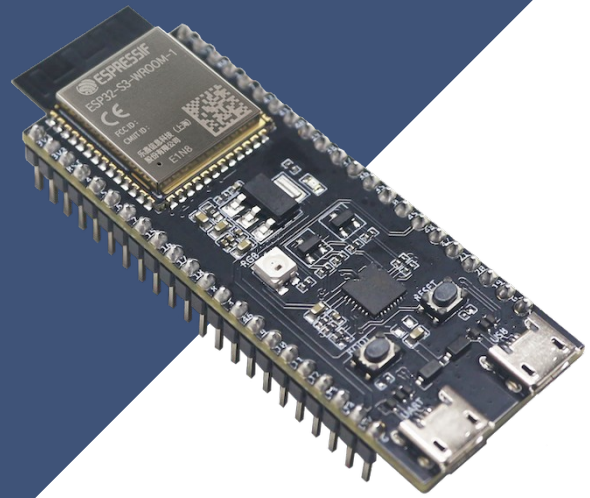
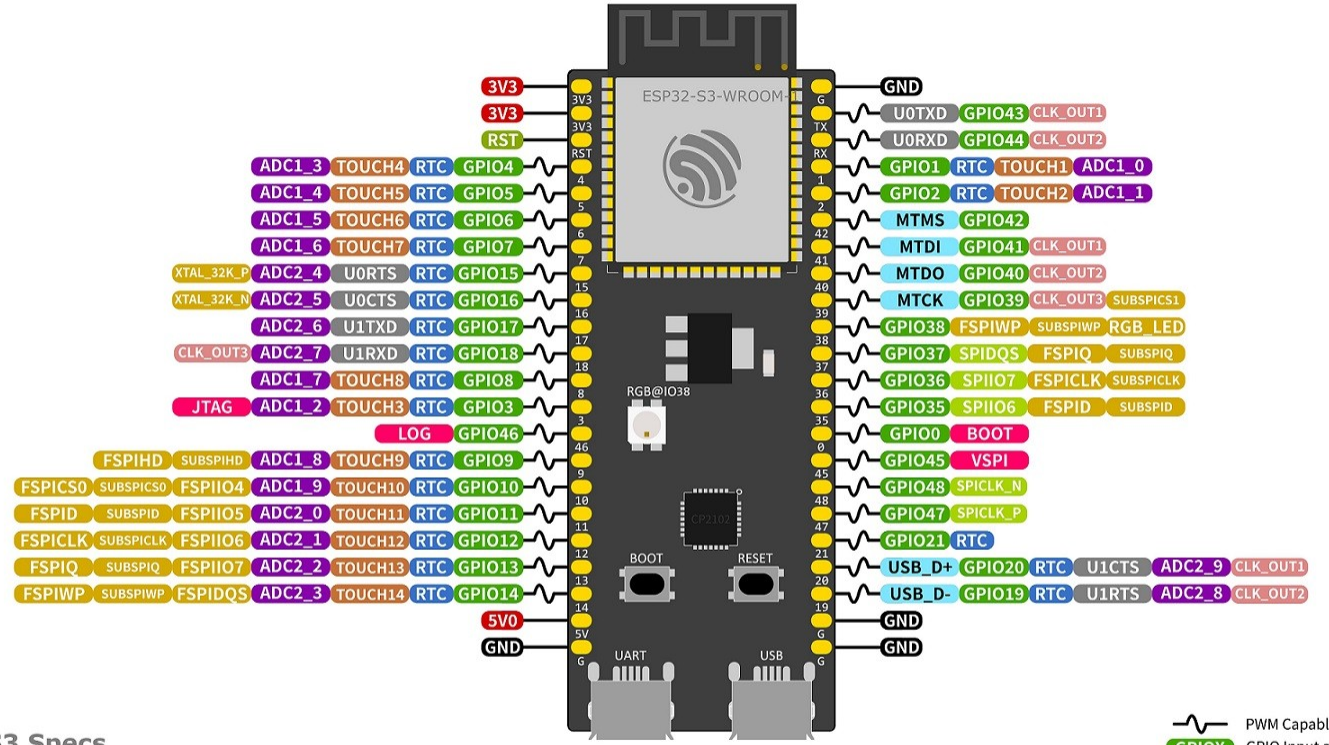
ESP32S3 PINOVI ULAZNI PINOVI



ESP32-S3-DevKitC-1



ESP32-S3 Specs

32-bit Xtensa® dual-core @240MHz
Wi-Fi IEEE 802.11 b/g/n 2.4GHz + BLE 5 Mesh
512 KB SRAM (16 KB SRAM in RTC)
384 KB ROM
45 GPIOs, 4x SPI, 3x UART, 2x I2C,
14x Touch, 2x I2S, RMT, LED PWM, USB-OTG,
TWAI®, 2x 12-bit ADC, 1x LCD interface, DVP

PWM Capable Pin
GPIOX GPIO Input and Output
JTAG/USB JTAG for Debugging and USB
ADCX_CH Analog-to-Digital Converter
TOUCHX Touch Sensor Input Channel
OTHER Other Related Functions
SERIAL Serial for Debug/Programming
STRAP Strapping Pin Functions
RTC RTC Power Domain (VDD3P3_RTC)
GND Ground
PWD Power Rails (3V3 and 5V)

MISC Miscellaneous/SPI functions
CLK_OUTx Clock Output

Funkcije millis() i micros()

Izbjegavanje upotrebe dužeg čekanja u skeču!

Vraća broj mulisekundi (mikrosekundi) koji je prošao od kada je Arduino ploča počela da izvršava određeni program.

```
unsigned long myTime;

myTime = millis();
```

Funkcije millis() i micros() - Overflow

Izbjegavanje problema prelivanja (overflow) kada se koristi millis() i micros()

Funkcije millis() i micros() prelijevaju nakon otprilike 50 dana i 70 minuti, respektivno. Potencijalni problem može se lako izbjeći malim prilagođenjem koda.

Umjesto ovako:

```
int period = 1000;
unsigned long time_now = 0;

void setup() {
    Serial.begin(115200);
}

void loop() {
    if(millis() > time_now + period){
        time_now = millis();
        Serial.println("Hello");
    }

    //Run other code
}
```

Napisati ovako:

```
int period = 1000;
unsigned long time_now = 0;

void setup() {
    Serial.begin(115200);
}

void loop() {
    if(millis() - time_now > period){
        time_now = millis();
        Serial.println("Hello");
    }

    //Run other code
}
```

Na ovo se može gledati kao na poređenje perioda, umesto da se radi sa vremenskim oznakama.

Funkcije millis() i micros() - Overflow

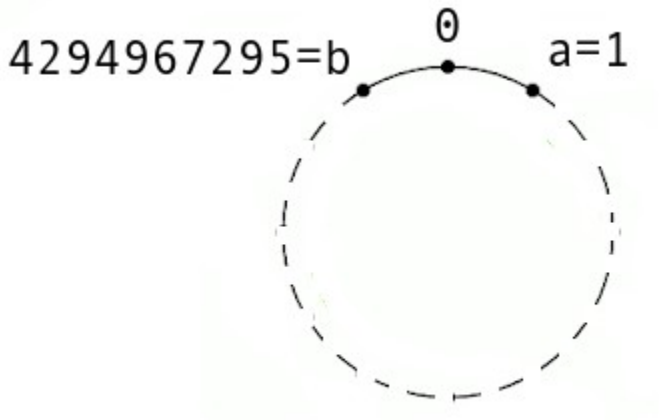
Može izgledati da predhodno nema puno smisla, odnosno da se problem samo pomjera, a da se ne rješava.
Gledajući matematički, nema puno smisla, jer će rezultat

```
millis() - time_now
```

postati negativan, kada millis() „prelije“. Rezultat oduzimanja: od veoma malog cijelog broja veoma velikog cijelog broja.

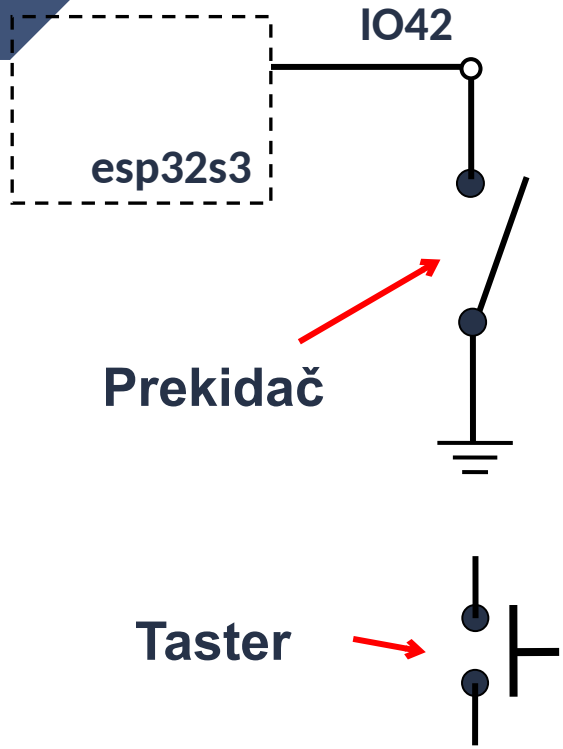
Dokaz da ipak neće:

```
void setup() {
    Serial.begin(115200);

    unsigned long a = 1;

      //unsigned long maximum value
    unsigned long b = 4294967295;

    Serial.println(a-b);
}

void loop() {
}
```



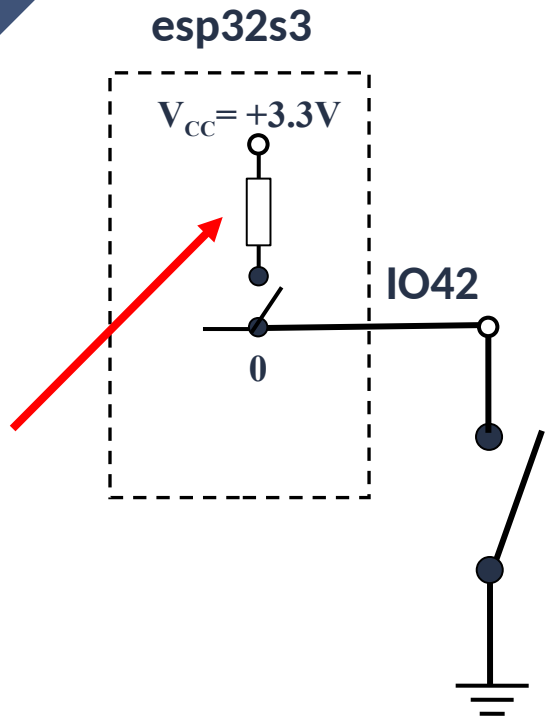
Pinovi – Pin kao ulazni

- Prekidač kao senzor
  - Pr. Senzor pojasa za sjedište u autu
  - Detekcija ***stanja prekidača***
    - Koji tok podataka treba biti za pin IO42?
    - `pinMode(42, INPUT);`
    - Koji će biti napon na IO42 kada je prekidač zatvoren?
    - Koji će biti napon na IO42 kada je prekidač otvoren?



Pin kao ulazni + Pull-up otpornik

- Prekidač kao senzor, nastavak.
  - Učinimo napon na pinu poznatim uključenjem pull-up otpornika.
    - Neka je IO42 ulazni port:
      - `digitalWrite(42,HIGH);`
        uključenje “pull-up” otpornika
      - `pinMode(42,INPUT_PULLUP);`
    - Koji će napon biti na IO42 kada je prekidač otvoren?
    - Koji će napon biti na IO42 kada je prekidač zatvoren?

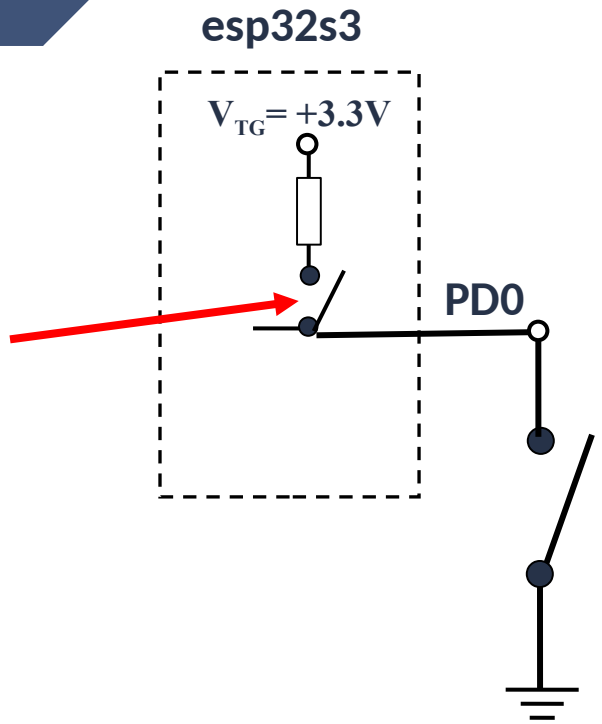


Pin kao ulazni + Pull-up otpornik

- Prekidač kao senzor, nastavak.
  - Za isključenje pull-up otpornika
    - Neka je IO42 ulazni port:

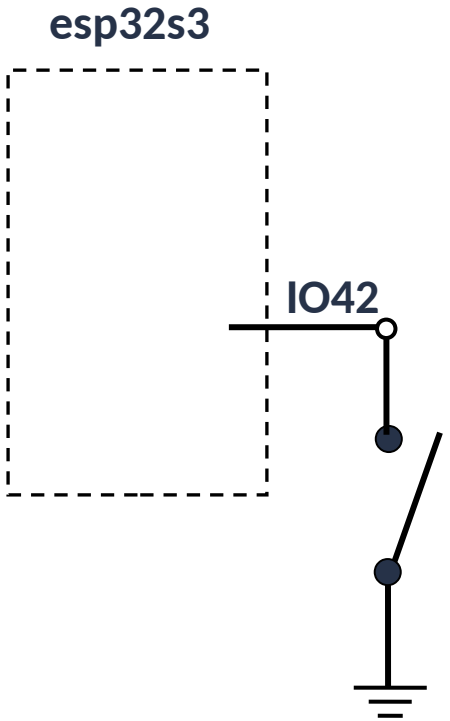
```
digitalWrite(42,LOW);
```

Isključuje “pull-up” otpornik



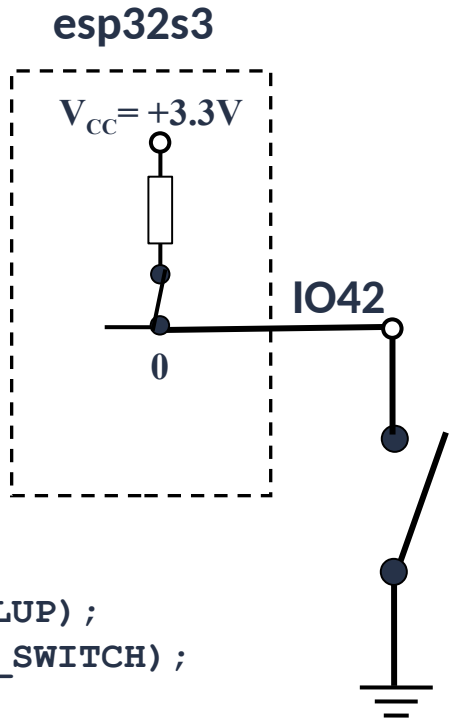
Ulazni digitalni pin – Primjer 1

- 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 oNEmogućeno je uključenje auta kroz poziv funkcije start_disable()
  - Napisaćemo najprije psudokod!



Ulazni digitalni pin – Primjer 1

- Pseudokod:
  Postaviti pin 42 kao ulazni
  Uključiti pull-up otpornik za pin 42
  Očitati napon sa pinu 42 (PIN_42)
  IF PIN_42 napon je LOW (vezan), THEN
  pozovi funkciju start_enable()
  ELSE
  pozovi start_disable()

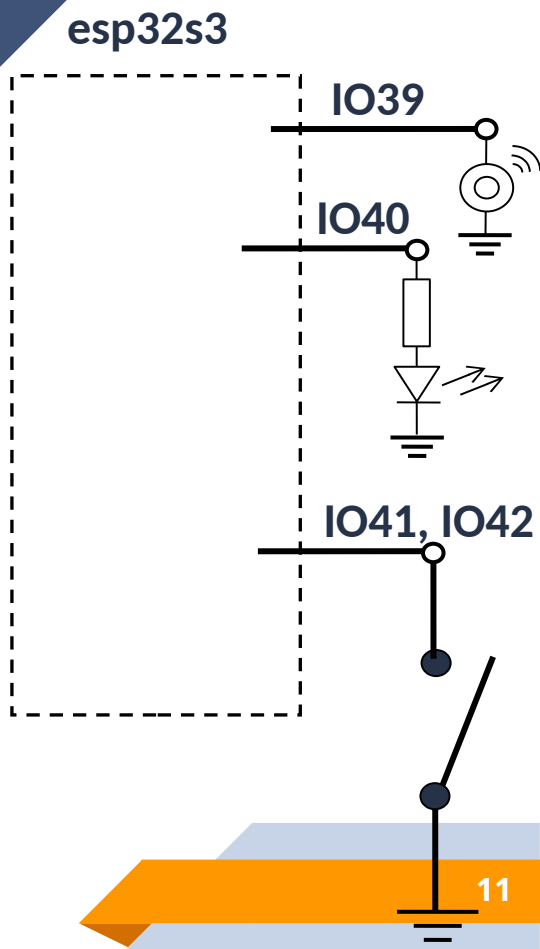


Fragment. Nije cijeli skeč.

```
#define PIN_SWITCH 42
#define LATCHED LOW
pinMode(PIN_SWITCH,INPUT_PULLUP);
belt_state = digitalRead(PIN_SWITCH);
if (belt_state == LATCHED)
{ ig_enable(); }
else
{ ig_disabled(); }
```

Ulazni digitalni pin – Primjer 2

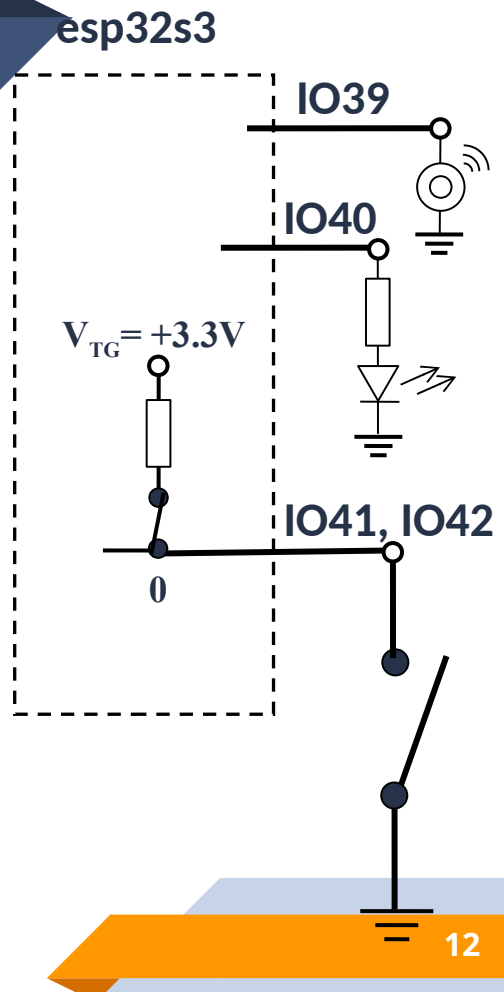
- Čitanje sa pina i upisivanje na pin
  - Napisaćemo nekoliko linija C koda za Arduino, s ciljem uključenja LED (IO40) i zvučnog signala (IO39) ako je ključ u bravi (IO41 zatvoren), ali pojas vozača nije vezan (IO42 otvoren)
  - Najprije pseudokod



Ulazni digitalni pin – Primjer 2

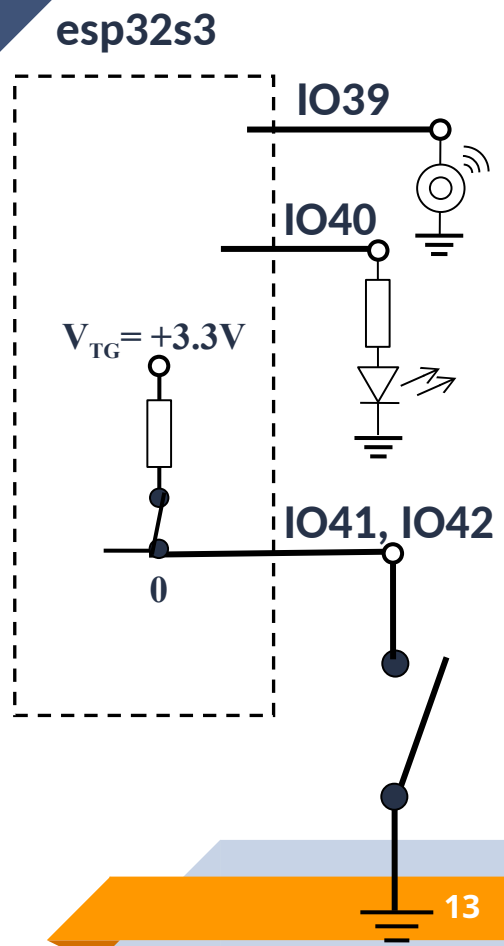
- Pseudokod:

Postavljanje toka podataka za pinove
Postaviti IO41 i IO42 kao ulaze
Uključiti pull-up otpornike za IO41 i IO42
Postaviti IO39 i IO40 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



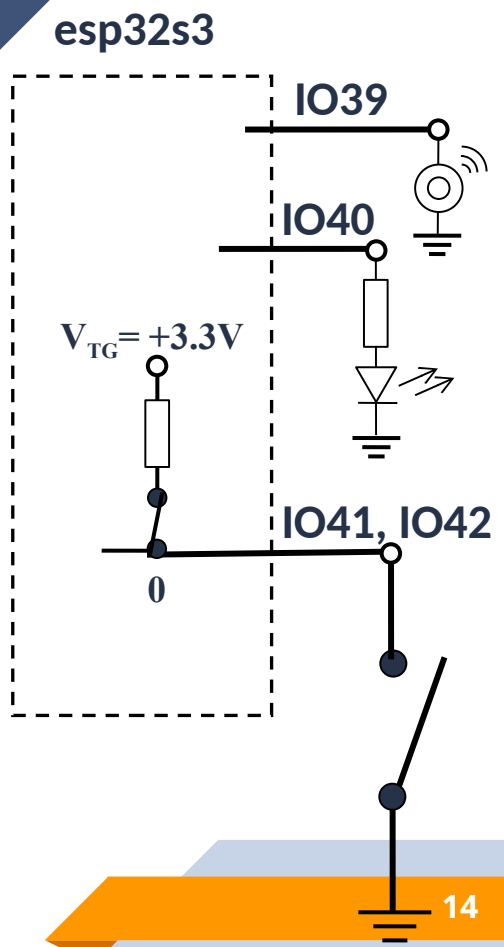
Ulazni digitalni pin – Primjer 2

```
#define PIN_IGNITION  41
#define PIN_SEATBELT  42
#define PIN_LED  40
#define PIN_BUZZER  39
#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 loop code */
```



Ulazni digitalni pin – Primjer 2

```
/* 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);
   }
}
```



Serijska komunikacija

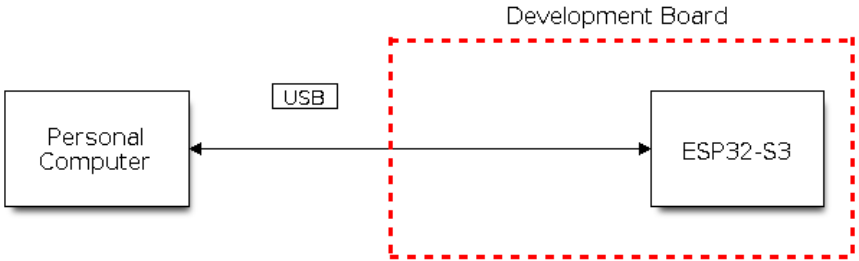
Uspostavljanje serijske veze sa ESP32-S3 ciljnim uređajem može se izvršiti korišćenjem

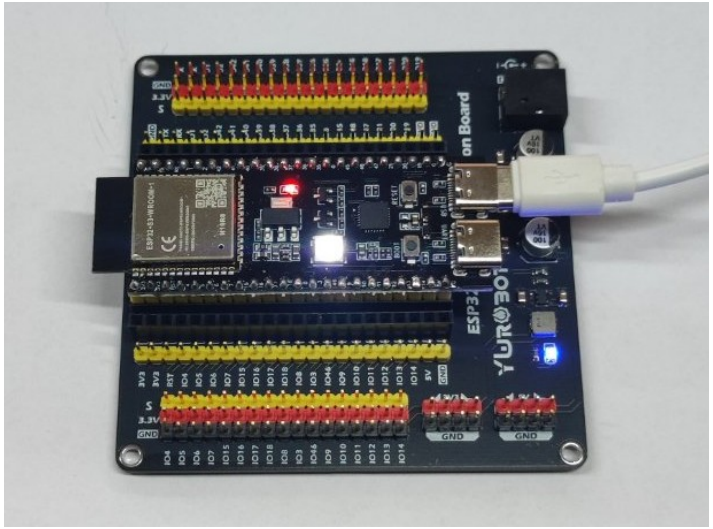
- USB periferije ili
- USB-to-UART mosta.

ESP32S3 podržava USB

ESP32-S3 podržava USB.

USB-to-UART most nije neophodan i uređaj se može direktno flešovati.





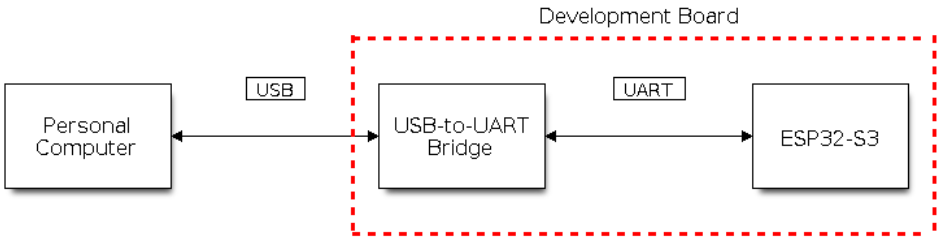
Takođe arduino skeč može direktno razmjenjivati podatke sa PC-em, bez potrebe za USB-to-UART mostom.

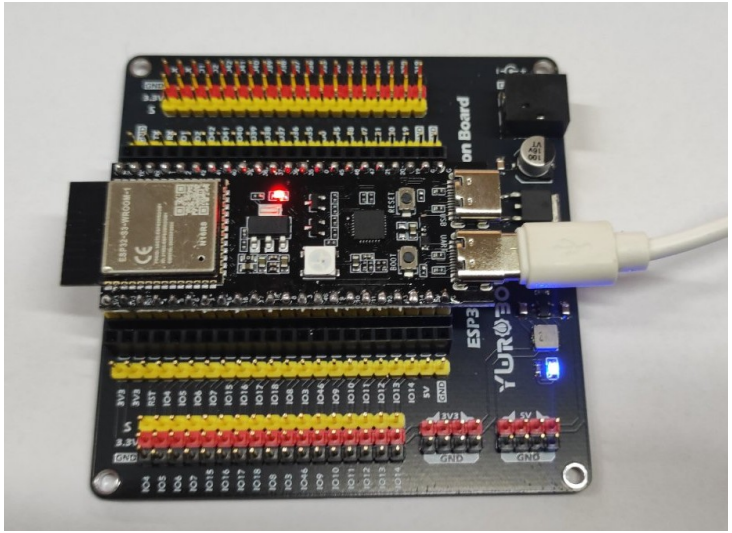
Ipak na početku skeča, potrebno je dodati liniju:

#define CONFIG_USB_CDC_ENABLED 1

ESP32S3 ima ugrađen USB-to-UART most

ESP32S3 koju koristimo ima ugrađen i USB-to-UART most.
U tom slučaju veza između ličnog računara i mosta je USB, dok je veza između mosta i ESP32-S3 UART.





U ovom sličaju arduino skeč razmjenjuje podatke sa PC-em, preko USB-to-UART mosta.

Na početku skeča nije potrebno je dodati liniju koda kao kod direktne USB konekcije.

Dva različita komunikaciona protokola

Serijski (TTL):

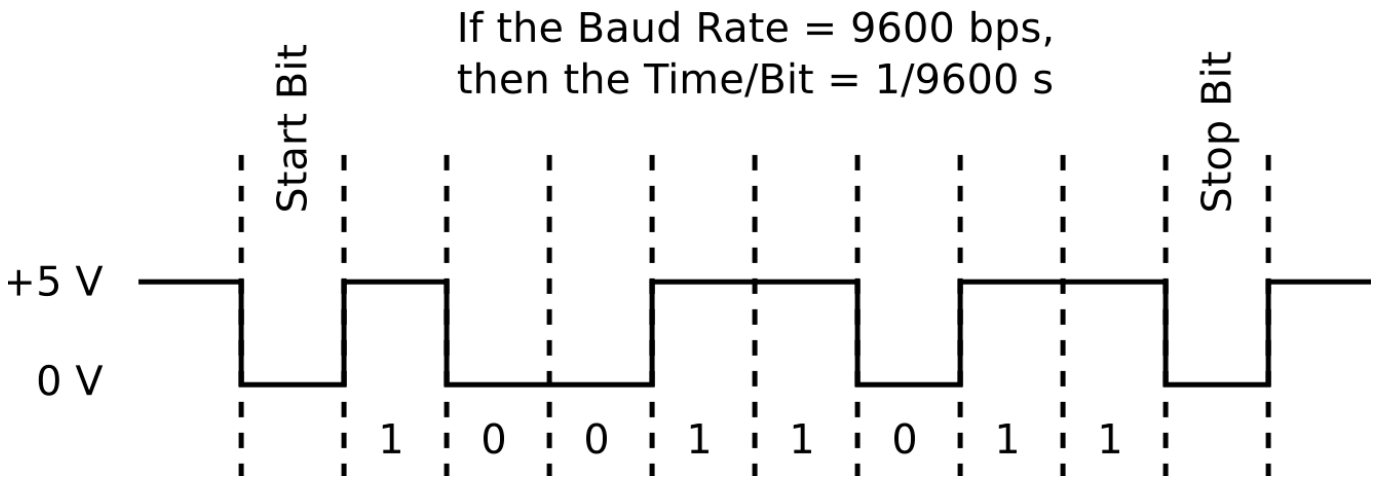


Image from http://www.fiz-ix.com/2013/02/introduction-to-arduino-serial-communication/

Dva različita komunikaciona protokola

USB protokol

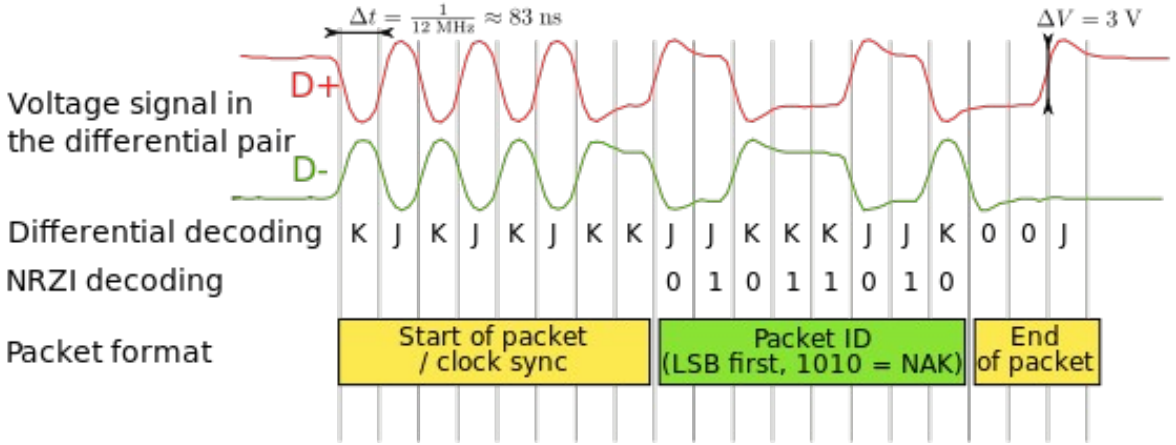


Image from http://en.wikipedia.org/wiki/USB

Osnovne komande za slanje podataka iz skeča

- Serial.begin()
  - pr., Serial.begin(9600)

- Serial.print() ili Serial.println()
- Serial.write()

Primjer 3 – Brojanje pritisaka tastera

```
#define TASTER 41
#define LED 40
#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;
}
```

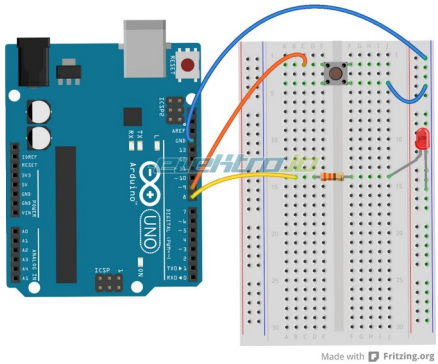
```
void loop() {
  if (millis() - nextTstCheck > TST_CHECK_PERIOD) {
    Taster = digitalRead(TASTER);
    if (!Taster && pTaster) {
      brojPritisaka++;
      Serial.println(brojPritisaka);
    }
    pTaster=Taster;
    nextTstCheck=millis();
  }

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

ZADACI ZA VJEŽBU – PRVI ZADATAK

Upotrijebiti taster za kontrolu načina rada LED.
Prvim neparnim pritiskom tatera i po njegovom otpuštanju, uključiti LED.
Drugim neparnim pristiskom tastera i po njegovom otpuštanju, inicirati treperenje LED, 300ms ukuljučena, 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, odnono 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)



ZADACI ZA VJEŽBU – DRUGI ZADATAK

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

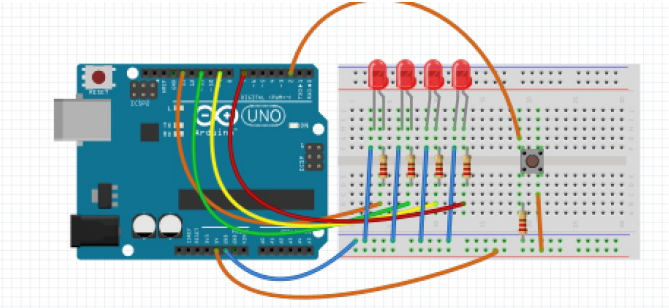
Dva pritiska taster unutar 1000ms iniciraju trčeće svjetlo. Promjena stanja svjetla svakih 500ms. Smjer trčećeg svjetla se odredjuje prekidačem. Prekidač otvoren - jedan smjer, prekidač zatvoren - drugi smjer.

Tri pritiska tastera unutar 1000ms iniciraju trečeću “tamu”. Promjena stanja svjetla svakih 500ms. Smjer trčeće tame se odredjuje 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.

ZADACI ZA VJEŽBU – 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 serijskom monitoru, sljedećim porukama:
- „OK nivo“. Poruku ispisati samo jednom, neposredno po uspostavljanja tog stanja.
- „Nizak nivo“. Poruku ispisati samo jednom, neposredno po uspostavljanja tog stanja.
- „Visok nivo“. Poruku ispisati samo jednom, neposredno po uspostavljanja tog stanja.
- „Neispravnost”. Poruku ispisati jednom u dvije sekunde, dok god stanje niespravnosti traje, uz kratkotrajan zvučni signal.

(2-1 poen)