUZZER, OUTPUT); // buzzer
}
```

/* 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()

{

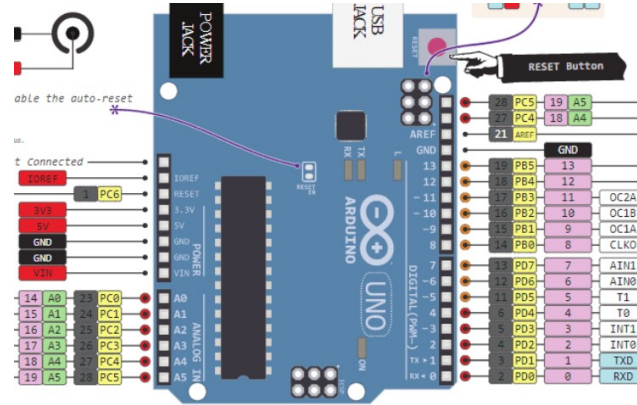
PORTD = 0; // all PORTD pullups off

DDRD = _BIT_MASK(PIN_LED) | _BIT_MASK(PIN_BUZZER); // LED and buzzer

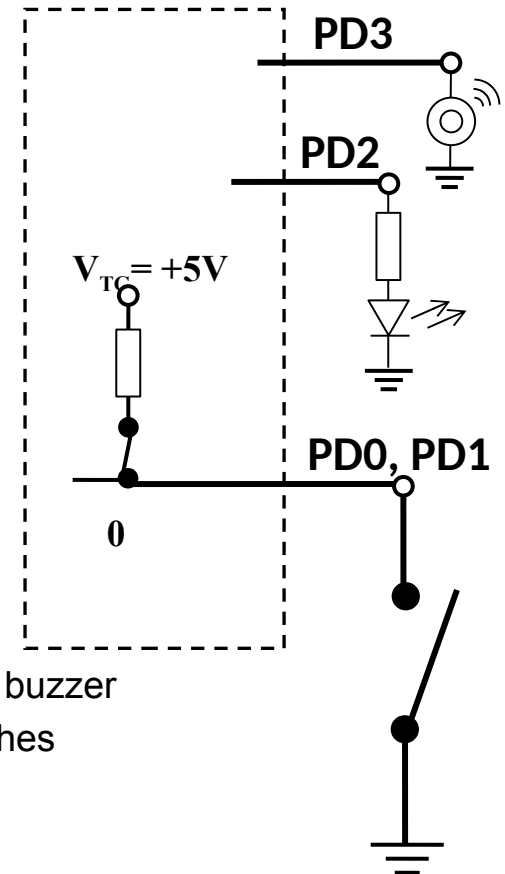
PORTD |= _BV(PIN_IGNITION) | _BV(PIN_SEATBELT); // pullups for switches

}

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

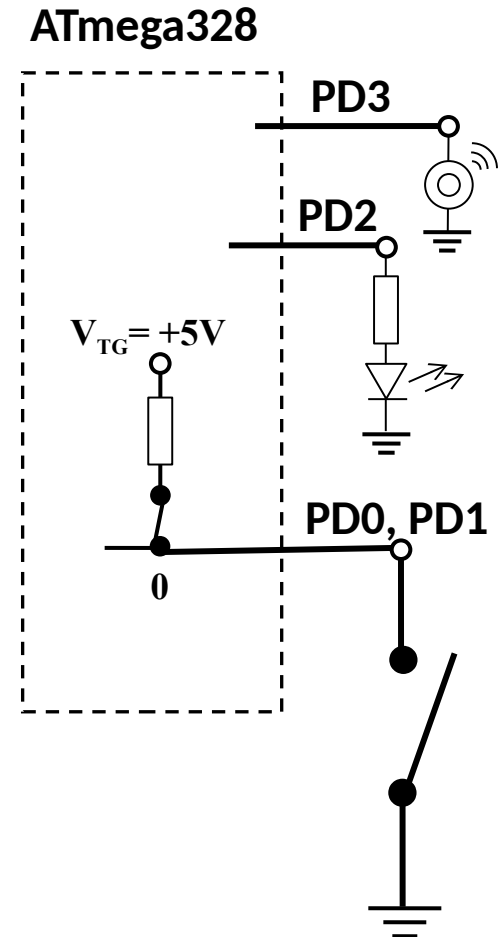


ATmega328



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));
    }
}
```



Serijska komunikacija

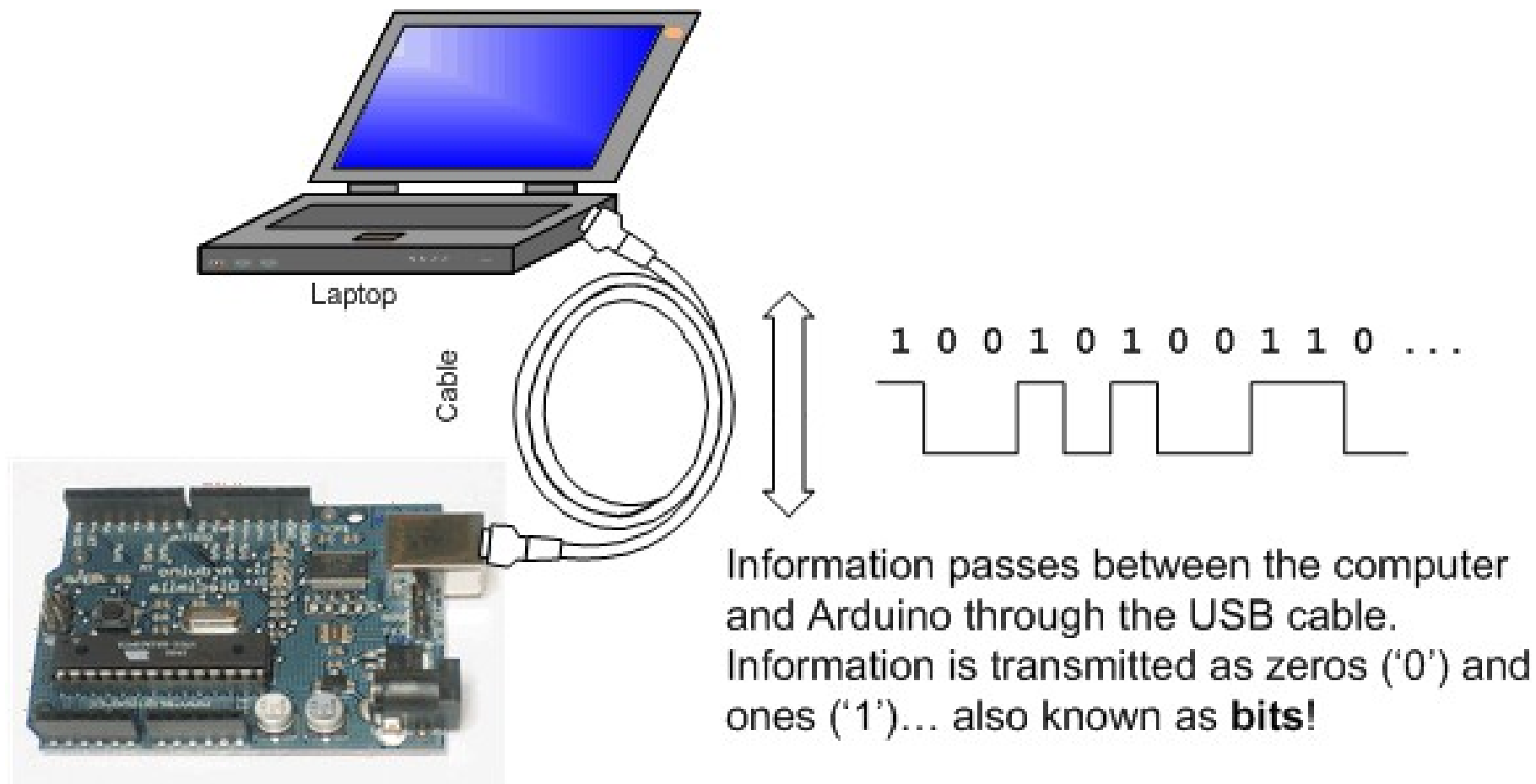
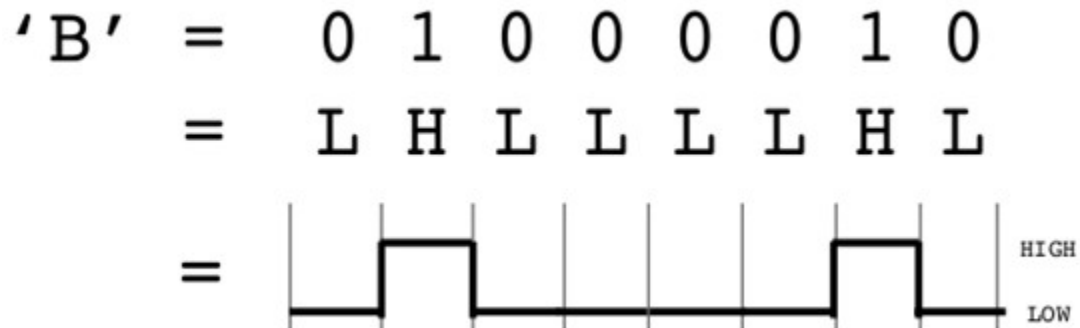


Image from <http://www.ladyada.net/learn/arduino/lesson4.html>

Serijska komunikacija

Serijska- jer su podaci razbijeni na bitove. Svaki bit se šalje jedan za drugim preko jedne žice

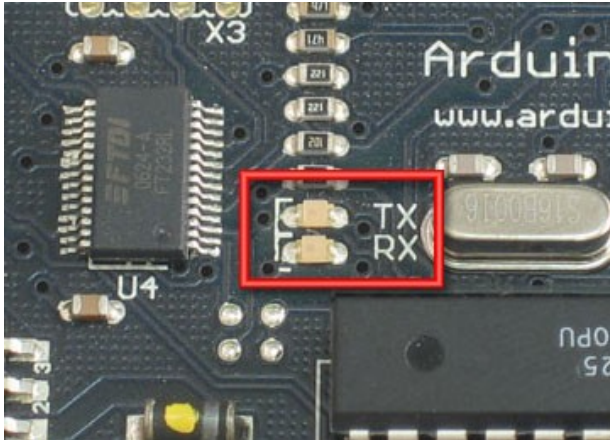
Primjer: ASCII karakter 'B' se šalje kao:



Mijenja se stanje na pinu baš kao kada se upravlja treperenjem LED.

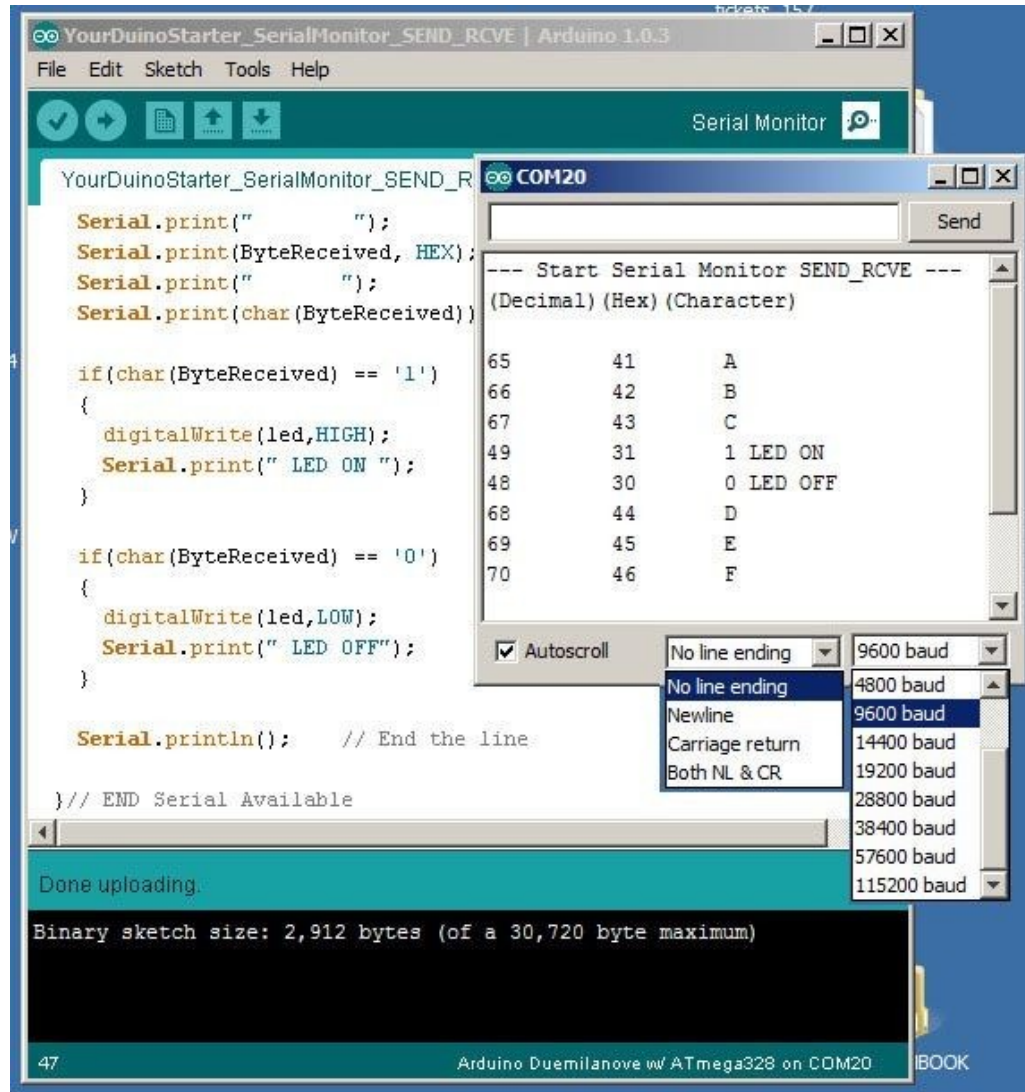
Jedna linija se koristi za slanje i jedna za prijem podataka.

Serijska komunikacija



- **Kompajliranje** prevodi program u binarne podatke (jedinice i nule)
- **Uploading (upisivanje)** šalje bitove kroz USB kabl do Arduina.
- Dvije LED diode blizu USB konektora trepere dok se podaci prenose
 - **RX** treperi kada Arduino prima podatke
 - **TX** treperi kada Arduino šalje

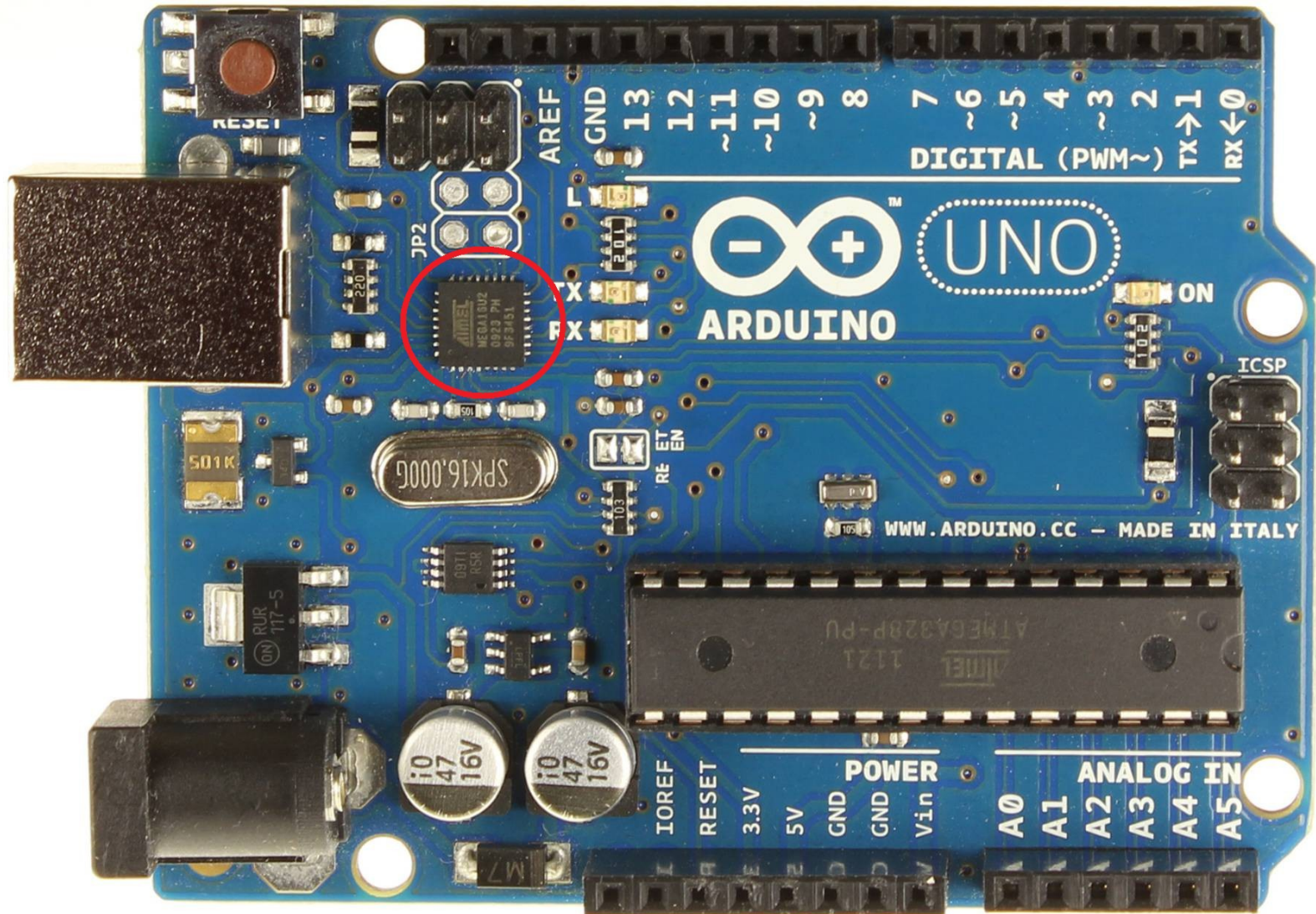
Serijski monitor



Osnovne komande

- `Serial.begin()`
 - pr., `Serial.begin(9600)`
- `Serial.print()` or `Serial.println()`,
`Serial.write()`

Serial-to-USB chip



Dva različita komunikaciona protokola

Serijski (TTL):

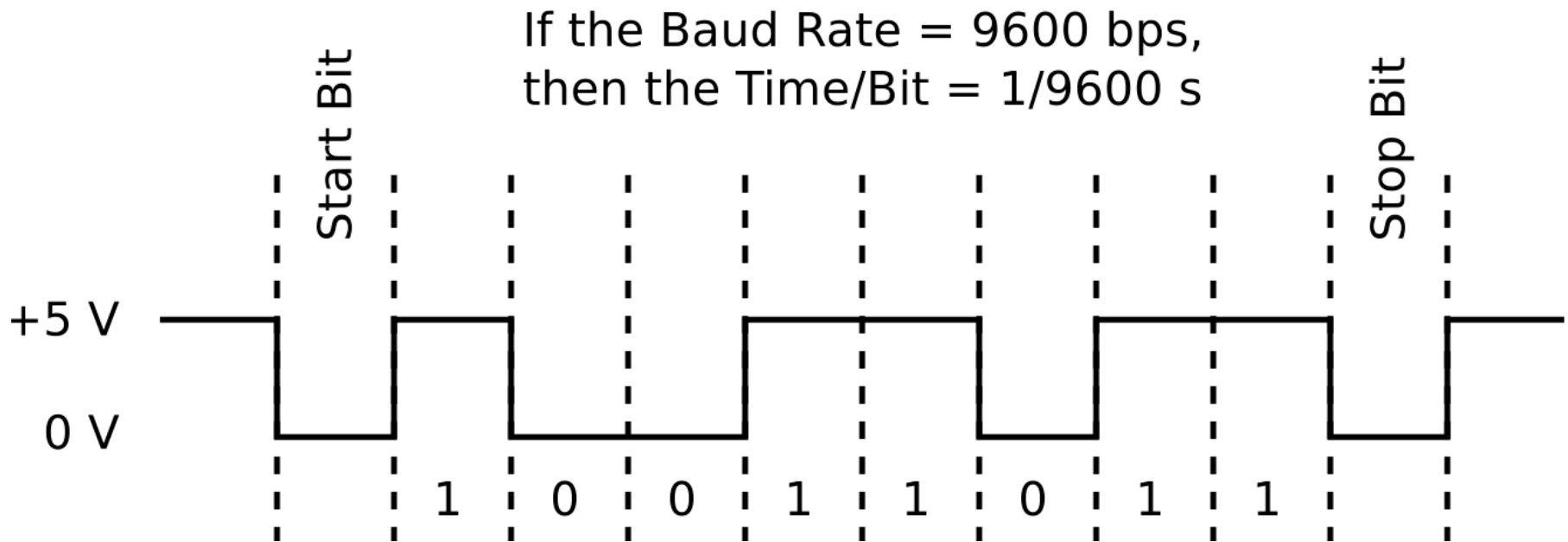
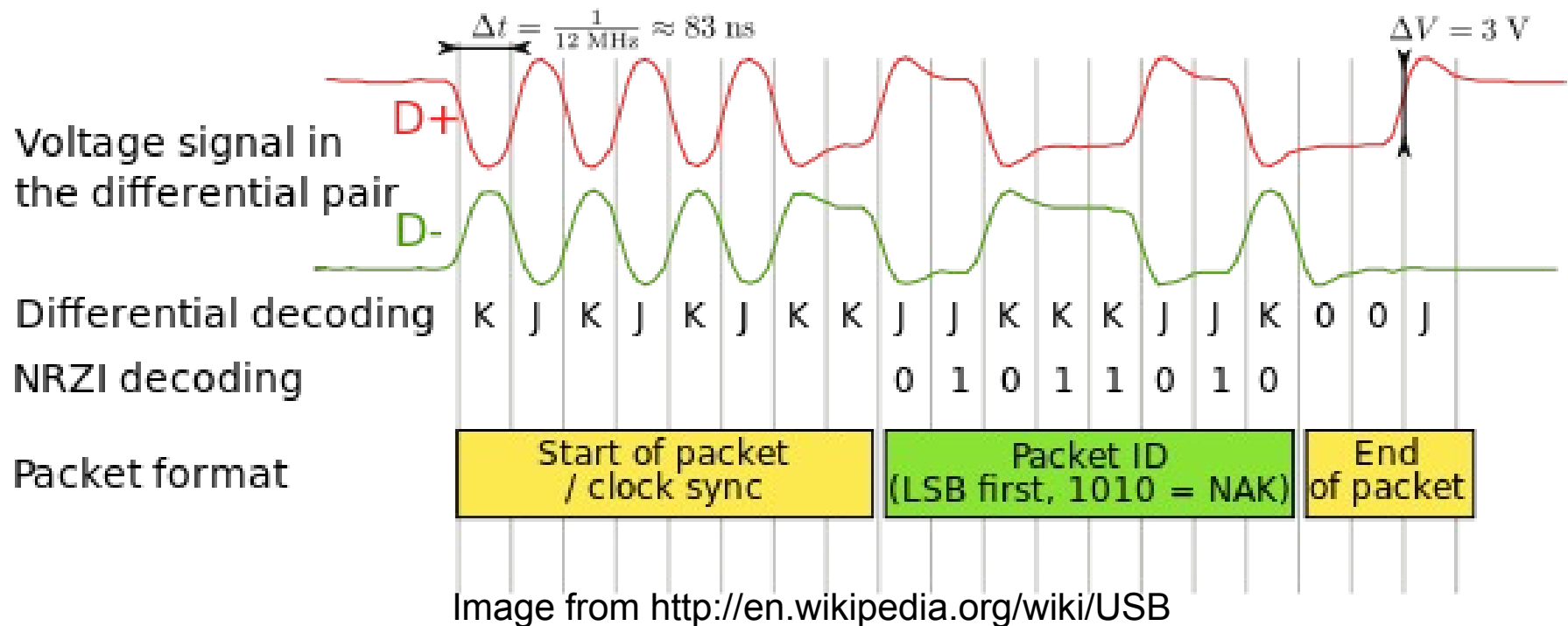


Image from <http://www.fiz-ix.com/2013/02/introduction-to-arduino-serial-communication/>

USB protokol



- Puno komplikovaniji

Brojanje koliko puta je pritisnut taster.



```
#define TASTER 6
#define LED 2
#define TST_CHECK_PERIOD 5

int brojPritisaka;
boolean Taster, pTaster;
unsigned long nextTstCheck;

void setup() {
    pinMode(TASTER, INPUT_PULLUP);
    pinMode(LED, OUTPUT);
    Serial.begin(115200);

    Taster = digitalRead(TASTER);
    pTaster = Taster;
}

void loop() {
    if (millis() - nextTstCheck > TST_CHECK_PERIOD)
    {
        Taster = digitalRead(TASTER);
        if (!Taster && pTaster) {
            brojPritisaka++;
            Serial.println(brojPritisaka);
        }
        pTaster=Taster;
        nextTstCheck=millis();
    }

    if (Taster) digitalWrite(LED, LOW);
    else digitalWrite(LED, HIGH);
}
```

Zadaci za vježbu

PRVI ZADATAK

Upotrijebiti taster za kontrolu načina rada LED.

Prvim neparnim pritiskom tastera i po njegovom otpuštanju, uključiti LED.

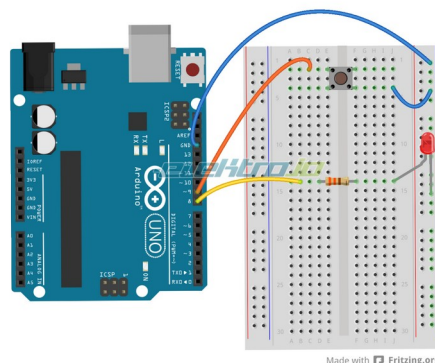
Drugim neparnim pritiskom tastera i po njegovom otpuštanju, inicirati treperenje LED, 300ms uključena, 2000ms isključena.

Trećim neparnim pritiskom tastera i po njegovom otpuštanju, inicirati treperenje LED, 300ms isključena, 2000ms uključena.

Parni pritisak tastera isključuje LED.

Obezbijediti da ovo radi u krug, odnosno da je za četvrti neparni pritisak tastera isto kao i za prvi, peti kao i za drugi, šesti kao i za treći, sedmi kao i za prvi, itd.

(2-1 poen)



Mogući način povezivanja.

DRUGI ZADATAK

Pomoću tastera i prekidača upravljati radom 4 LED na sljedeći način:

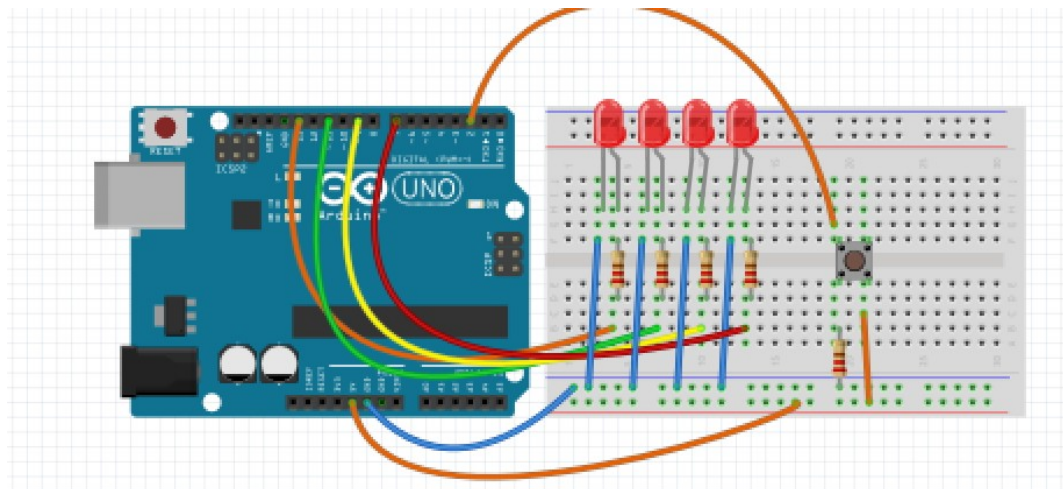
Dva pritiska taster unutar 1000ms iniciraju trčeće svjetlo. Promjena stanja svjetla svakih 500ms. Smjer trčećeg svjetla se određuje prekidačem. Prekidač otvoren - jedan smjer, prekidač zatvoren - drugi smjer.

Tri pritiska tastera unutar 1000ms iniciraju trećeću "tamu". Promjena stanja svjetla svakih 500ms. Smjer trčećeg svjetla se određuje prekidačem. Prekidač otvoren - jedan smjer, prekidač zatvoren - drugi smjer.

Jedan pristisak tastera unutar 1000ms zaustavlja protrčavanje svjetla ili tame i isključuje LED.

Način rada LED treba se mijanjati neposredno po ispunjenju uslova za promjenu.

(2-1 poen)



Mogući način povezivanja. Treba još dodati prekidač.

TREĆI ZADATAK

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 indicirati na četvorocifarskom sedmo-segmentnom displeju:

- OK nivo – druga cifra s lijeva slovo O, treća cifra slijeva slovo K, prva i četvrta cifra isključene.
- nizak nivo – na displeju ispisati slovo 'LOUU',
- visok nivo – na displeju ispisati 'HIGH',
- neispravnost – na displeju treba ispisati 'EEEE', koje treperi, tako što je 500ms na displeju, a 500ms cifre isključene.

(2-1 poen)

