RAČUNARSKE PERIFERIJE I INTERFEJSI

O PREDMETU

Studijski programi za koje se organizuje : -Specijalističke studije Elektrotehnike, studijski program Elektronika, Telekomunikacije i Računari (ETR), smjer Elektronika.
Uslovljenost drugim predmetima: Nema formalnih uslova. Podrazumijeva se poznavanje C/C++ jezika.
Ciljevi izučavanja predmeta: Interfejsi (međusklopovi) i periferije (spoljni uređaji) povezuju računare sa spoljašnjom sredinom. Cilj ovog predmeta je da studenti ovladaju znanjem i tehnikama, pomoću kojih će moći da uz pomoć računara (automatski) prikupljaju informacije iz spoljašnjeg svijeta i da upravljaju procesima van računara. Osim teoretskog dijela, značajna pažnja se poklanja praktičnom radu.
Metod nastave i savladanja gradiva: Predavanja, računske vježbe i vježbe u računarskoj učionici / laboratoriji. Učenje i samostalna izrada praktičnih zadataka. Konsultacije.

O PREDMETU

Sadržaj predmeta:	
Pripremna sedmica	Priprema i upis semestra
I sedmica	Uvod; Upoznavanje sa predmetom, ciljevima i načinom rada;
II sedmica	Mikrokontroleri; Razvojne ploče; Arduino Uno;
III sedmica	Portovi: izlazni, ulazni;
IV sedmica	Koračni motori;
V sedmica	Optički interfejsi: Inkrementalni davači položaja, optički difuzioni senzor, ...;
VI sedmica	***Slobodna sedmica***
VII sedmica	*I provjera znanja;*
VIII sedmica	D/A konverzija; Upravljanje analognim uređajima;
IX sedmica	A/D konverzija: prateća, sukcesivna, paralelna, V/f i f/V konvertori;
X sedmica	Komunikacioni interfejsi: -paralelni, -serijski;
XI sedmica	Priključci (sockets);
XII sedmica	*II provjera znanja;*
XIII sedmica	Modemi: dial-up, govorni, adsl, gprs; AT komande;
XIV sedmica	Modemi sa integrisanim script interpreterom (Pyton); GPS;
XV sedmica	Interfejsi u industriji; Vizuelizacija industrijskih procesa;
XVI sedmica	*Završni ispit*
Završna sedmica	Ovjera semestra i upis ocjena
XVIII-XXI sedmica	Dopunska nastava i popravni ispitni rok

O PREDMETU

Opterećenje studenata na predmetu	
Sedmično **6 kredita x 40/30 = 8 časova** **Struktura:** **3** časa predavanja **1** čas računskih vježbi **4** časa samostalnog rada, uključujući konsultacije	**U toku semestra** **Nastava i završni ispit**: (8 časova) x 16 = **128 časova** **Neophodne pripreme** prije početka semestra (administracija, upis, ovjera) 2 x (8 časova) = **16 časova** **Ukupno opterećenje za predmet 6x30 = 180 časova** **Dopunski rad** za pripremu ispita u popravnom ispitnom roku, uključujući i polaganje popravnog ispita od 0 do 30 časova (preostalo vrijeme od prve dvije stavke do ukupnog opterećenja za predmet 150 časova) **Struktura opterećenja**: 128 časova (Nastava)+16 časova (Priprema)+36 časova (Dopunski rad)
Studenti su obavezni da pohađaju nastavu, rade i predaju sve domaće zadatke, odrade laboratorijske vježbe i obje provjere znanja.	

O PREDMETU

Literatura: Osnovna i pomoćna literatura u elektronskom obliku na www.etf.ac.me Praktični zadaci za laboratorijske vježbe na www.etf.ac.me Z.Mijanović i ostali, »Računarski interfejsi i periferije«, Univerzitet Crne Gore
Oblici provjere znanja i ocjenjivanje: -Dvije provjere znanja po 25 poena -Završni ispit 50 poena. -Prelazna ocjena se dobija ako se kumulativno sakupi najmanje 50 poena.
Posebnu naznaku za predmet: U slučaju da je to potrebno nastava se može izvoditi i na engleskom jeziku.

Što je mikrokontroler?

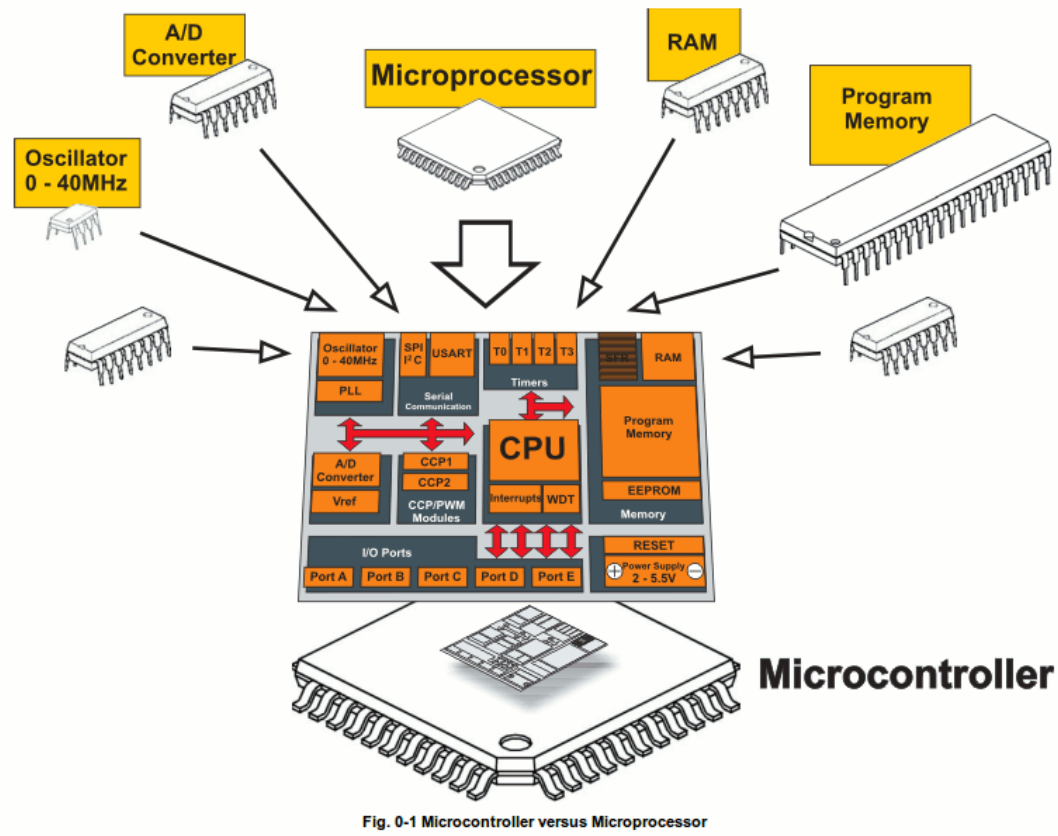
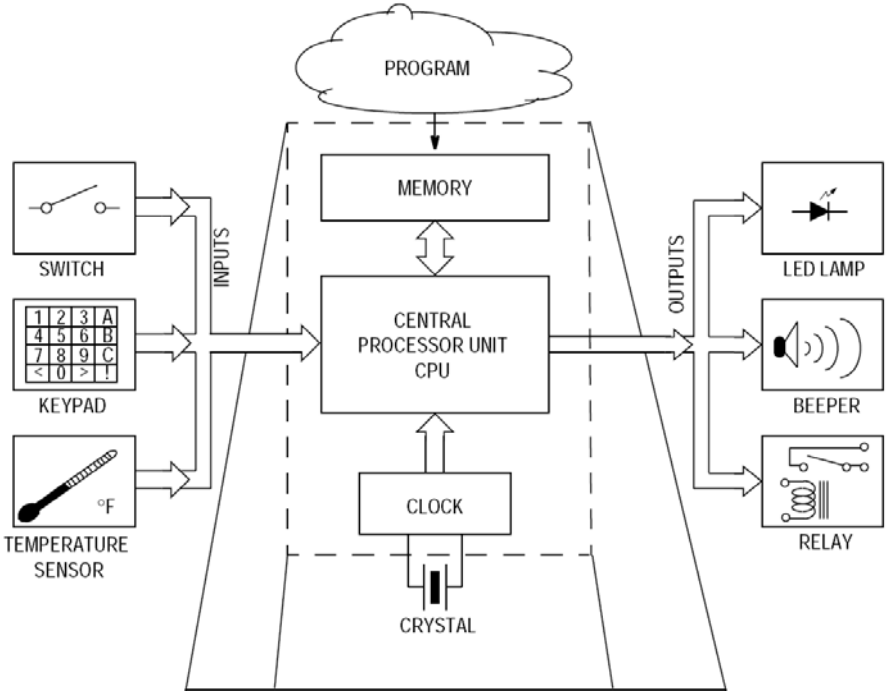
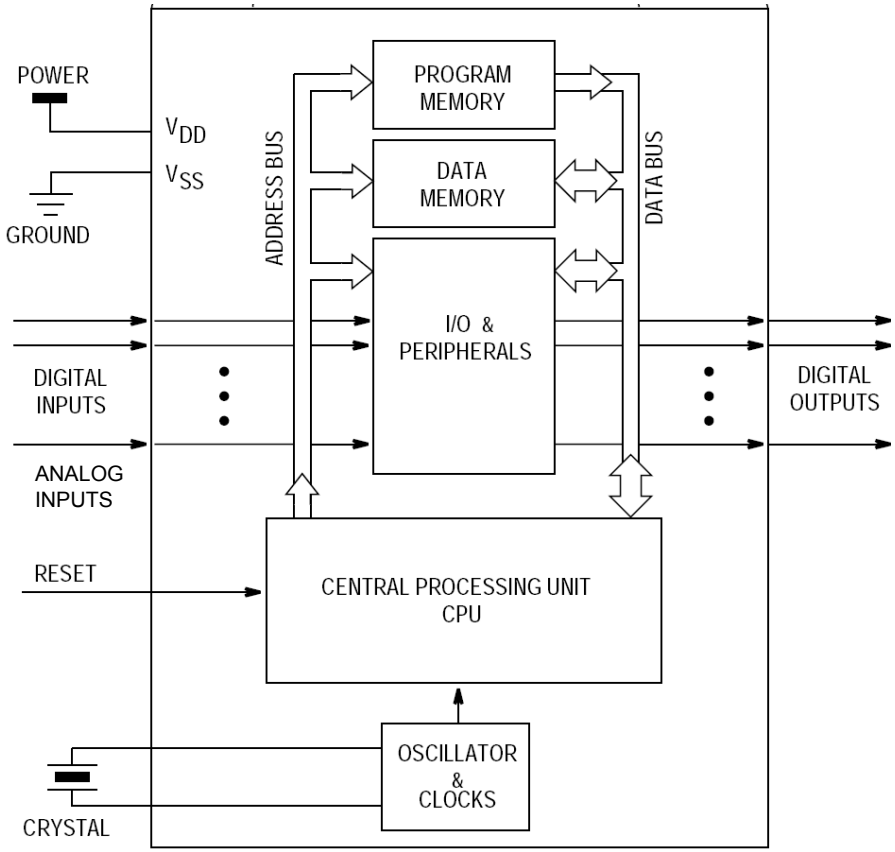


Fig. 0-1 Microcontroller versus Microprocessor

- Mali kompjuter u jednom čipu
  - Sadrži procesor, memoriju, i ulaze/izlaze
- Tipično je „**ugrađen**" unutar uređaja i kontroliše njegov rad.
- Mikrokontroler je često mali i jeftin

Što je mikrokontroler?



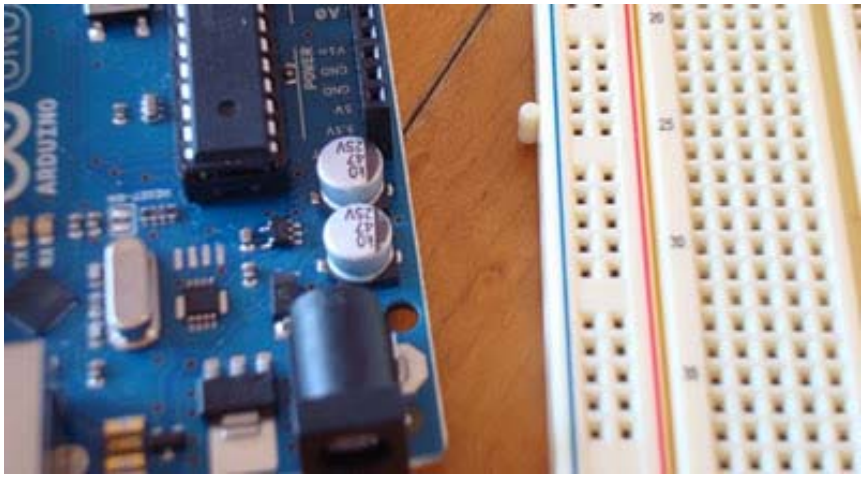


Što je razlika između '*Digital* Input' i '*Analog* Input'?

Mikrokontroleri – definicija

- Programeri rade u virtuelnom svijetu.
- Uređaji rade u fizičkom svijetu.
- Kako povezati vituelni i fizički svijet?
- Uvedite mikrokontroler.
- Mikrokontroler je u osnovi mali računar koji posjeduje programabilne ulaze i izlaze opšte namjene.
- Ulazi mogu biti upravljani od strane fizičkog okruženja dok izlazi mogu upravljati fizičkim okruženjem.

Što je razvojna ploča?



- Štampana matična ploča dizajnirana da olakša rad sa mikrokontrolerom

- Razvojna ploča tipično uključuje:
  - napojno kolo;
  - programerski interfejs;
  - Lako dostupne ulazno/izlazne pinove.

Arduino – Zvanična definicija

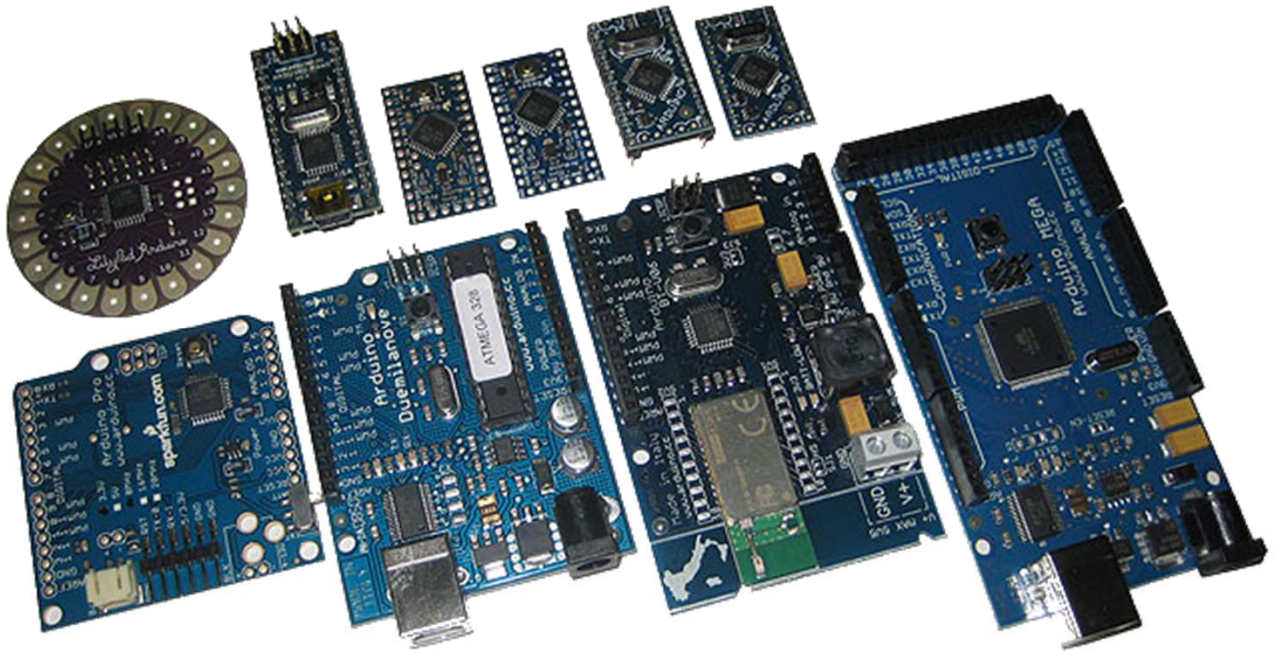
- Uzeto sa zvaničnog web sajta (arduino.cc):
  - Arduino je open-source elektronska prototipna platforma zasnovana na fleksibilnom, jednostavnom za upotrebu, hardveru i softveru.
  - Namijenjen je dizajnerima, hobistima, i svima drugima koji su zainteresovani za kreiranje interaktivnih objekata i okruženja.

Zašto Arduino?

- Arduino platforma je postala standard.
  - Postoji puno realizovanih, dostupnih, projekta koji koriste arduino platformu.
- Teži ravnoteži između jednostavnosti upotrebe i korisnosti.
  - Programski jezici se uglavnom vide kao glavna poteškoća.
  - Arduino C je značajno uproštena verzija C++.
- Nije skup.

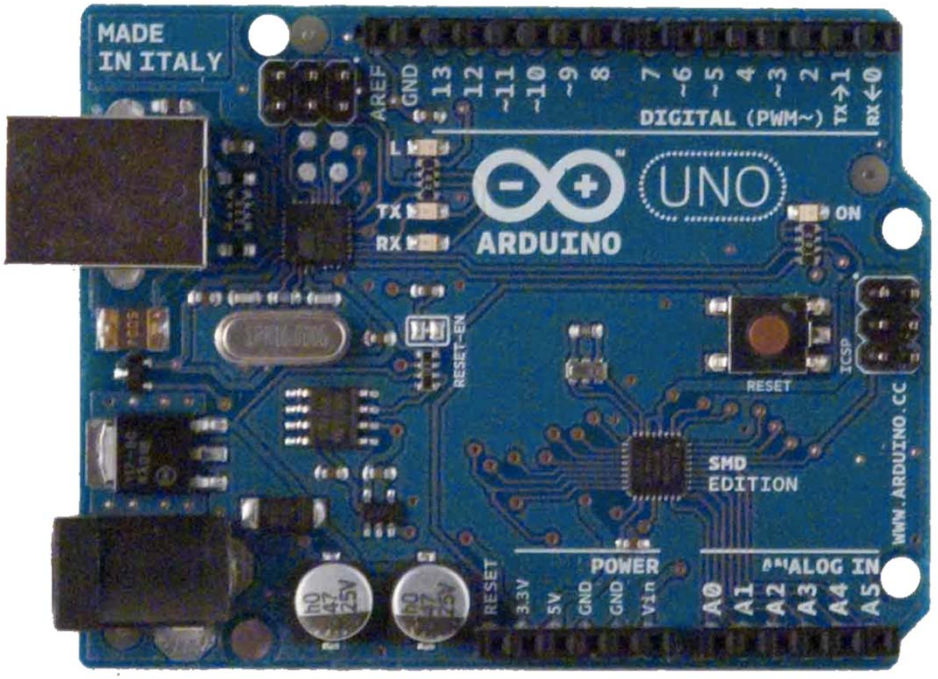
Tipovi Arduino-a

- Više različitih verzija
  - Broj ulaznih/izlaznih kanala
  - Oblik (gabariti)
  - Procesorska snaga
- Leonardo
- Due
- Micro
- LilyPad
- Esplora
- Uno

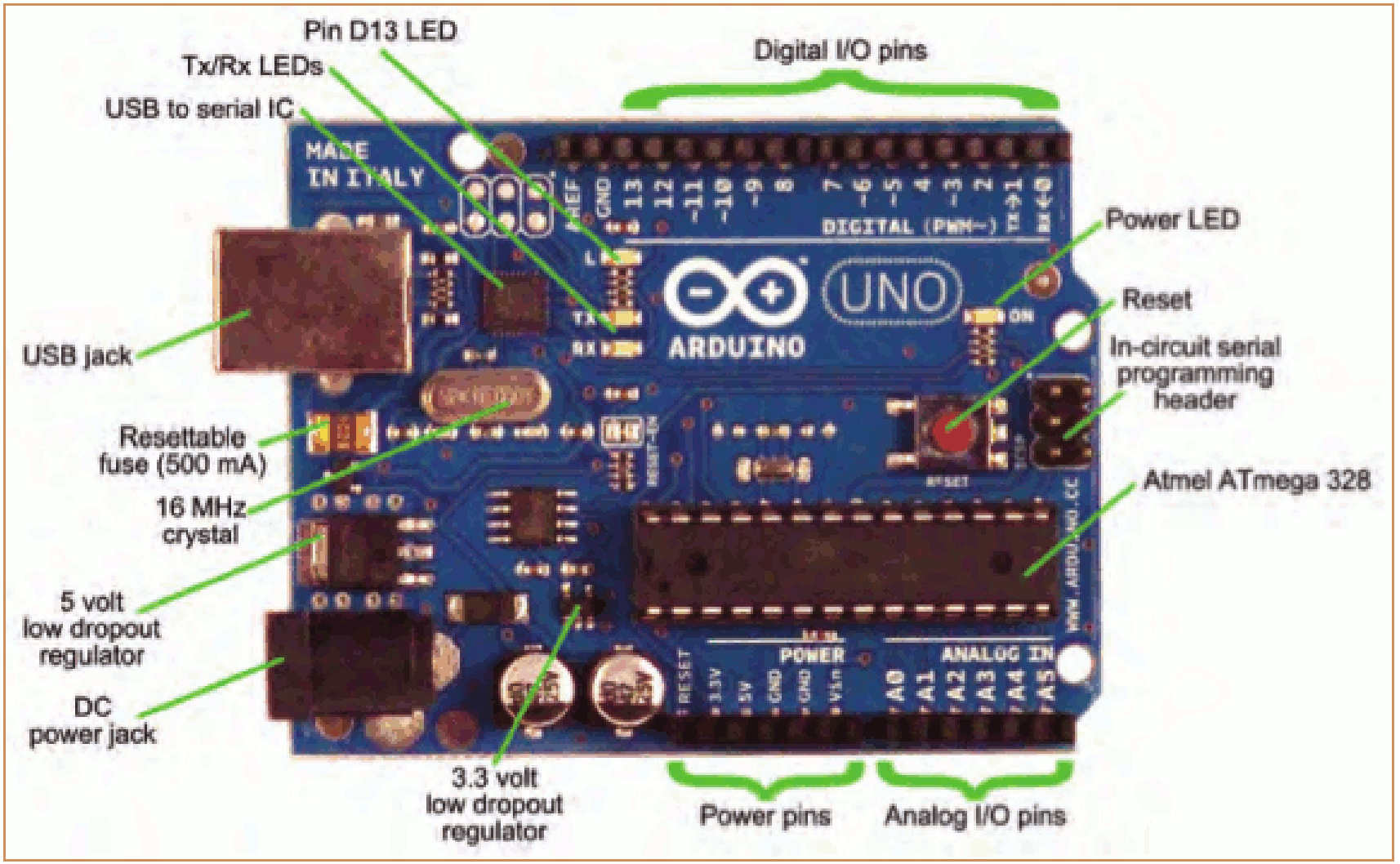


Arduino Uno

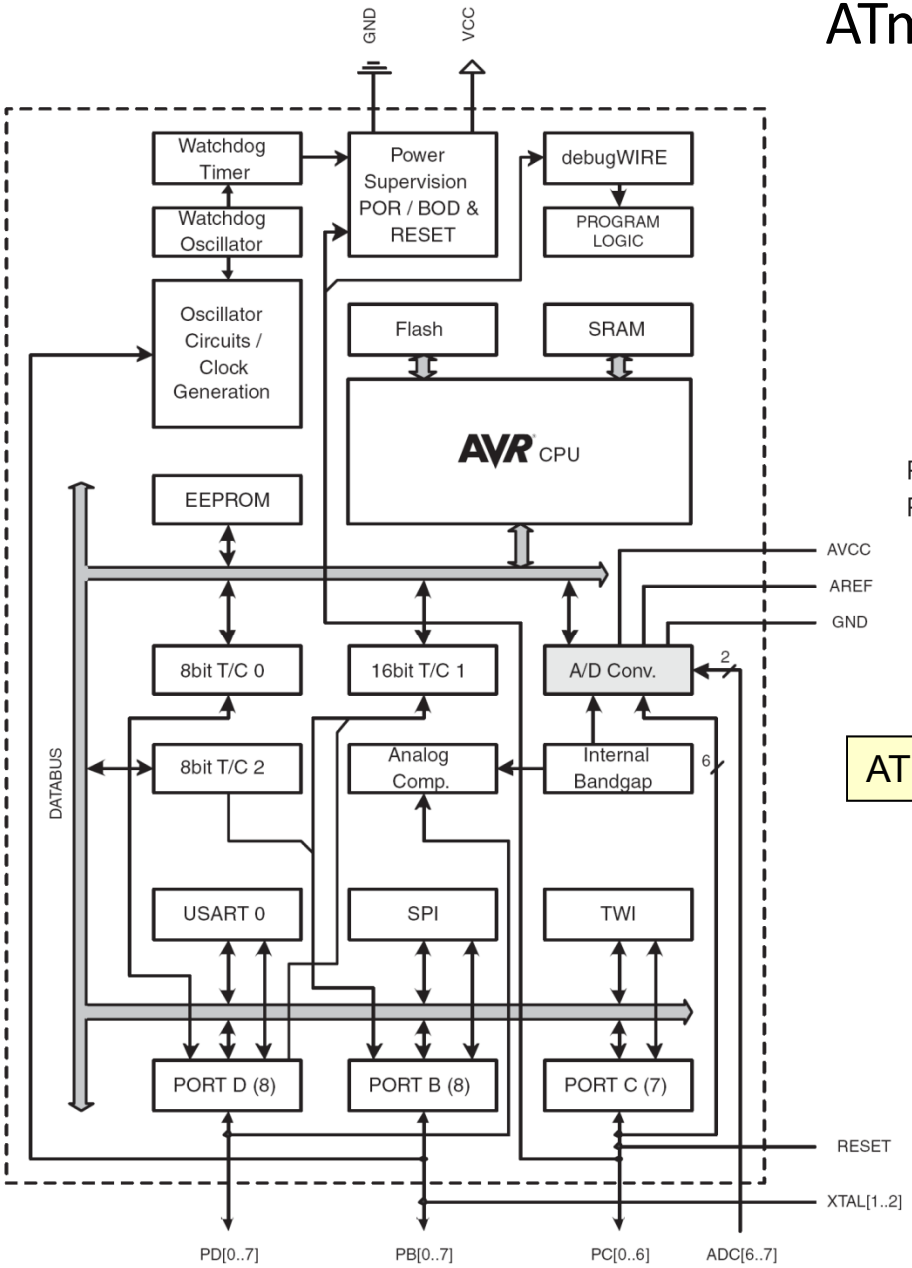
- Pinovi su grupisani u 3 grupe:
  - 14 digitalnih pinova
  - 6 analognih pinova
  - Napajanje
  - Pojavio se 2010

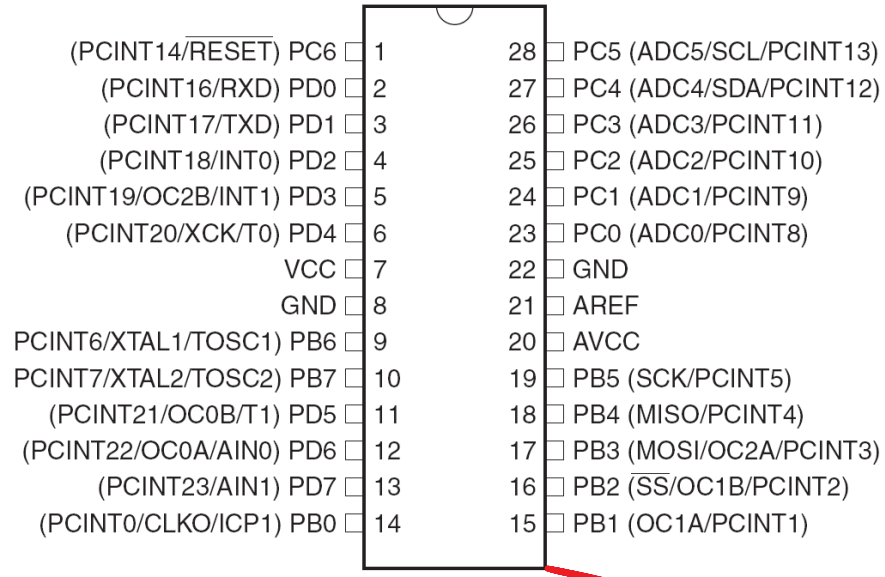


Arduino Uno razvojna ploča



ATmega328 unutrašnja architectura





ATmega328 data sheet pp. 2, 5

Making-robots-with-arduino.pdf

Specification

ATmega328 karakteristike

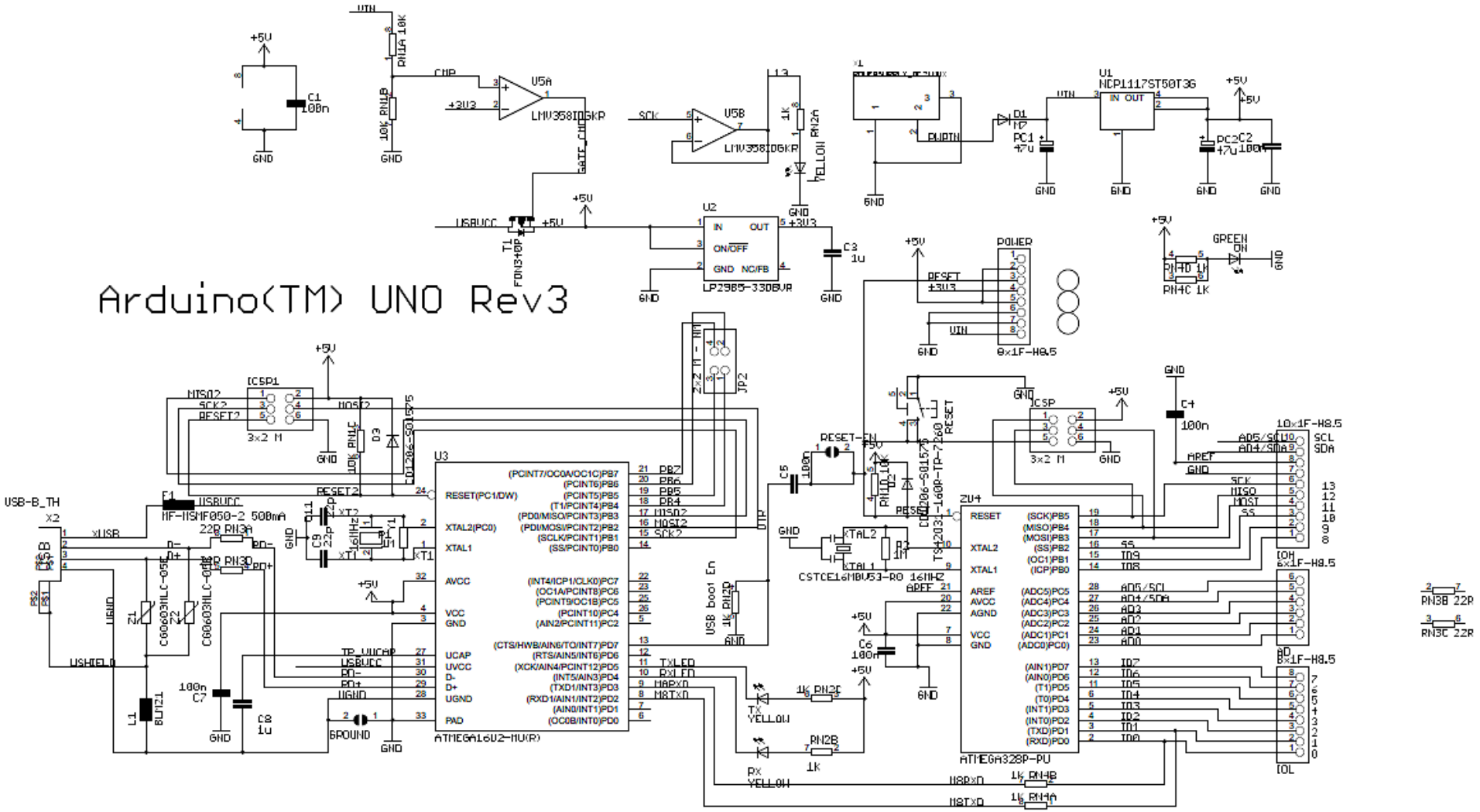
Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory
  - 256/512/512/1K Bytes EEPROM
  - 512/1K/1K/2K Bytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C(1)
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
    - Temperature Measurement
  - 6-channel 10-bit ADC in PDIP Package
    - Temperature Measurement
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Byte-oriented 2-wire Serial Interface (Philips I^2C compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
  - 1.8 - 5.5V
- Temperature Range:
  - -40°C to 85°C
- Speed Grade:
  - 0 - 4 MHz@1.8 - 5.5V, 0 - 10 MHz@2.7 - 5.5.V, 0 - 20 MHz @ 4.5 - 5.5V
- Power Consumption at 1 MHz, 1.8V, 25°C
  - Active Mode: 0.2 mA
  - Power-down Mode: 0.1 µA
  - Power-save Mode: 0.75 µA (Including 32 kHz RTC)

ATmega328 data sheet p. 1

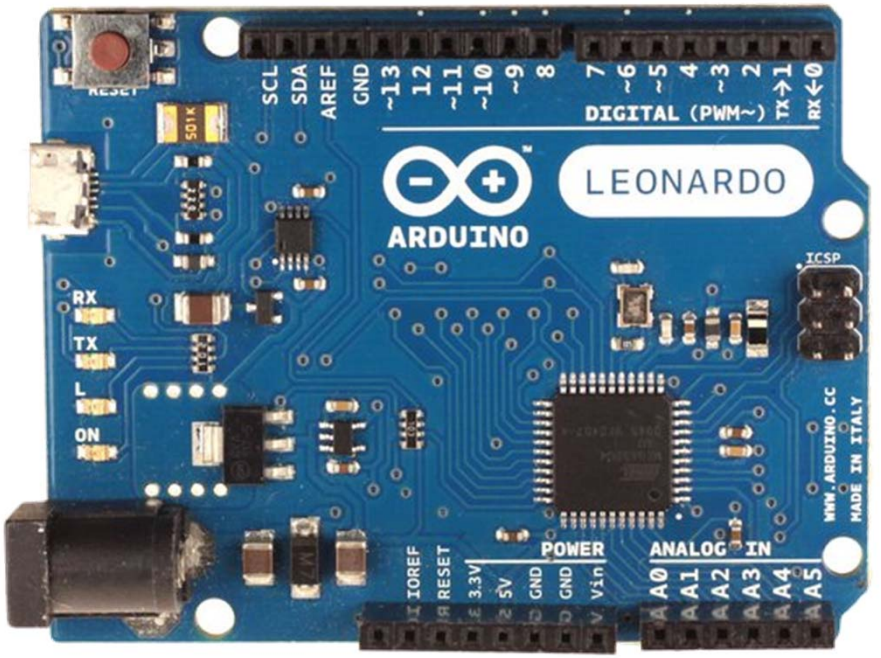
http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet.pdf

Arduino Uno – električna šema



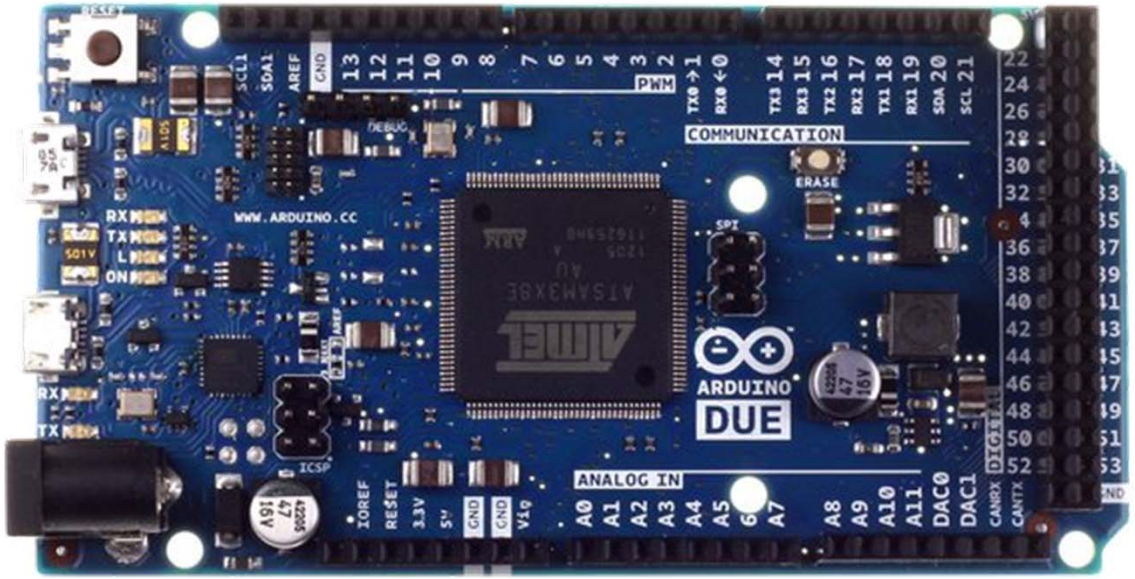
Leonardo

- U poređenju sa Uno, malo unapijedjen.
- Koristi ATmega32u4 mikrokontroler koji ima ugrađenu USB komunikaciju
  - Nema potrebe za dodatnim mikrokontrolerom
  - Može se prikazati PC-u kao miš ili tastatura



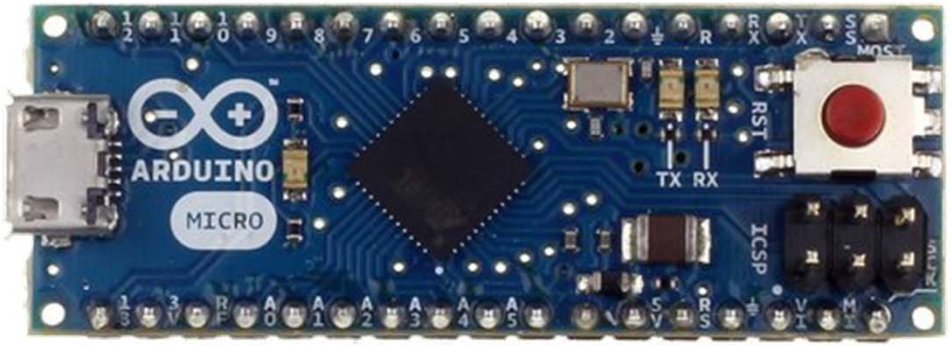
Due

- Mnogo brži procesor, mnogo više pinova
- Radi na 3.3 volta
- Izgledom sličan Mega



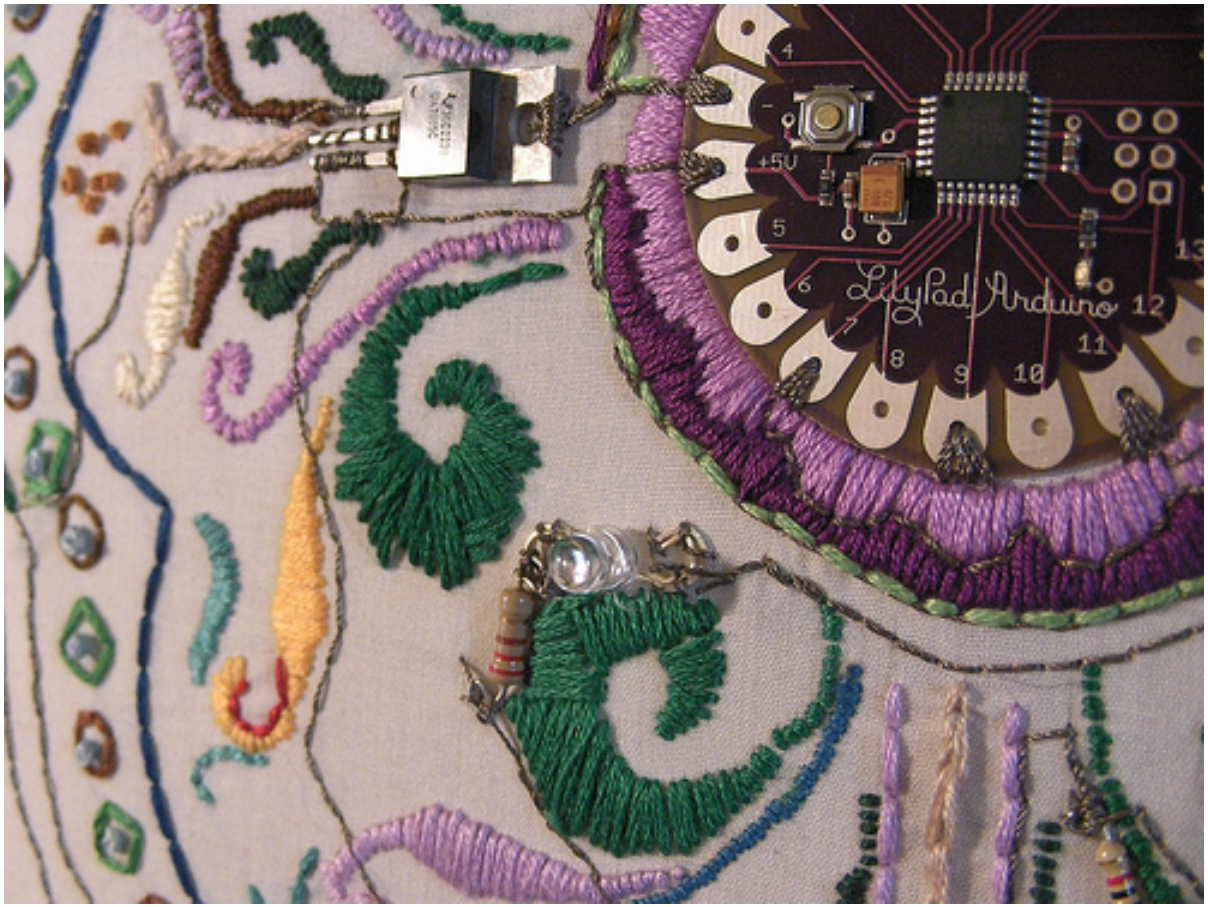
Micro

- Kad je veličina važna: Micro, Nano, Mini
- Uključuju sve funkcionalnosti Leonardo-a



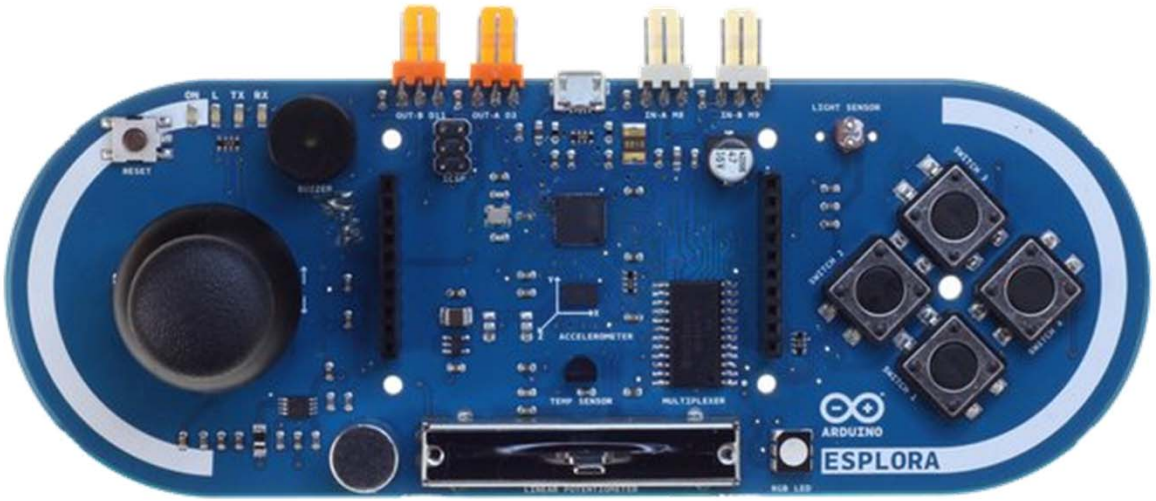
LilyPad

- LilyPad je pogodan za primjenu na odjeći.



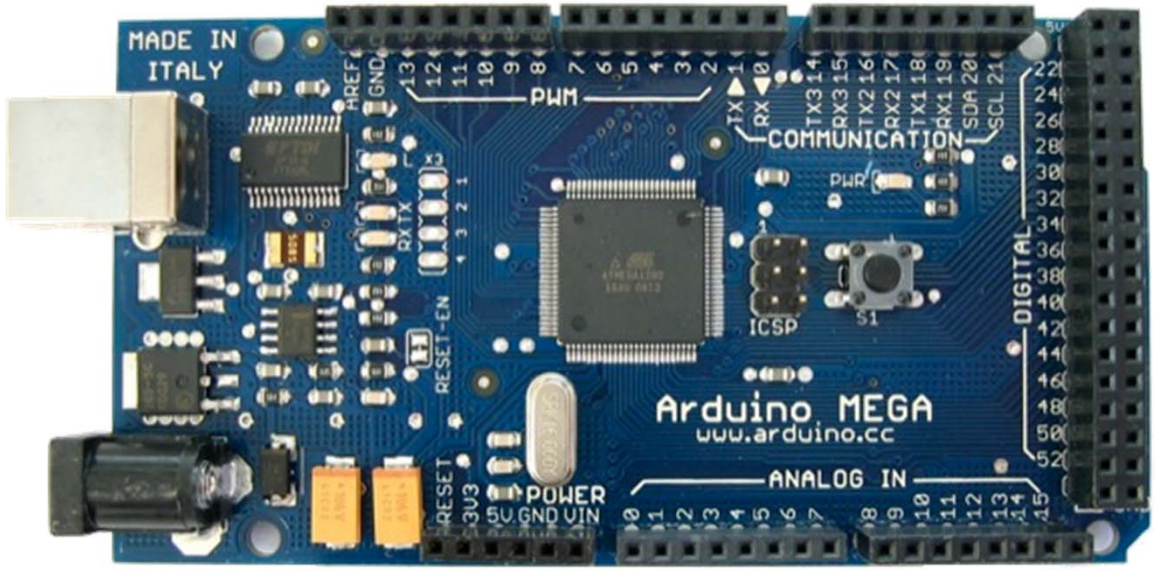
Esplora

- Game controller
- Sadrži džojstik, četiri tastera, linearni poteciometar (klizač), mikrofon, svjetlosni senzor, senzor temperature, tro-osni akceleromatar.
- Nema standardi set IO pinova.

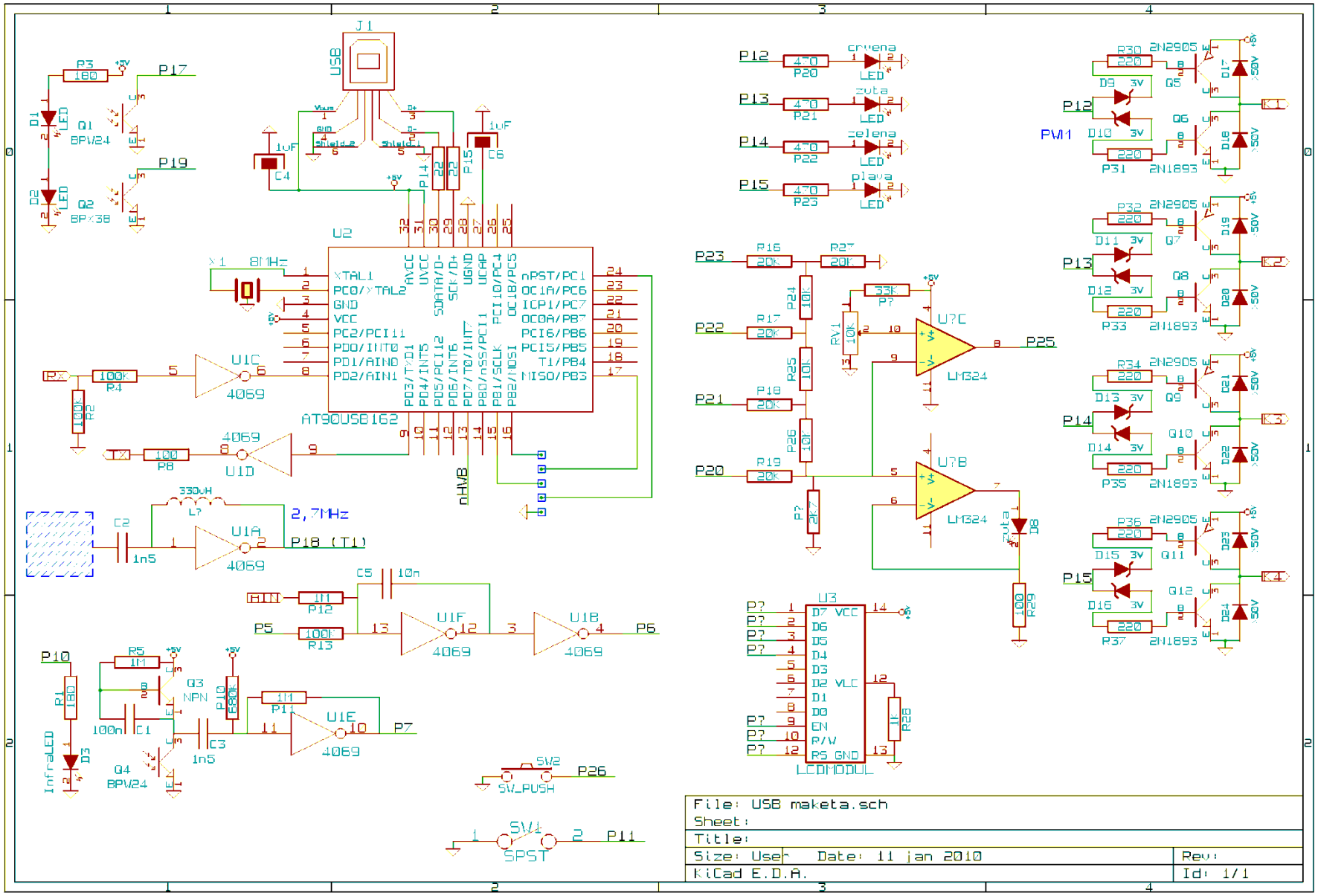


Mega

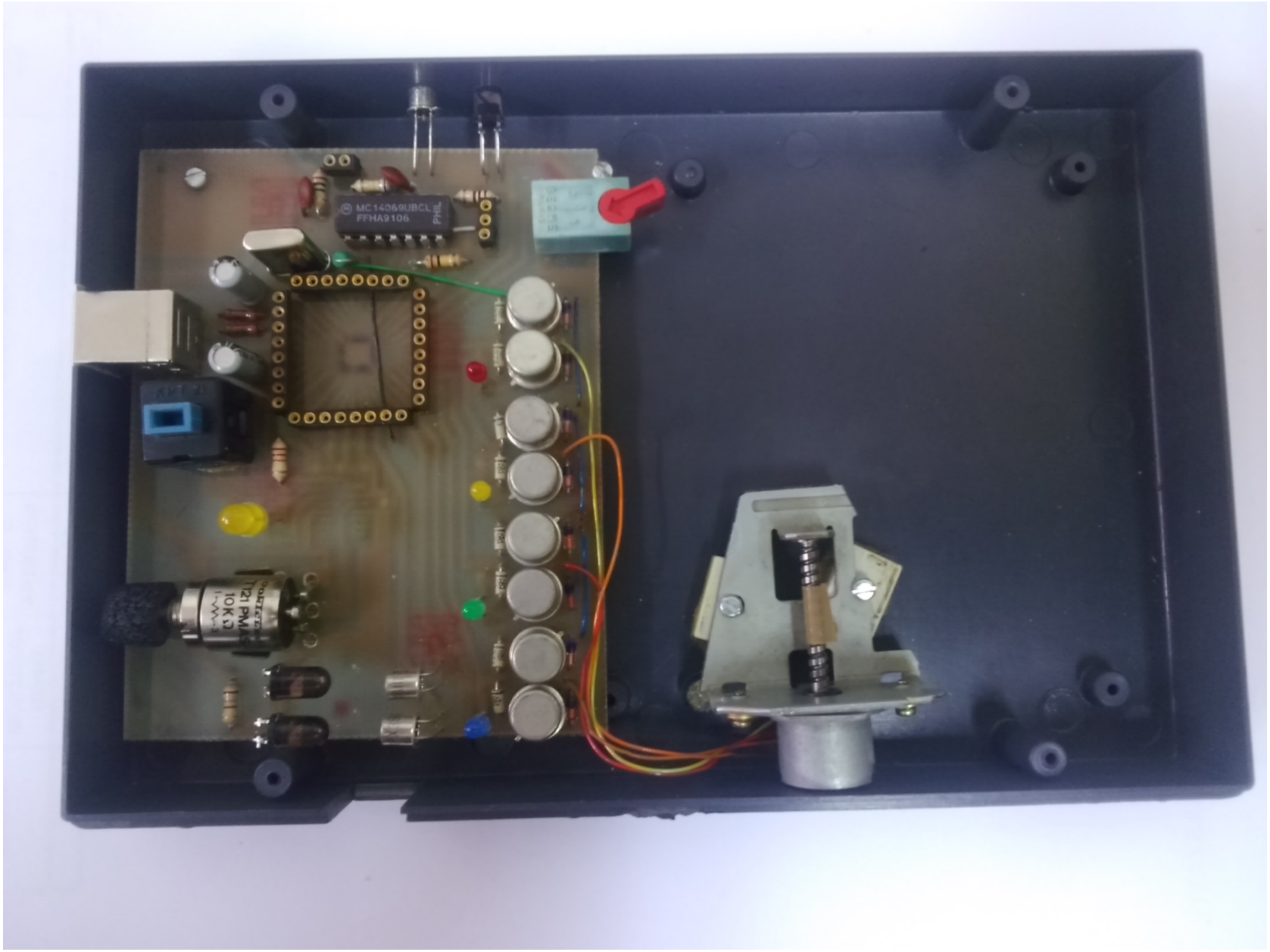
- U poređenju sa Uno, Mega:
  - Mnogo više komunikacionih pinova
  - Više memorije



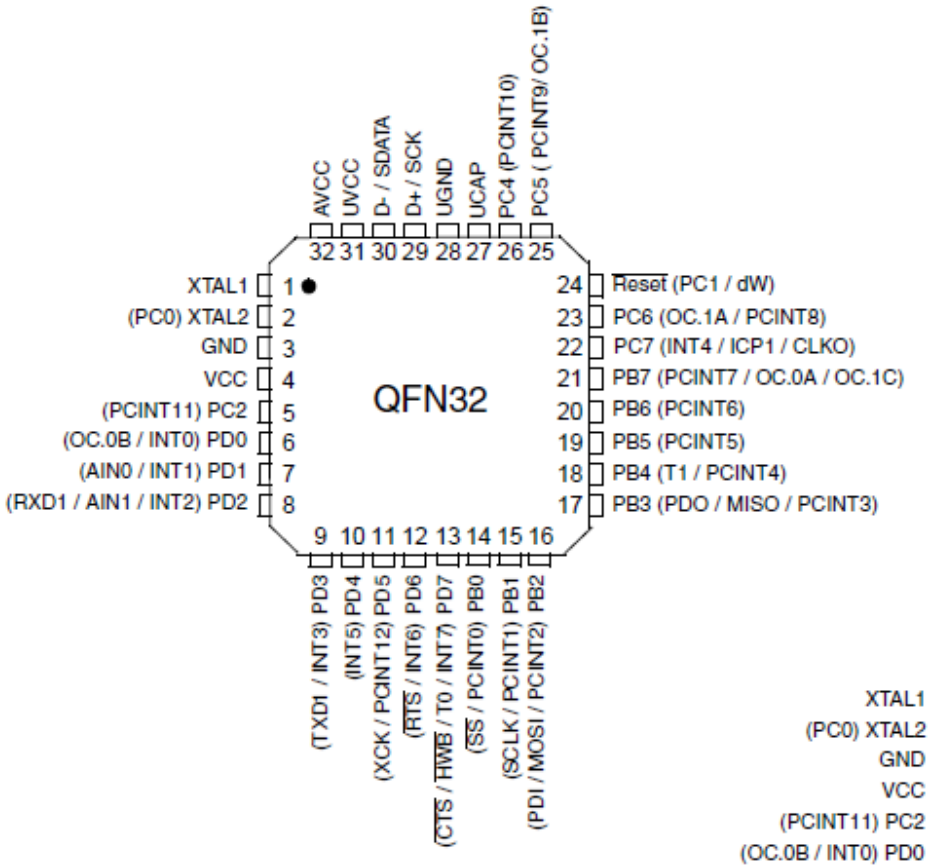
Naša razvojna ploča

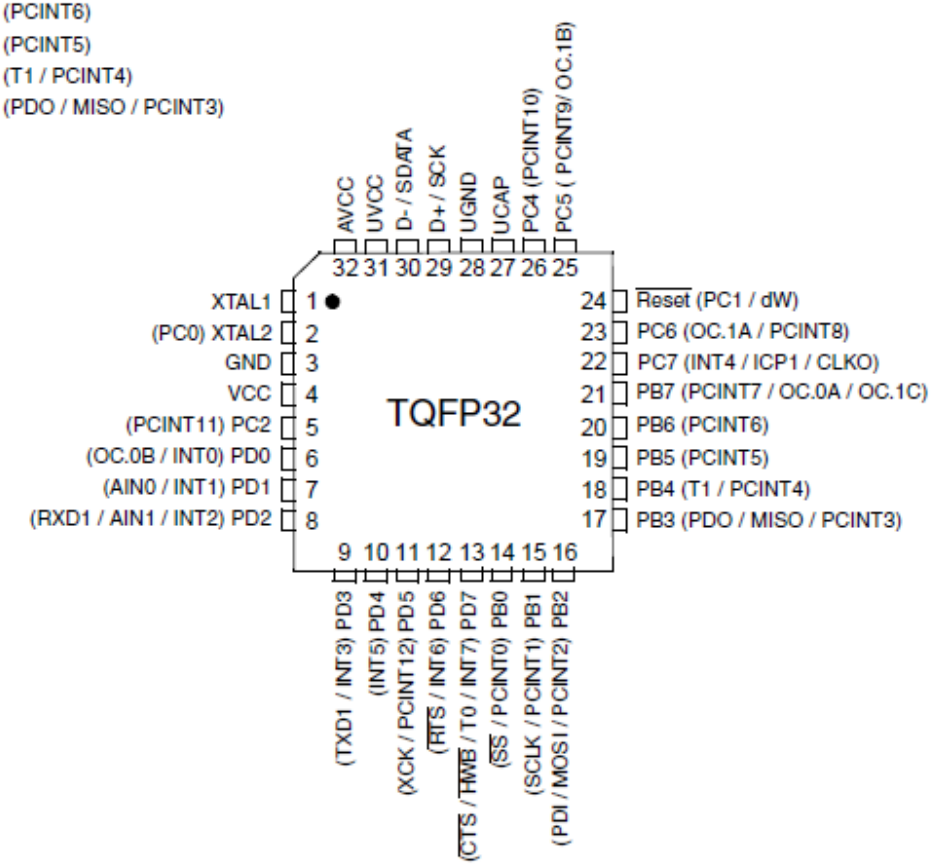


Izgled razvojne ploče

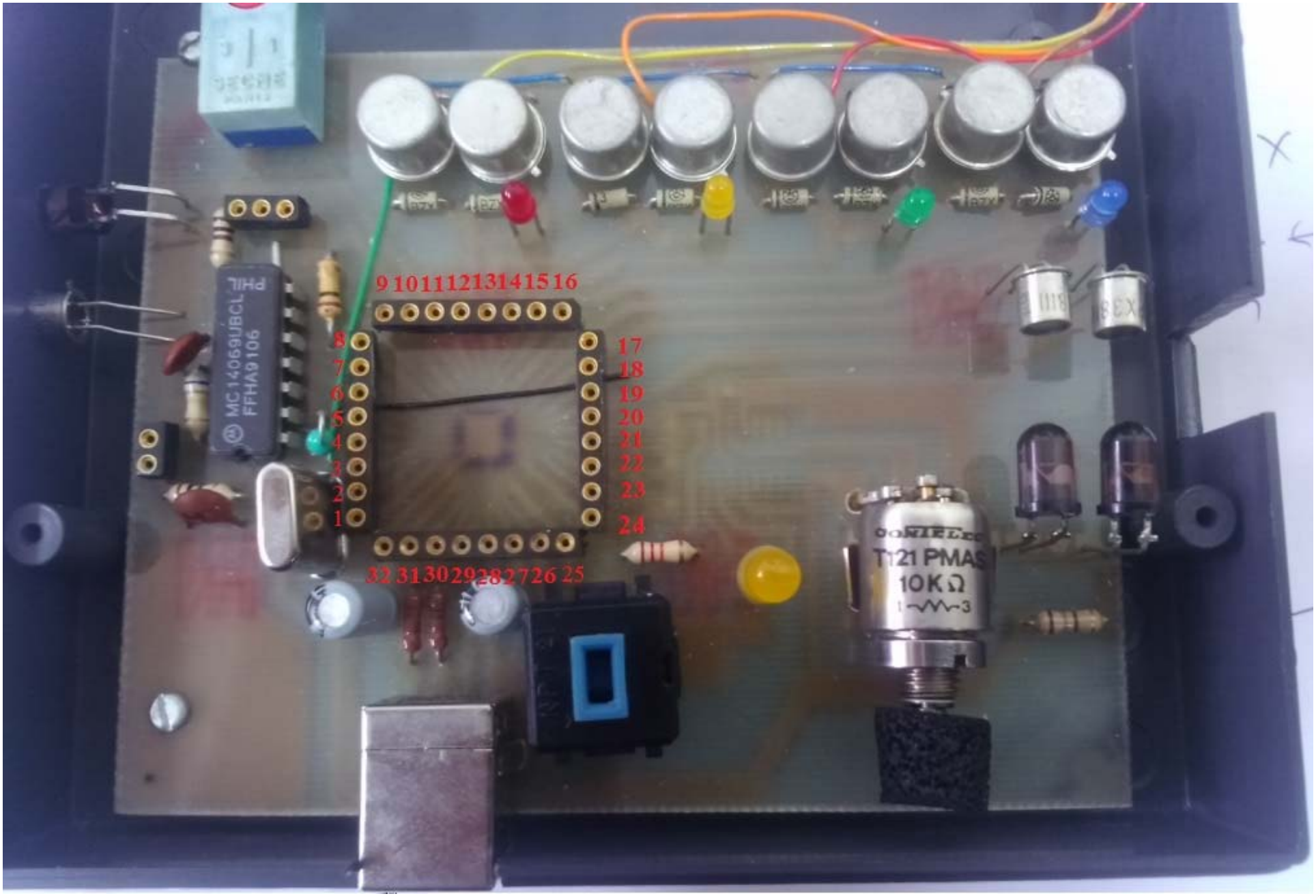


Naša razvojna ploča –Mikrokontroler AT90USB162





Raspored pinova uC na realizovanoj ploči



Naša razvojna ploča

- U Arduino razvojnom okruženju kompatibilna sa razvojnom pločom

  MattairTech MT-DB-U1 (AT90USB162)

- Uputsvo za instalaciju u Arduino razvojno okruženje, može se naći na adresi:

  https://github.com/mattairtech/ArduinoCore-avr#mt-db-u1mt-db-u2-at90usb162atmega32u2

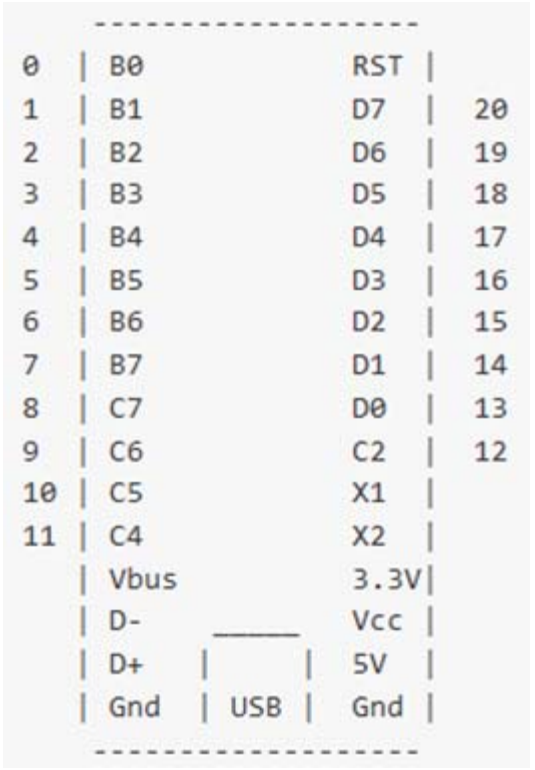
Portovi mikrokontrolera i pinovi u Arduino razvojnom okruženju

MT-DB-U1/MT-DB-U2 (AT90USB162/ATmega32U2)

```
=============== MattairTech MT-DB-U1/MT-DB-U2 (AT90USB162/ATmega32U2) ===================
 Comm    Interrupt   PWM   Digital                  Digital  Interrupt    PWM  Comm/other
=========================================================================================
                              -------------------
SPI SS                      0 | B0           RST |
SPI SCLK                    1 | B1           D7  |  20   20 (INT7)              JUMPER
SPI MOSI                    2 | B2           D6  |  19   19 (INT6)
SPI MISO                    3 | B3           D5  |  18
                            4 | B4           D4  |  17   17 (INT5)
                            5 | B5           D3  |  16   16 (INT3)           USART1 TX
                            6 | B6           D2  |  15   15 (INT2)           USART1 RX
                7 (TC1C)    7 | B7           D1  |  14   14 (INT1)
        8 (INT4)            8 | C7           D0  |  13   13 (INT0)   13 (TC0B)   LED
                9 (TC1A)    9 | C6           C2  |  12
                10 (TC1B)  10 | C5           X1  |
                           11 | C4           X2  |
                              | Vbus         3.3V|
                              | D-    _____  Vcc |
                              | D+   |     | 5V  |
                              | Gnd  | USB | Gnd |
                              -------------------
```

Microcontrolerski portovi i pinovi

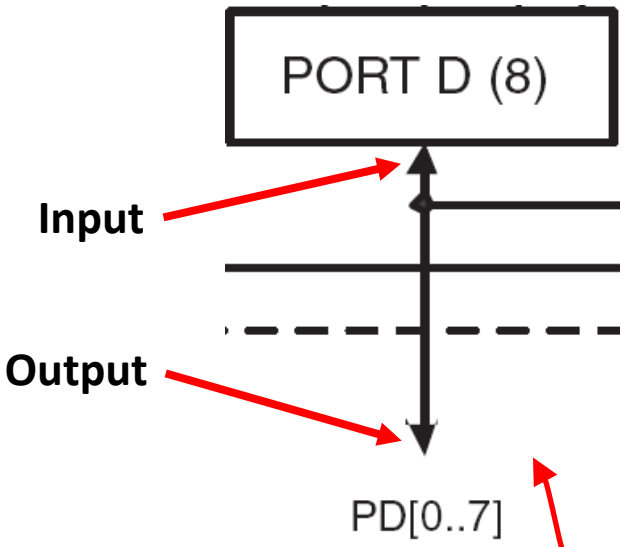
- Priključci kroz koje mikrokontroler opšti sa spoljašnjom sredinom
  - Pr. PORTB
    - Pinovi PB0 – PB7
      - Ne moraju biti susjedni
      - Često bi-*direcioni*

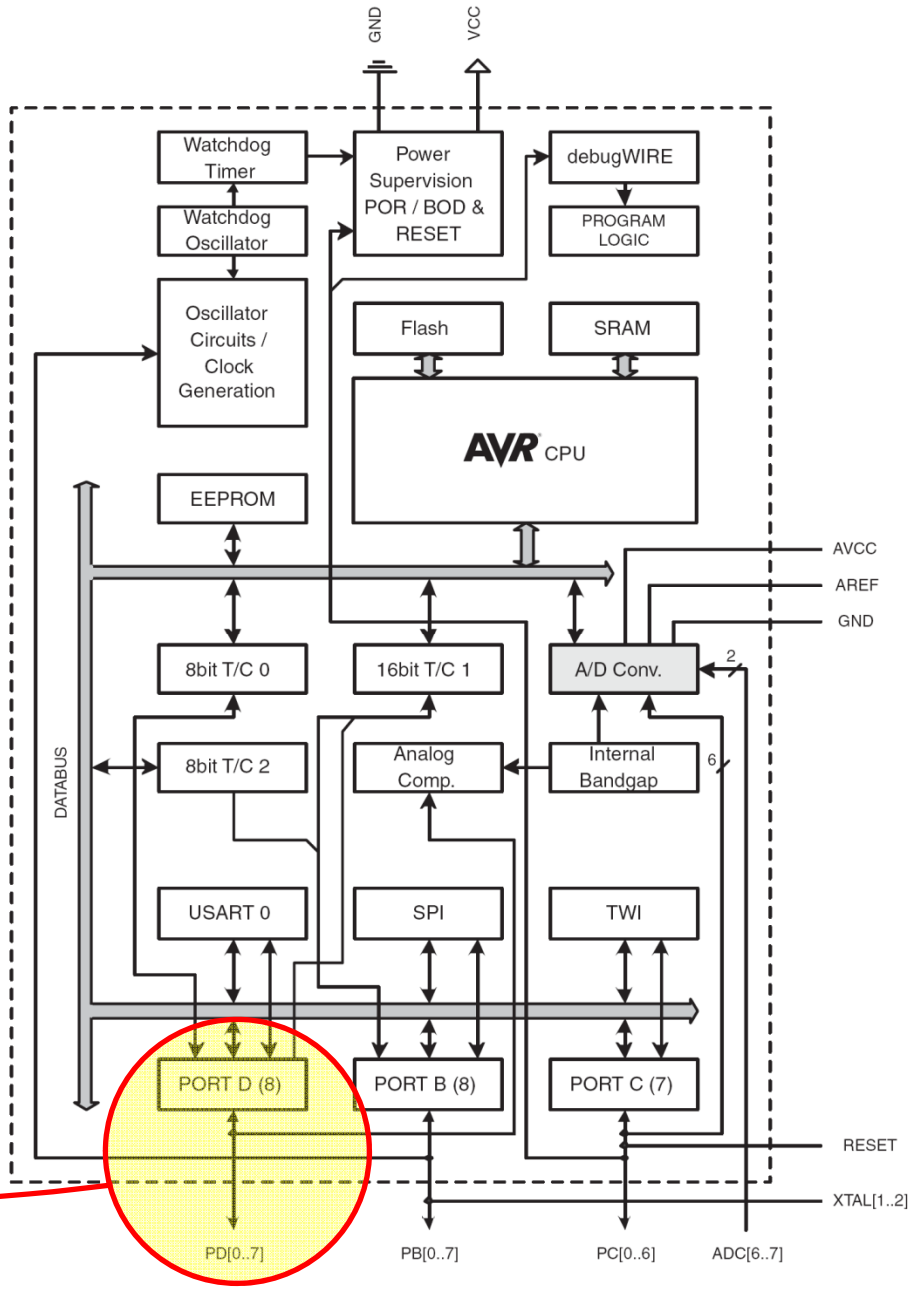


Port Pin – Usmjerenje podataka

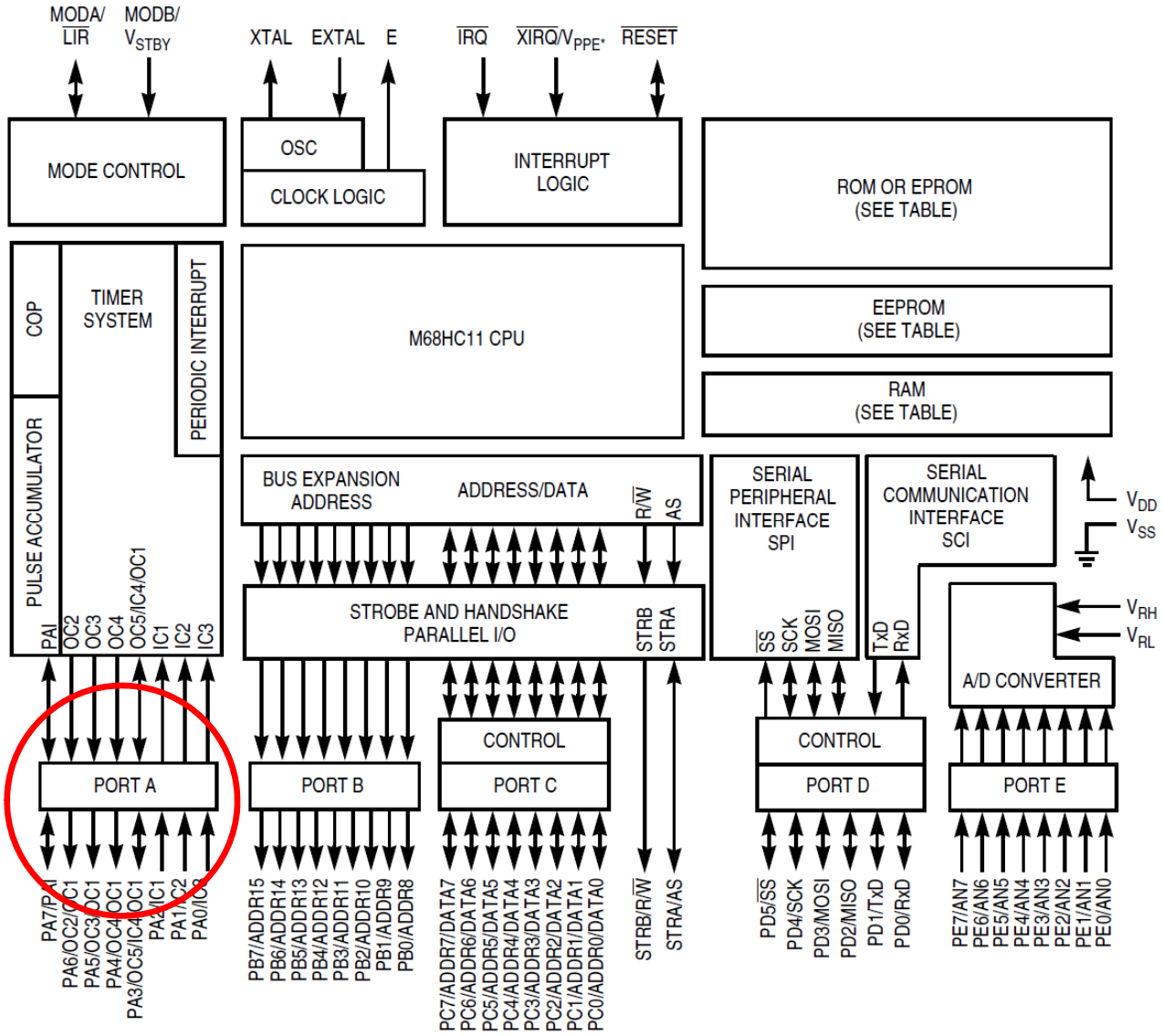
- Ulaz
  - Kada se želi uzeti informacija iz spoljašnjeg svijeta (senzori) **u** MCU
- Izlaz
  - Kada se želi izmijeniti stanje nečega **izvan** MCU (uključiti ili isključiti motor, itd.)
- Po uključenju napajanja svi pinovi su ulazni.
- Program može mijenjati usmjerenja podataka za svaki pin u svakom trenutku.

ATmega328
Blok diagram



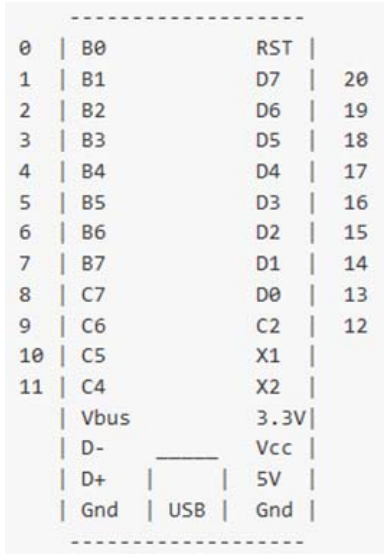


M68HC11 microkontroler



Postavljenje smjera toka podatka za pin

- Arduino
  - pinMode(*pin_no.*, *dir*)
    - Pr. postaviti Arduino pin 3 (PB3) kao izlazni
      - **`pinMode(3, OUTPUT);`**
  - Napomena: jedan pin u jednom trenutku
    - Predpostavimo da se želi postaviti pinove 3, 5, i 7 (PB3, PB5, i PB7) kao izlazne?
    - Postoji li način da se oni postave istovremeno?
      - Da! Kako, slijedi kasnije…

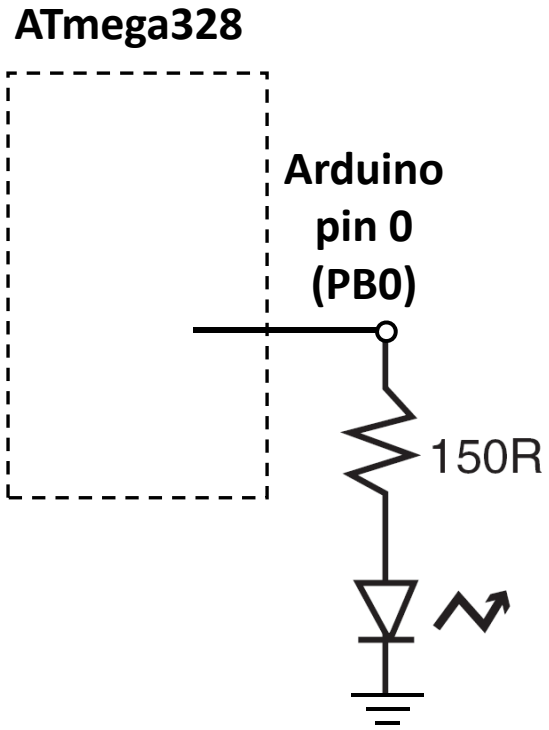


Napon na pinu

- Mikrokontroleri su u osnovi ***digitalni*** uređaji. Za digitalne ulazno/izlazne (IO) pinove:
  - Informacija je ‘kodirana’ u dva diskretna stanja:
    - HIGH or LOW (logic: 1 or 0)
    - Naponi
      - TTL
        - » 5 V (za HIGH)
        - » 0 V (za LOW)
      - 3.3 V CMOS
        - » 3.3 V (za HIGH)
        - » 0 V (za LOW)

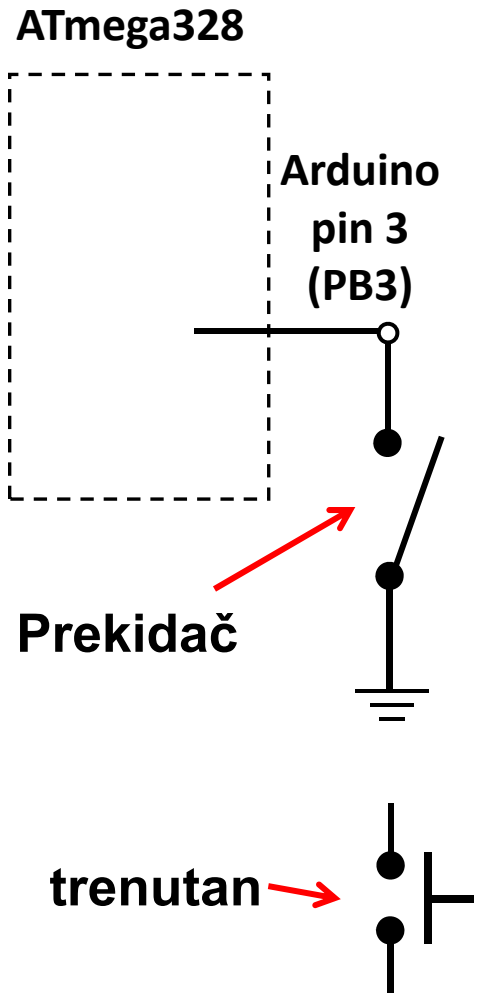
Pin upotrijebljen kao izlazni

- Uključiti LED, koja je povezana na Arduino pin 0 (PB0) (otpornik!)
  - Koji tok podataka treba biti za pin 0 (PB0)?
    - **`pinMode(____, ____);`**
  - Uključenje LED
    - **`digitalWrite(0,HIGH);`**
  - Isključenje LED
    - **`digitalWrite(0,LOW);`**



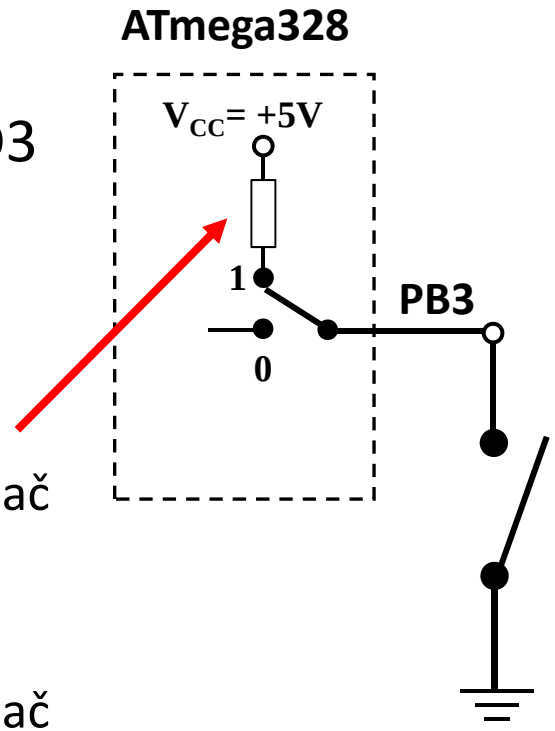
Pin kao ulazni + Pull-up otpornik

- Prekidač kao senzor
  - Pr. Senzor pojasa za sjedište u autu
  - Detekcija ***stanja prekidača***
    - Koji tok podataka treba biti za Arduino pin 3 (PB3)?
    - **`pinMode(____, ____);`**
    - Koji će biti napon na PB3 kada je prekidač zatvoren?
    - Koji će biti napon na PB3 kada je prekidač otvoren?
      - Neodređeno!



Pin kao ulazni + Pull-up otpornik

- Prekidač kao senzor, nastavak.
  - Učinimo napon na pinu poznatim uključenjem pull-up otpornika za PD3
    - Neka je PB3 ulazni port:
      - **`digitalWrite(3,HIGH);`**
        uključenje "pull-up" otpornika
      - **`pinMode(3,INPUT_PULLUP);`**
    - Koji će napon biti na PB3 kada je prekidač otvoren?
      - V_{CC}
    - Koji će napon biti na PB3 kada je prekidač zatvoren?

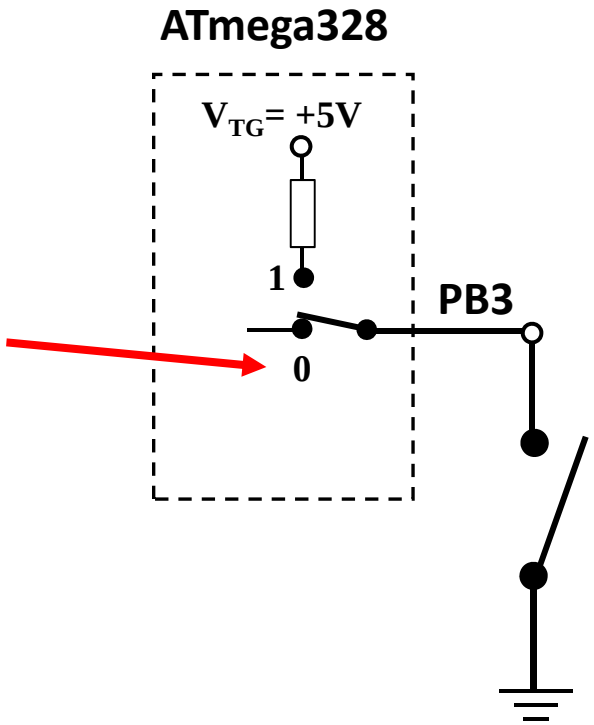


Pin kao ulazni + Pull-up otpornik

- Prekidač kao senzor, nastavak.
  - Za isključenje pull-up otpornika
    - Neka je PB3 ulazni port:

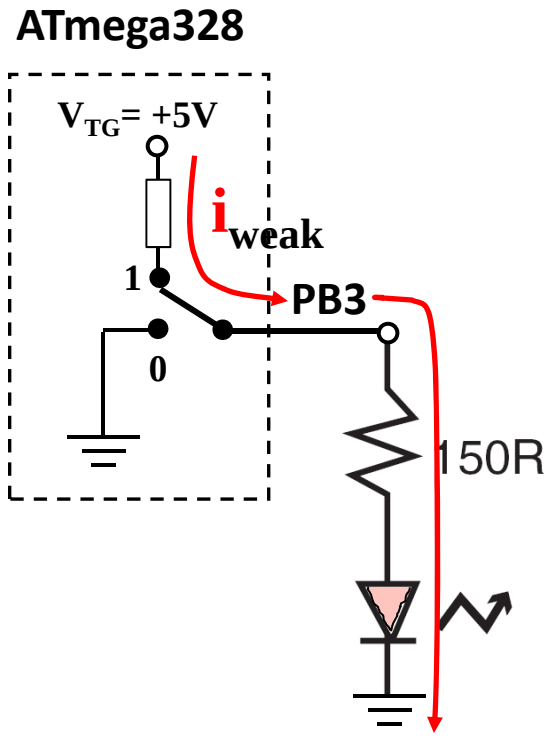
```
digitalWrite(3,LOW);
```

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.
    - Zapamtiti ovo!



Pritanje od prije?

- Pitanje od prije:
  - Postoji li način da se tok podataka postavi za više pinova istovremeno?
- Sav rad na MCU dešava se kroz *registre* (posebne memorijske lokacije)
  - Registri na AT90USB162 du dužine 8-bita
- Data direction register (DDRx) upravlja tokom podataka za pinove u PORTx

Bit	7	6	5	4	3	2	1	0	
0x04 (0x24)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Izvor:http://www.atmel.com/dyn/products/product_card.asp?PN=ATmega328P **p. 93**

Data Direction Register

- Ako je bit *nula* -> pin će biti *ulazni*
  - Postavljenje bit na nulu == '**čišćenje bita**' ('clearing the bit')
- Ako je bit *jedan* -> pin će biti *izlazni*
  - Postavljenje bit na jedinicu == '**postavljanje bita**' ('setting the bit')
- Za istovremenu promjenu toka podataka za više pinova koji pripadaju portu PORTx:
  1. Određivanje koje bitove treba postaviti a koje očistiti u registru DDRx.
  2. Upisati binarni (hex) broj u DDRx.

AT90USB162 registri za rad sa portovima

- Vidijeti AT90USB162 data sheet!
- Za digitalne IO, važni registi su:
  - DDRx
    - Data Direction bit u DDRx registru (read/write)
  - PORTx
    - PORTx data registar (read/write)
  - PINx
    - PINx registar (read only)

PORT Pin i registar detailji

AT90USB162 datasheet, pp. 76-94

Figure 13-1. I/O Pin Equivalent Schematic

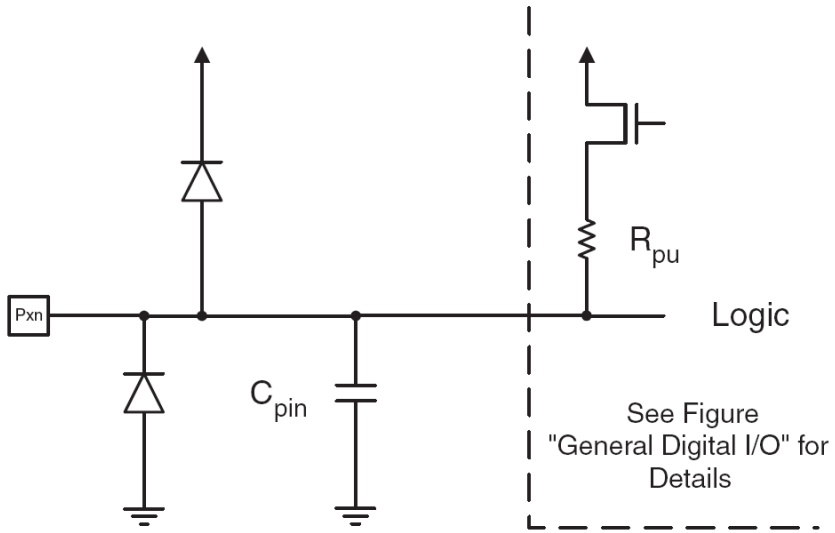
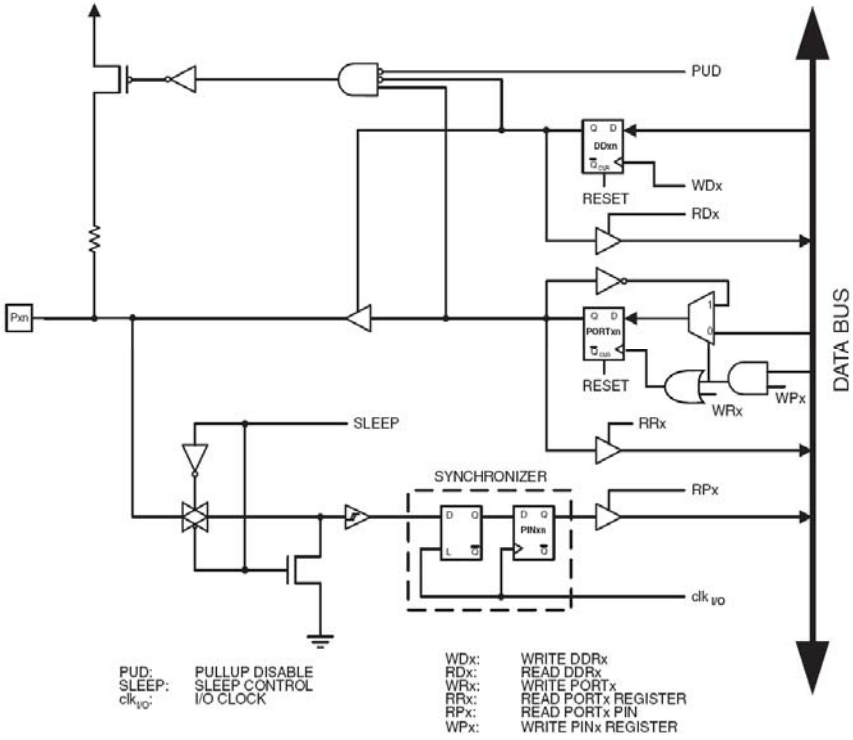


Figure 13-2. General Digital I/O[1]



PORTD – The Port D Data Register

Bit	7	6	5	4	3	2	1	0	
0x0B (0x2B)	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

DDRD – The Port D Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x0A (0x2A)	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

PIND – The Port D Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x09 (0x29)	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Primjer 1

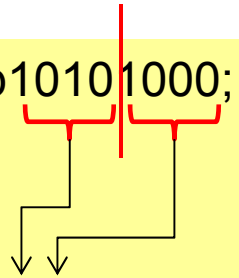
- Postaviti Arduino pinove 3, 5, i 7 (PB3, PB5, i PB7) kao izlazne

- Arduino pristup

```
pinMode(3, OUTPUT);
pinMode(5, OUTPUT);
pinMode(7, OUTPUT);
```

- Alternativni pristup

```
DDRB = 0b10101000;

ili

DDRB = 0xA8;

ili

DDRB | = 1<<PD7 | 1<<PD5 | 1<<PD3;
```



Primjer 2

- Postaviti Arduino pinove 0 i 1 (PB0 i PB1) kao ulazne, i uključiti pull-up otpornike

- Arduino pristup

```
pinMode(0, INPUT);
pinMode(1, INPUT);
digitalWrite(0, HIGH);
digitalWrite(1, HIGH);
```

- Alternativni pristup

```
DDRB = 0; // all PORTD pins inputs
PORTB = 0b00000011;
ili
PORTB = 0x03;

Ili još bolje:
DDRB & = ~(1<<PD1 | 1<<PD0);
PORTB | = (1<<PD1 | 1<<PD0);
```

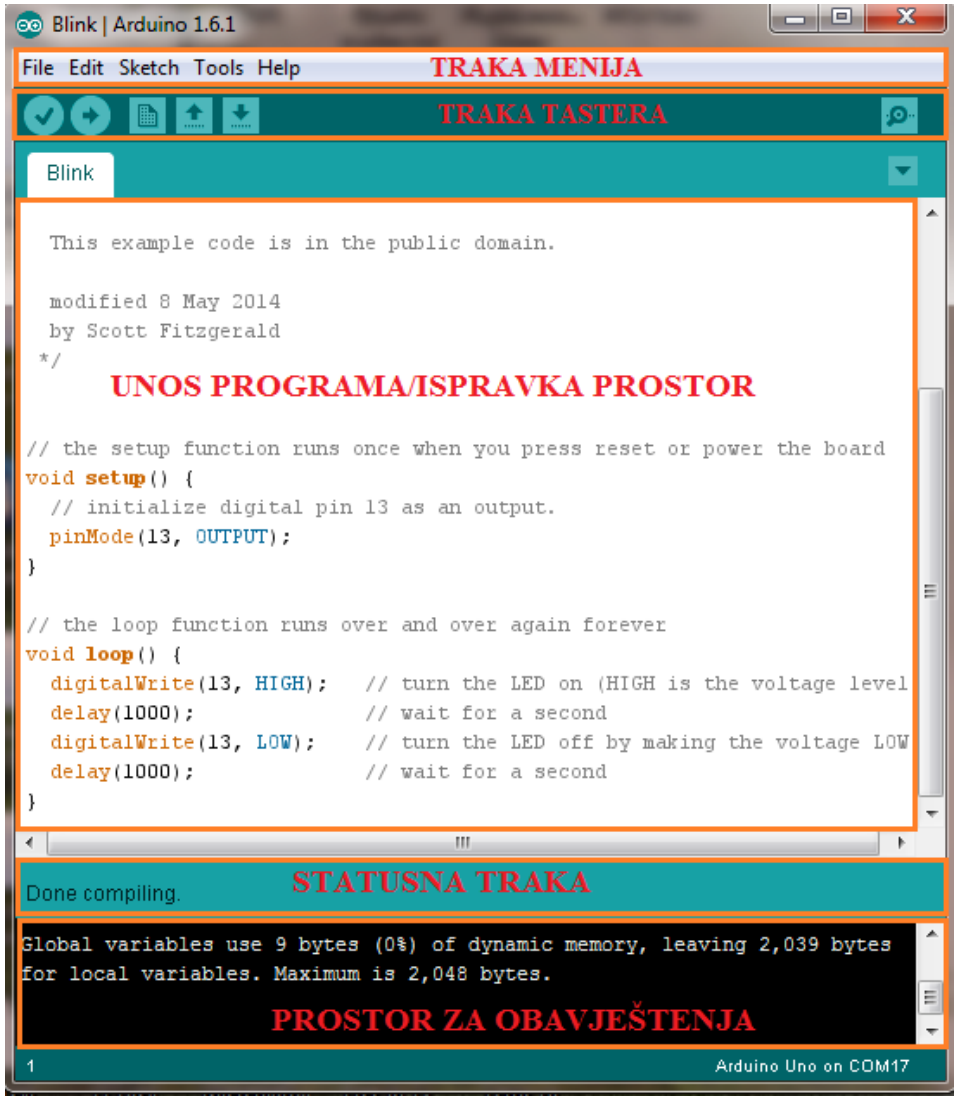
Kako startovati?

- Posjetite: http://arduino.cc/en/Guide/HomePage
  1. Preuzmite & instalirajte Arduino environment (IDE)
  2. Povezite ploču sa računarom pomoću USB kabla
  3. Ako je potrebno, instalirajte dodatne drajvere
  4. Pokrenite Arduino IDE
  5. Selektujte razvojnu ploču
  6. Selektujte serijski port
  7. Otvorite blink primjer
  8. Upišite program u razvojnu ploču

  ...

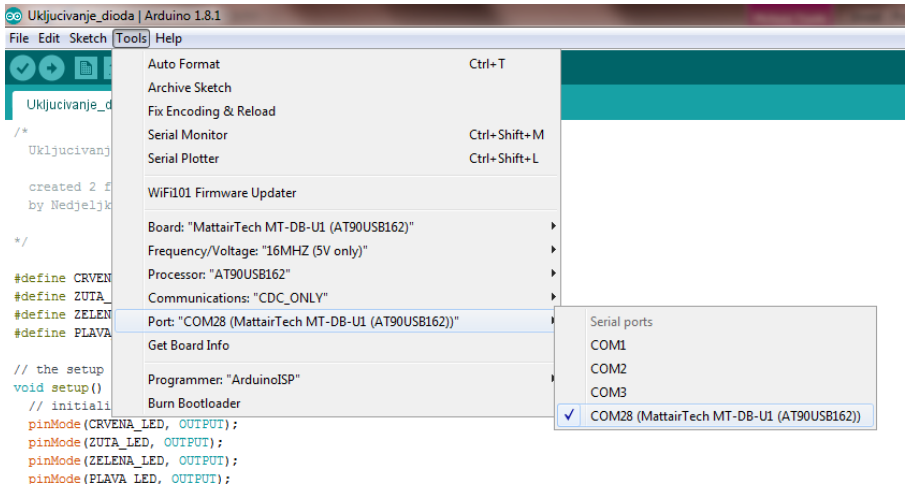
  9. Pisanje vlastitog programa
  10. Nerviranje/Debagiranje/Primoravanje da radi
  11. Oduševljenje i neposredno započinjanje novog projekta
  12. (spavanje je za slabiće)

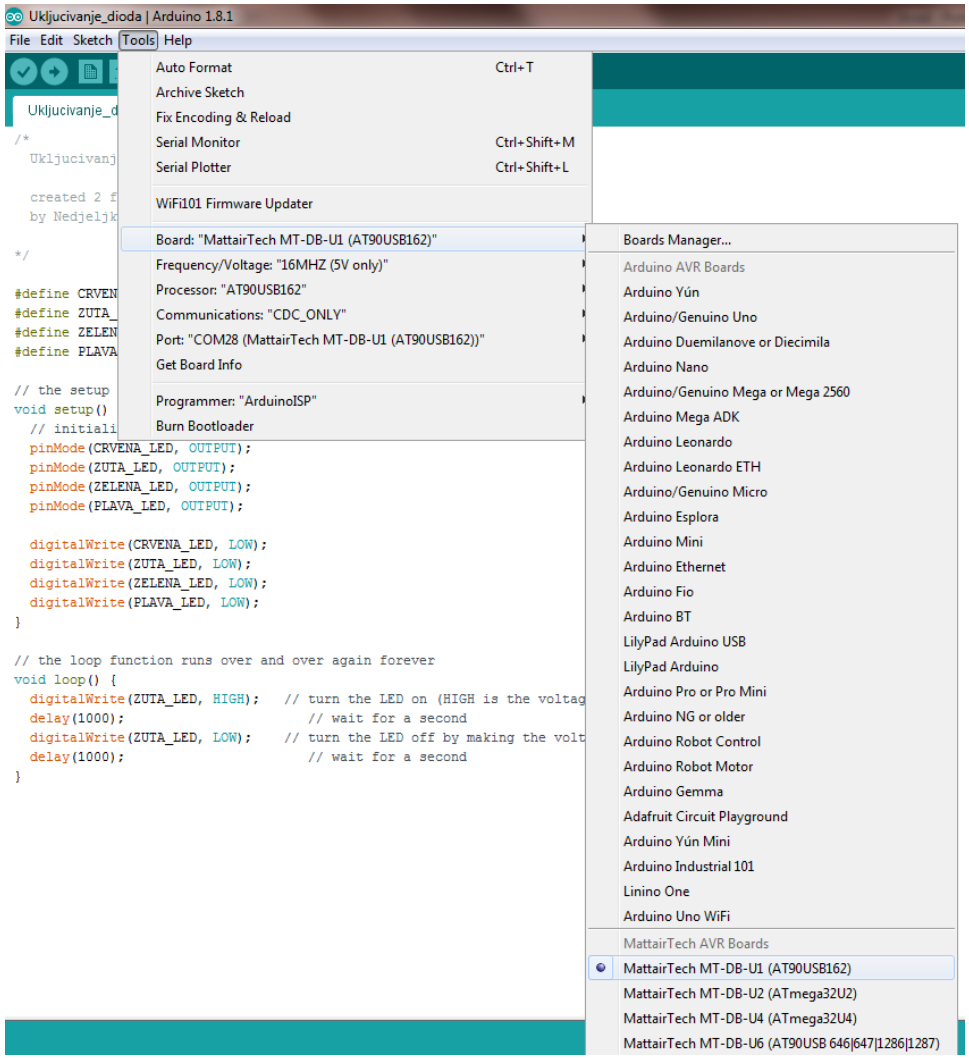
Arduino IDE



Pogledajte: http://arduino.cc/en/Guide/Environment za više informacija

Odaberite serijski port i ploču





Razvoj Arduino programa

- Zasnovan na C++ bez 80% komandi.
- Pregršt novih komandi.
- Programi se nazivaju 'sketches' (skečevi, skice) .
- Skečevi obevezno sadrže dvije funkcije:
  - void setup( )
  - void loop( )
- setup( ) se pokreće prvi i samo jedanput.
- loop( ) se pokreće neprestano, dok se ne isključi napajanje ili se ne učita novi skeč.

Arduino C

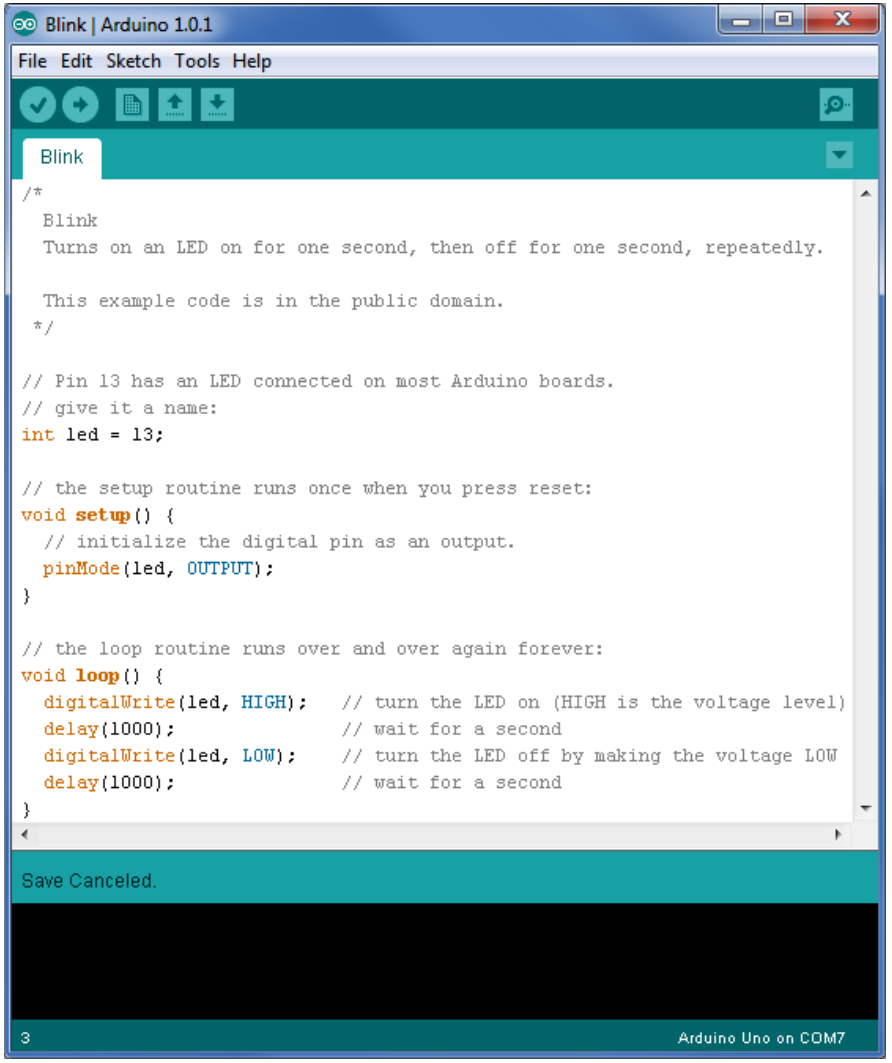
- Arduino skečevi uglavnom upravljaju pinovima na arduino ploči.
- Arduino skečevi su uvijek petlja.
  - void loop( ) {} je isto što i while(1) { }

Arduino tajming

- `delay(`*`ms`*`)`
  - Pauza nekoliko millisekundi
- `delayMicroseconds(`*`us`*`)`
  - Pauza nekoliko mikrosekundi
- Više komandi: arduino.cc/en/Reference/HomePage

Osobine kompajlera

- Brojni jednostavni skečevi su uključeni u kompajler
- Nalaze se pod opcijom File, Examples
- Kada je skeč napisan, može se upisati u programsku memoriju mikrokontrolera na Arduino štampanoj ploči kroz opcije File, Upload, ili pritiskom na <Ctrl> U



Arduino C je izveden iz C++

- Ovaj program radi treperenje LED na pinu 19

- avr-libc

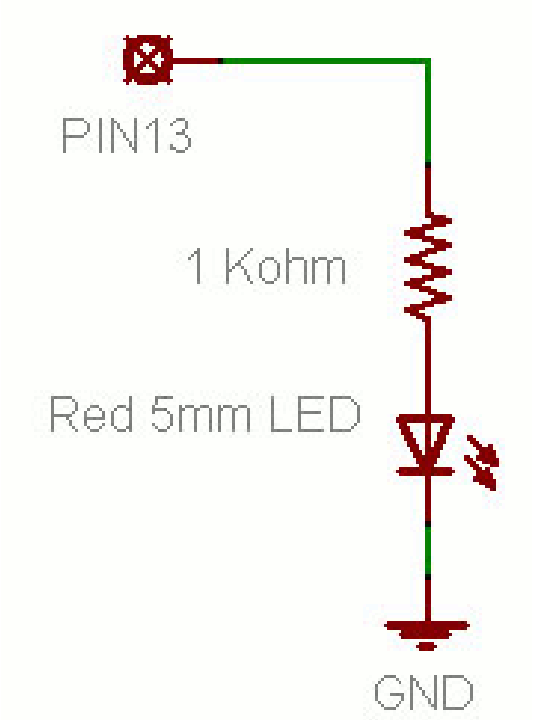
```
#include <avr/io.h>
#include <util/delay.h>

int main(void) {
    while (1) {
      PORTD = 0x40;
      _delay_ms(1000);
      PORTD = 0x00;
      _delay_ms(1000);
    }
    return 1;
}
```

- Arduino C

```
void setup( ) {
    pinMode(19, OUTPUT);
}

void loop( ) {
    digitalWrite(19, HIGH);
    delay(1000);
    digitalWrite(19, LOW);
    delay(1000);
}
```

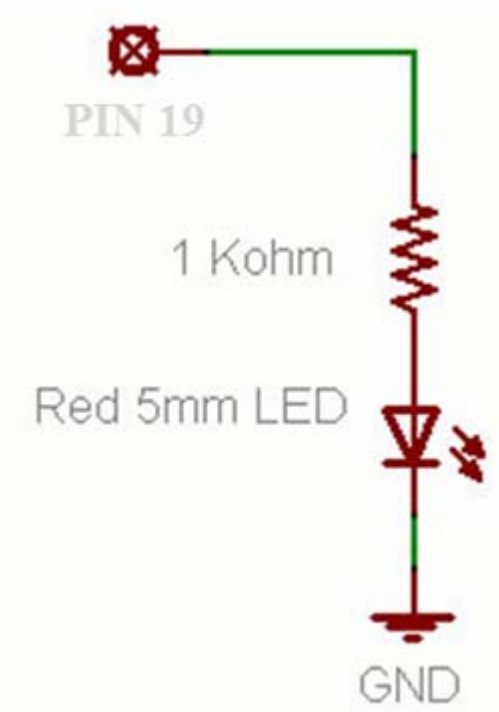
Prosto elektronsko kolo

- Najjednostavniji sklop.
- Uključi/isključi svjetlo.
- Struja teče iz pina (izvora napajanja), kroz potrošač (LED).



Osnovno LED kolo

- Pin 19 mikrokonrolera je preko otpornika spojen na CRVENU LED.
- Drugu nožicu otpornika spojite na dužu nožicu LED.
  - Veća otpornost znači slabije svjetlo.
  - Manja otpornost znači jače svjetlo.
  - Bez otpornosti znači pregorijevanje LED ili port.
- Kraću nožicu LED spojite na negativni priključak napajanja (masu).



Blink Skeč (Treperenje)

```
void setup( ) {
  pinMode(19, OUTPUT);
}
void loop( ) {
  digitalWrite(19, HIGH);
  delay(1000);
  digitalWrite(19, LOW);
  delay(1000);
}
```

Struktura Arduino programa

- Arduino program == '**sketch**'
  - Mora imati:
    - setup()
    - loop()
  - setup()
    - Konfiguriše pinove i registre
  - loop()
    - Pokreće glavno tijelo programa neprestano
      - Kao while(1) {…}
  - Gdje je main() ?
    - Arduino uprošćava stavri
    - Odrađuje za Vas

```
/* Blink  -  turns on an LED for DELAY_ON msec,
then off for DELAY_OFF msec, and repeats
*/

#define LED_PIN  19    // LED on digital pin 19
#define DELAY_ON  1000
#define DELAY_OFF 1000

void setup()
{
  // initialize the digital pin as an output:
  pinMode(LED_PIN, OUTPUT);
}

// loop() method runs forever,
// as long as the Arduino has power

void loop()
{
  digitalWrite(LED_PIN, HIGH);    // set the LED on
  delay(DELAY_ON);  // wait for DELAY_ON msec
  digitalWrite(LED_PIN, LOW);   // set the LED off
  delay(DELAY_OFF);  // wait for DELAY_OFF msec
}
```

Treperenje 4 LED skeč

```
void setup( ) {
  pinMode(19, OUTPUT);
  pinMode(20, OUTPUT);
  pinMode(0, OUTPUT);
  pinMode(1, OUTPUT);
}
```

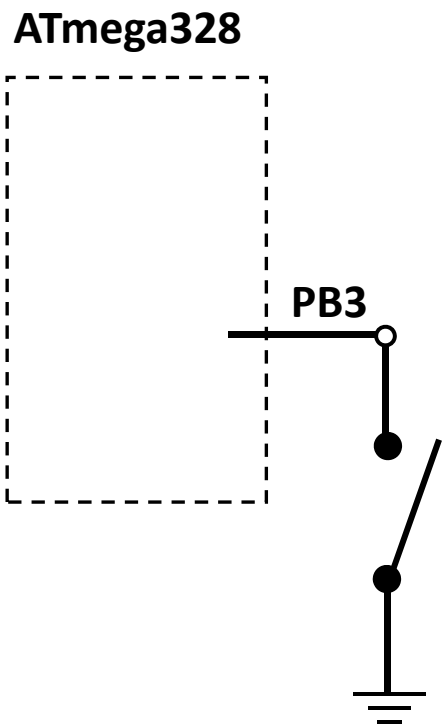
```
void loop( ) {
  digitalWrite(1, HIGH);
  delay (200);
  digitalWrite(1, LOW);

  digitalWrite(3, HIGH);
  delay (200);
  digitalWrite(3, LOW);

  digitalWrite(5, HIGH);
  delay (200);
  digitalWrite(5, LOW);

  digitalWrite(7, HIGH);
  delay (200);
  digitalWrite(7, LOW);
}
```

Rezime?

- Super. Treperi svjetlo. Ništa posebno.
- Obuhvatili smo samo izlazne postove za sada.
- Možemo li upotrijebiti ulaze za detekciju fizičkih pojava?

Ulazni digitalni pin – Primjer 1

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



Ulazni digitalni pin – Primjer 1

- ‘Očitavanja pina’
  - Pseudokod:

    Postaviti PB3 kao ulazni

    Uključiti PB3 pull-up otpornik

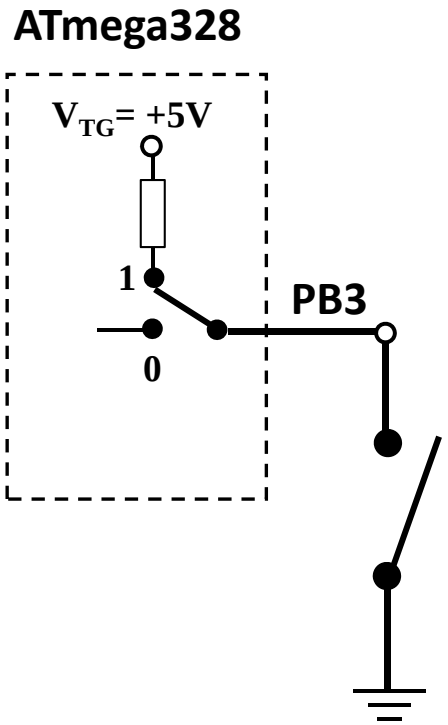
    Očitati napon sa Arduino pin 3 (PIN_B3)

    IF PIN_B3 napon je LOW (vezan), THEN

        pozovi funkciju start_enable()

    ELSE

        pozovi start_disable()



Ulazni digitalni pin – Primjer 1

- ‘Očitavanja pina’
  - Pseudokod:

    Postaviti PB3 kao ulazni

    Uključiti PB3 pull-up otpornik

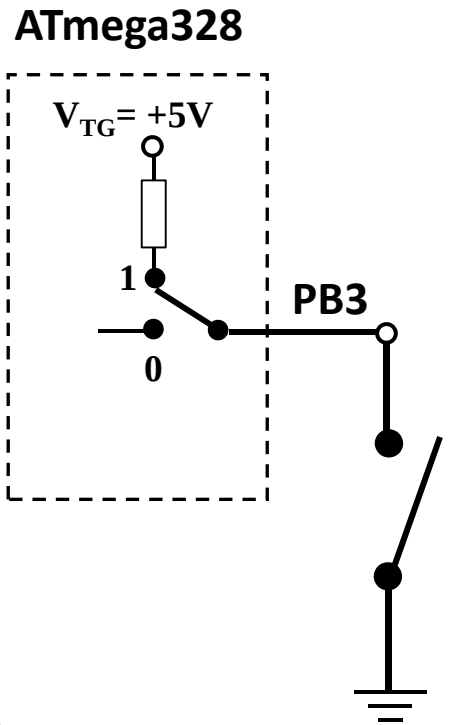
    Očitati napon sa Arduino pin 3 (PIN_B3)

    IF PIN_B3 napon je LOW (vezan), THEN

        pozovi funkciju start_enable()

    ELSE

        pozovi start_disable()

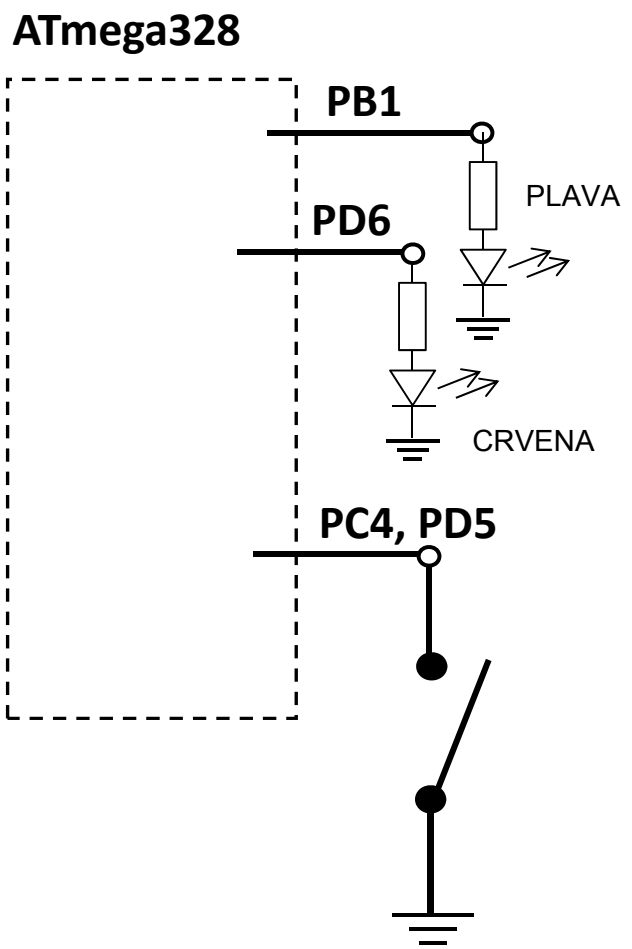


Fragment, nije cijeli program

```
#define PIN_OP_UN 3
#define LATCHED LOW
pinMode(PIN_ OP_UN,INPUT_PULLUP);
belt_state = digitalRead(PIN_ OP_UN);
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 CRVENE LED (PD6) i Plave LED (PB1) ako je ključ u bravi (PC4 zatvoren), ali pojas vozača nije vezan (PD5 otvoren)
  - Najprije pseudokod



Ulazni digitalni pin – Primjer 2

- Pseudokod:

Postavljanje toka podataka za pinove

Postaviti PC4 i PD5 kao ulaze

Uključiti pull-up otpornike za PC4 i PD5

Postaviti PB1 i PD6 kao izlaze

Beskonačna petlja

IF je ključ u bravi THEN

IF ako je pojas vezan, THEN

Isključi zvučni signal

Isključi LED

ELSE

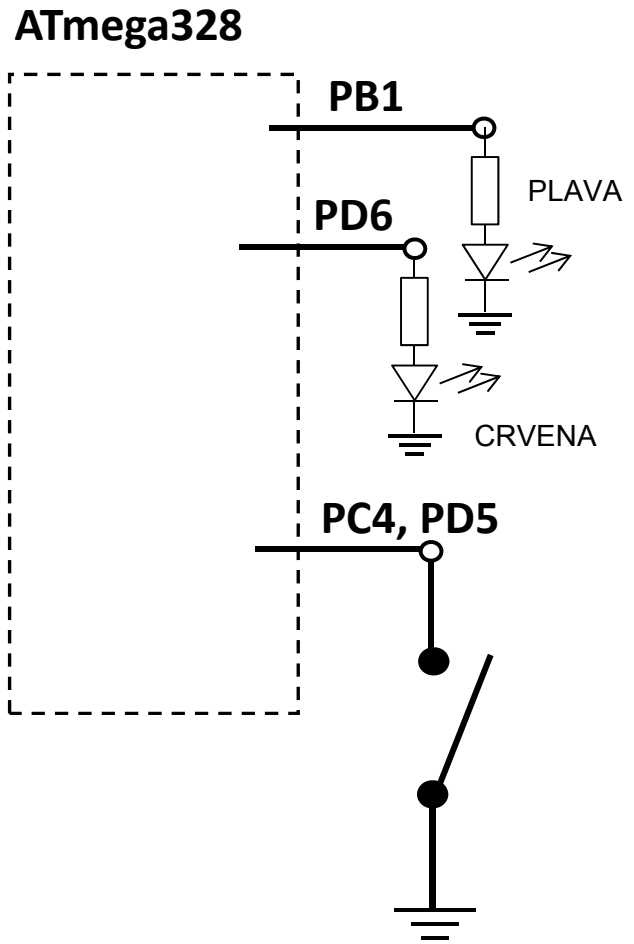
Uključi LED

Uključi zvučni signal

ELSE

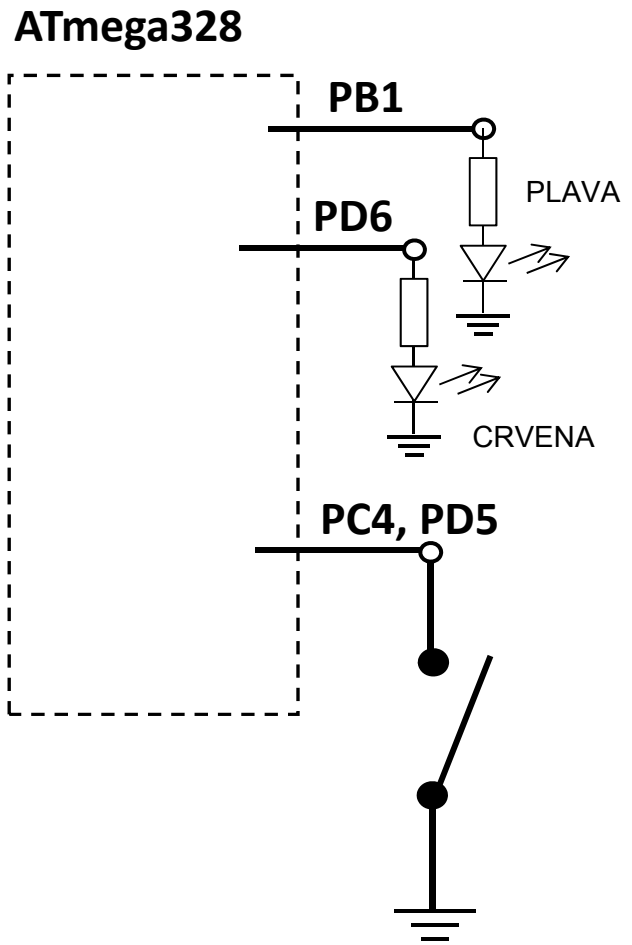
Isključi zvučni signal

Isključi LED



Ulazni digitalni pin – Primjer 2 (Arduino kod)

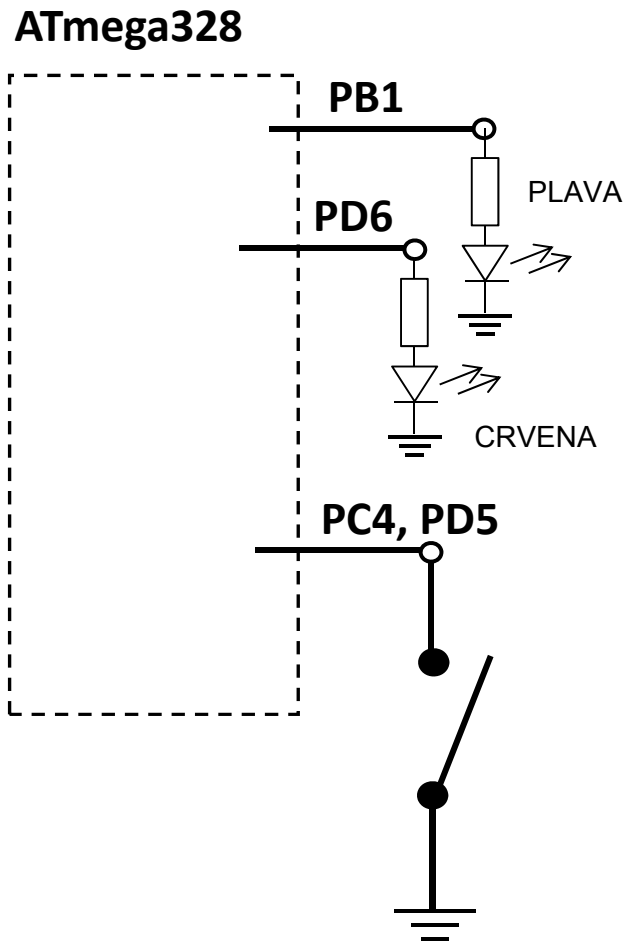
```
#define PIN_IGNITION  11
#define PIN_SEATBELT  18
#define PIN_RED_LED  19
#define PIN_BLUE_LED  1
#define SEATBELT_LATCHED  LOW
#define KEY_IN_IGNITION LOW
#define LED_ON  HIGH
#define LED_OFF  LOW
#define BUZZER_ON  HIGH
#define BUZZER_OFF  LOW

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



/* see next page for code */

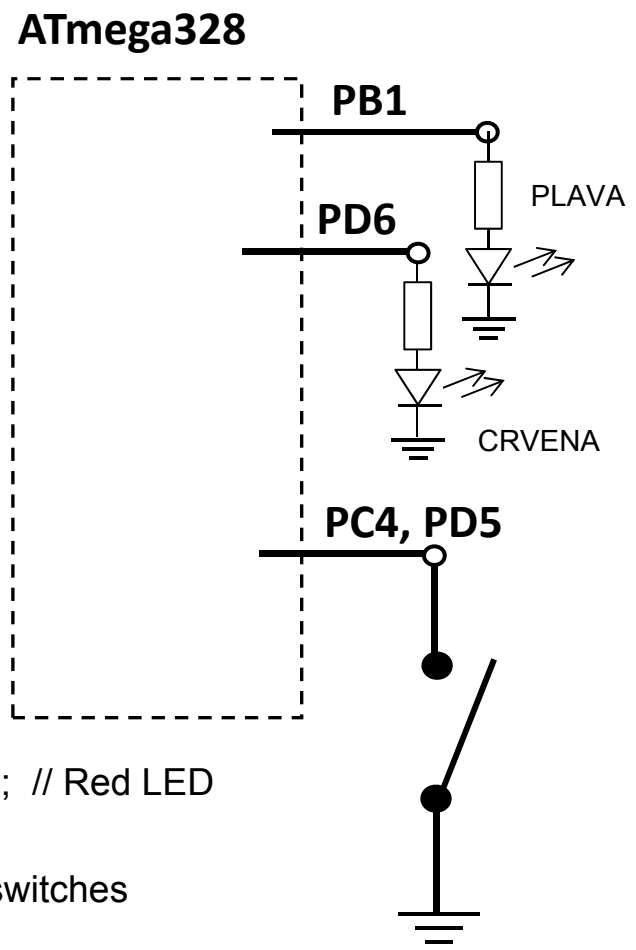
Ulazni digitalni pin – Primjer 2 (Arduino kod)

```
/* 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_BLUE_LED, BUZZER_OFF);
          digitalWrite(PIN_RED_LED, LED_OFF);
      }
      else  // key is in ignition, but seatbelt NOT latched
      {
           digitalWrite(PIN_BLUE_LED, BUZZER_ON);
           digitalWrite(PIN_RED_LED, LED_ON);
      }
  }
  else  // key is NOT in ignition
  {
        digitalWrite(PIN_BLUE_LED, BUZZER_OFF);
        digitalWrite(PIN_RED_LED, LED_OFF);
  }
}
```



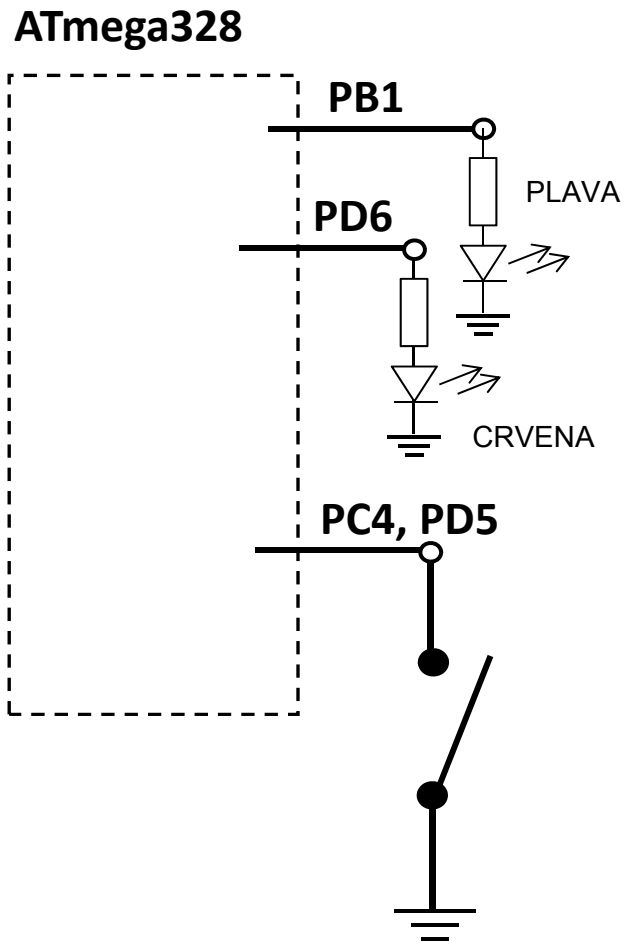
Ulazni digitalni pin – Primjer 2 (Alternativni kod)

```
/* NOTE: #defines use predefined PORT pin numbers  for ATmega328 */
#define PIN_IGNITION  PC4
#define PIN_SEATBELT  PD5
#define PIN_RED_LED  PD6
#define PIN_BLUE_LED  PB1
#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_RED_LED)  | _BIT_MASK(PIN_BLUE_LED);  // Red LED
    and Blue LED (BUZZER)
    PORTD | = _BV(PIN_IGNITION) | _BV(PIN_SEATBELT);  // pullups for switches
}

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



Ulazni digitalni pin – Primjer 2 (Arduino kod)

```
/* see previous page for setup() code */
void loop()
{
    uint8_t current_PORTD_state, key_state, belt_state;
    current_PORTC_state = PINC;  // snapshot of PORTC pins
    current_PORTD_state = PIND;  // snapshot of PORTD pins
    key_state = current_PORTC_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_RED_LED)  |  _BV(PIN_BLUE_LED) );
        }
        else
        {
          PORTD |=  ( _BV(PIN_RED_LED)  |  _BV(PIN_BLUE_LED) );
        }
    }
    else
    {
      PORTD  & = ~( _BV(PIN_RED_LED)  |  _BV(PIN_BLUE_LED) );
    }
}
```



Zaključak

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


Kraj