

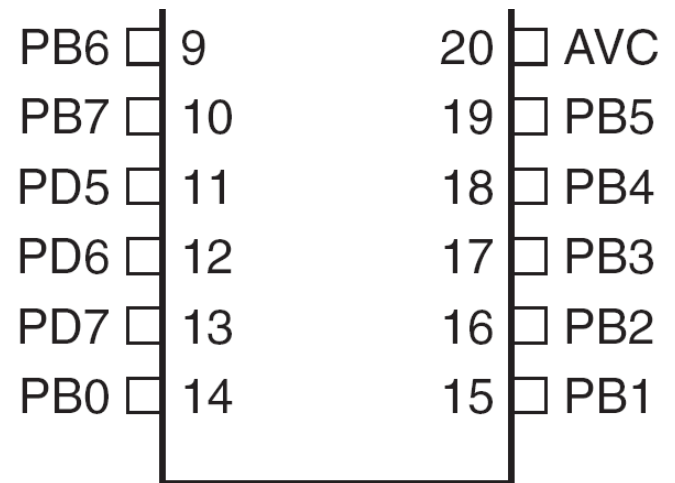
# PERIFERIJE I INTERFEJSI

**Portovi i pinovi**

**Izlazni portovi**

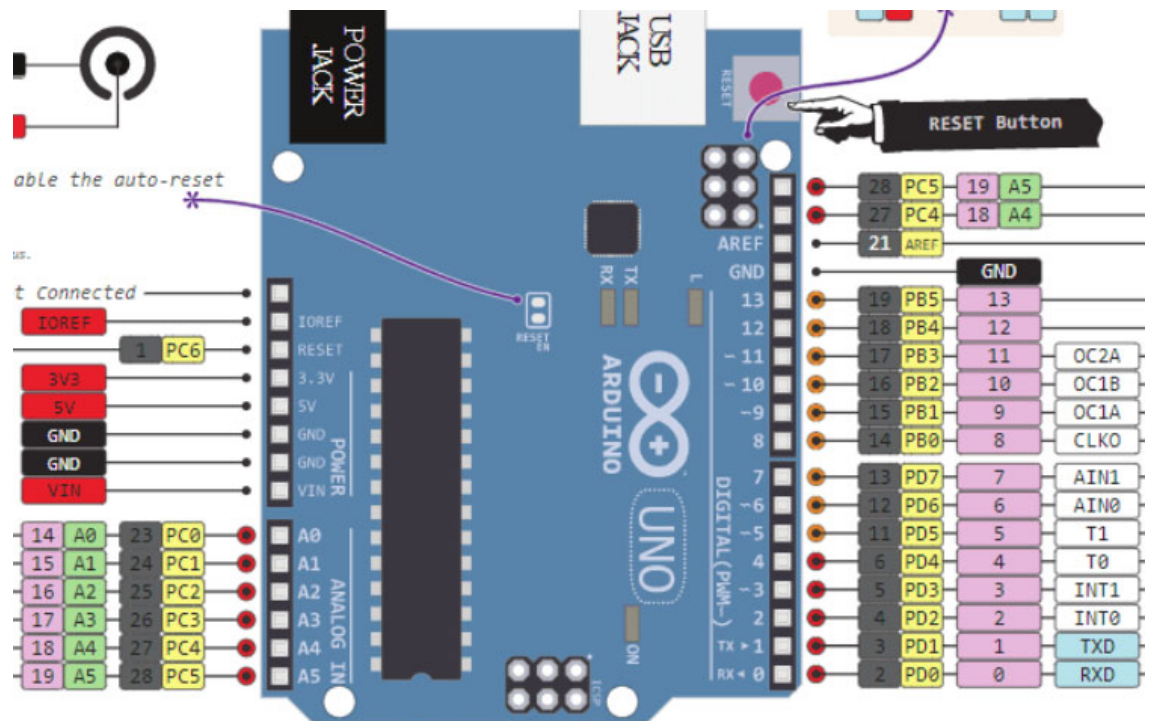
# Microcontrolerski portovi i pinovi

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



# Microcontrollerski portovi i pinovi

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 ( $\overline{SS}$ /OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

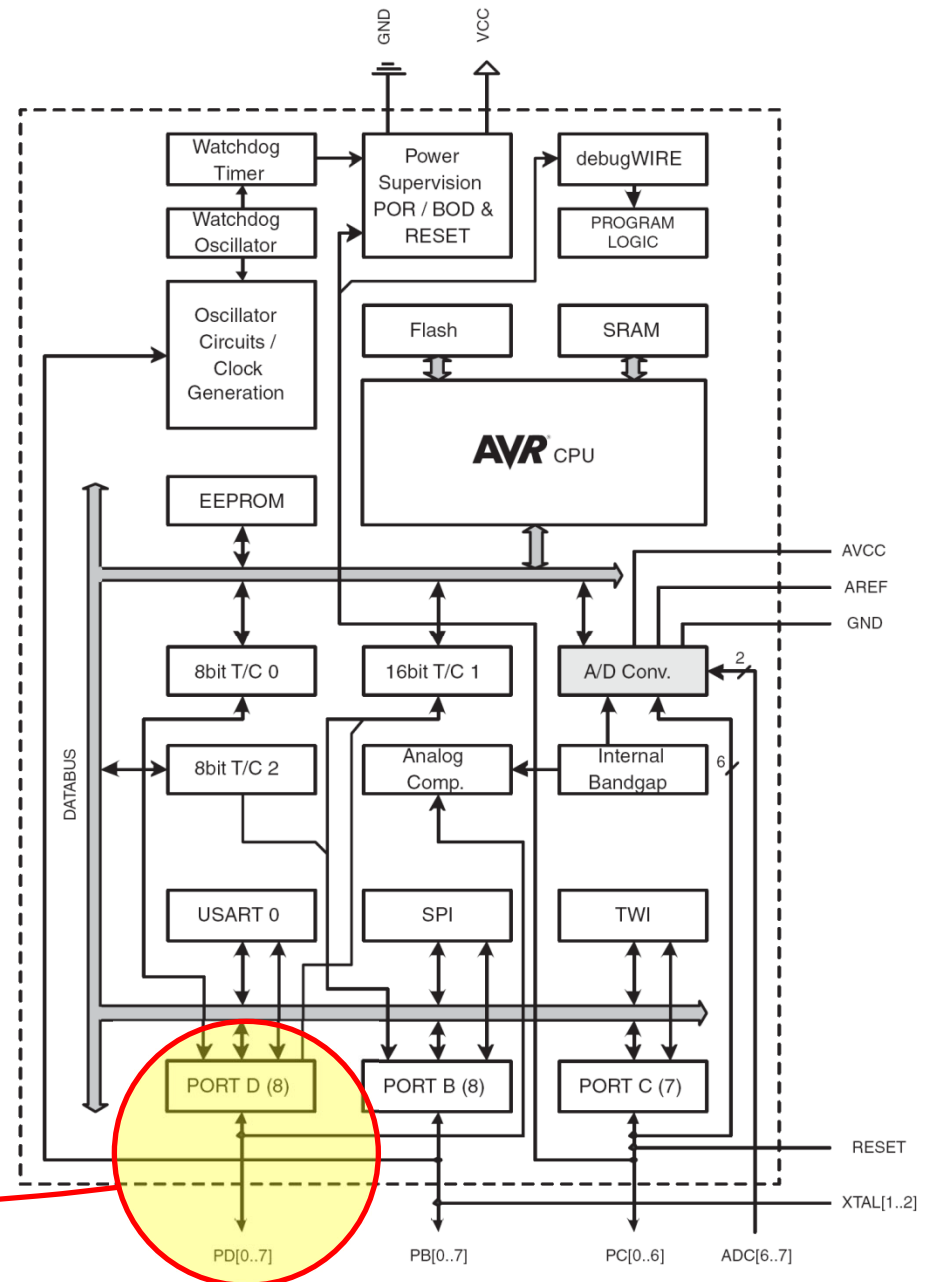
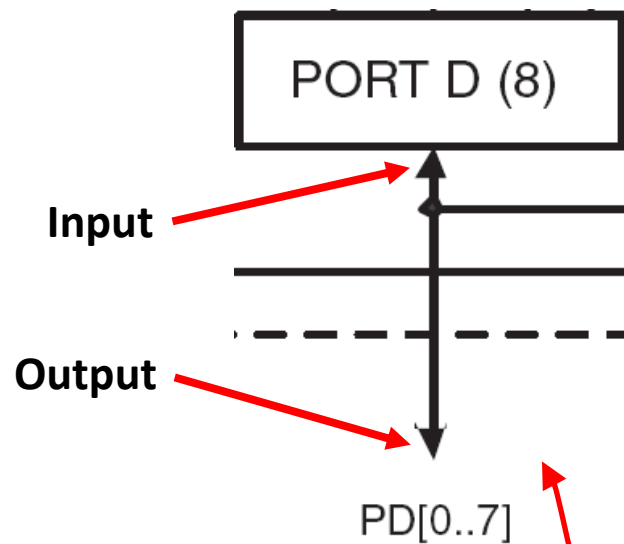


# Port Pin – Usmjerenje podataka

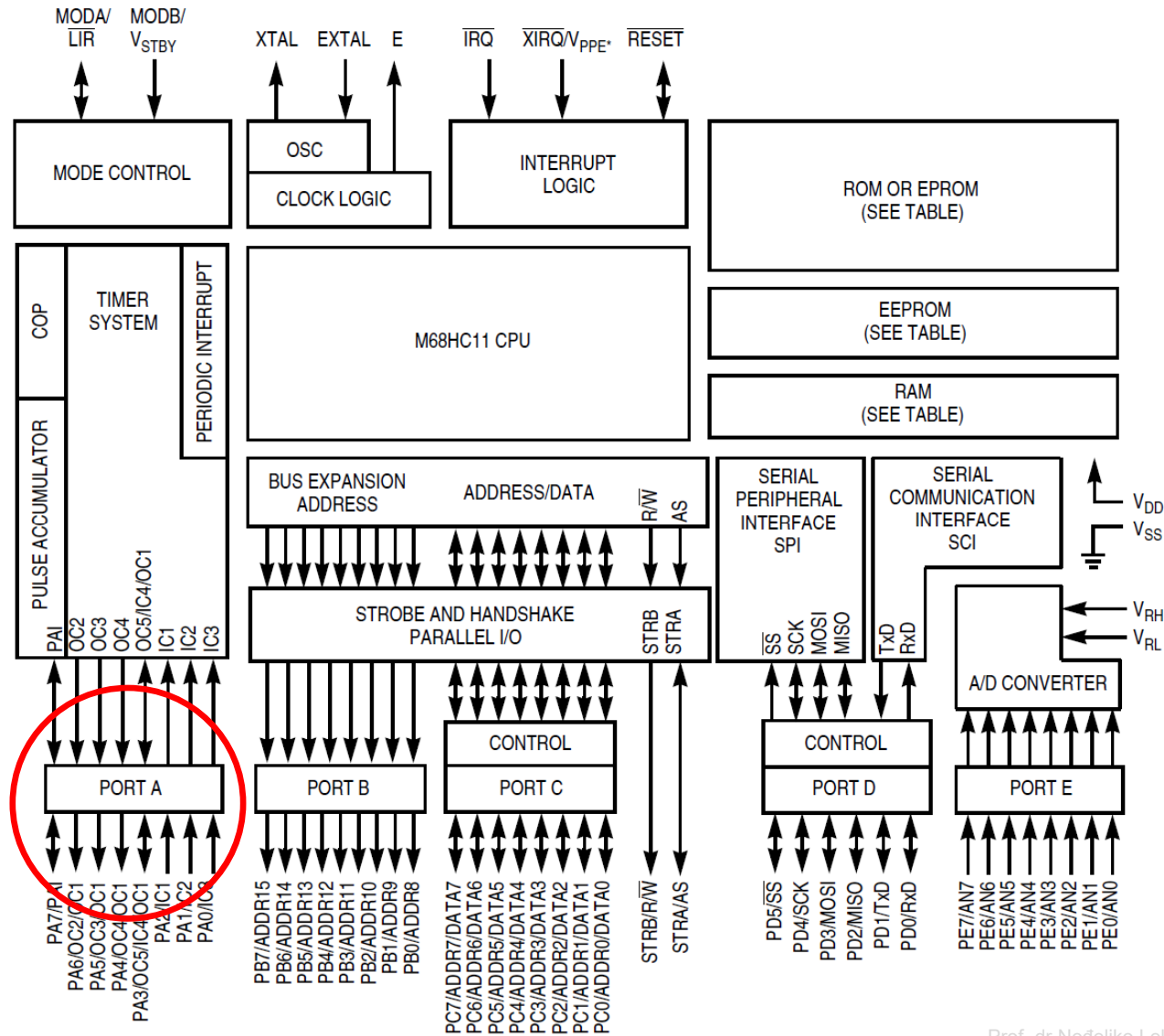
- Ulaz
  - Kada se želi uzeti informacija iz spoljašnjeg svijeta (senzori) **u** MCU
- Output
  - Kada se želi izmijeniti stanje nečega **izvan** MCU (uključiti ili isključiti motor, itd.) (aktuatori)
- Po uključanju napajanja svi pinovi su ulazni.
- Program može mijenjati usmjerenja podataka za svaki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

# ATmega328

## Blok diagram



# M68HC11 mikrokontroler



# Postavljenje smjera toka podatka za pin

- Arduino
  - `pinMode(pin_no., dir)`
    - Pr. postaviti Arduino pin 3 (PD3) kao izlazni
      - `pinMode(3, OUTPUT);`
  - Napomena: jedan pin u jednom trenutku
    - Predpostavimo da se želi postaviti pinove 3, 5, i 7 (PD3, PD5, i PD7) kao izlazne?
    - Postoji li način da se oni postave istovremeno?
      - Da! Kako, slijedi kasnije...

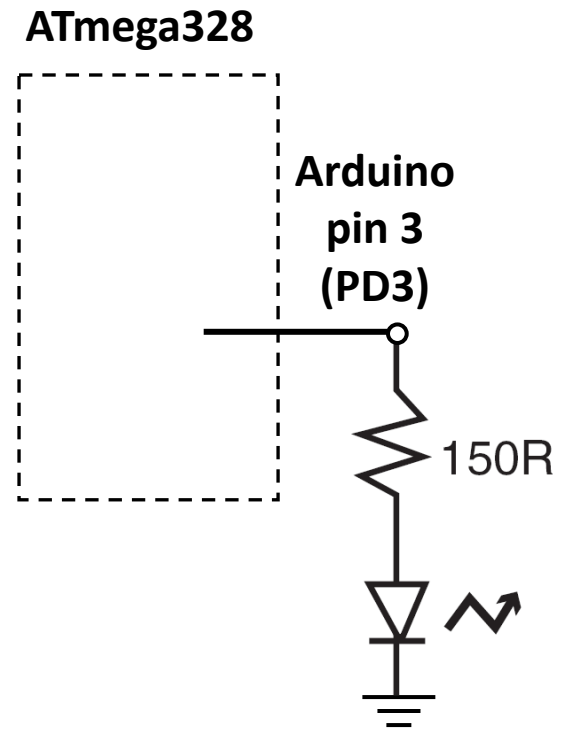
# Napon na pinu

- 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)



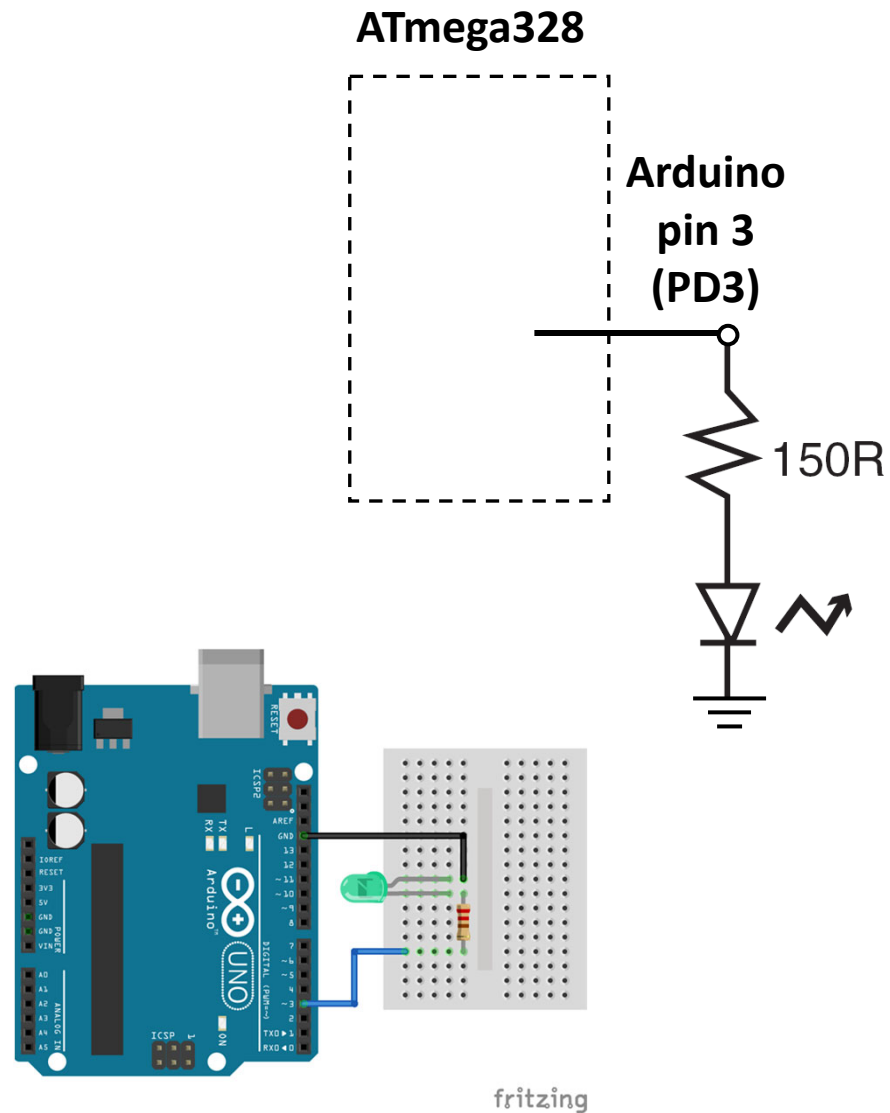
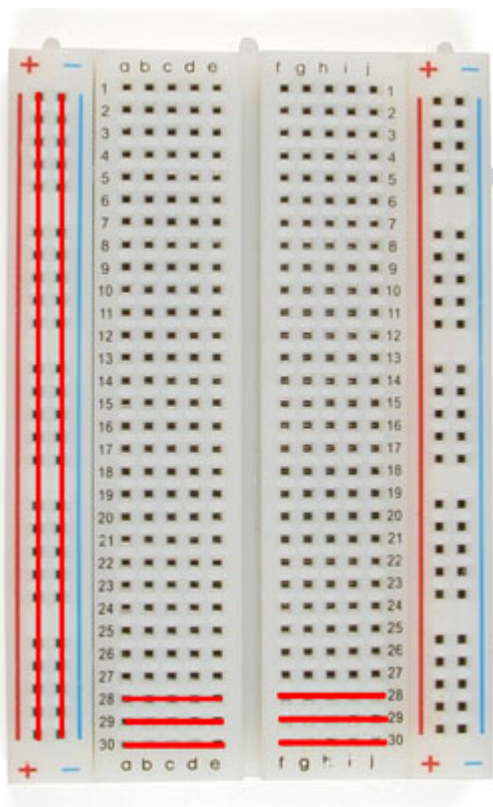
# Pin upotrijebljen kao izlazni

- Uključiti LED, koja je povezana na Arduino pin 3 (PD3) (otpornik!)
  - Koji tok podataka treba biti za pin 3 (PD3)?
    - `pinMode(____, ____);`
  - Uključenje LED
    - `digitalWrite(3,HIGH);`
  - Isključenje LED
    - `digitalWrite(3,LOW);`

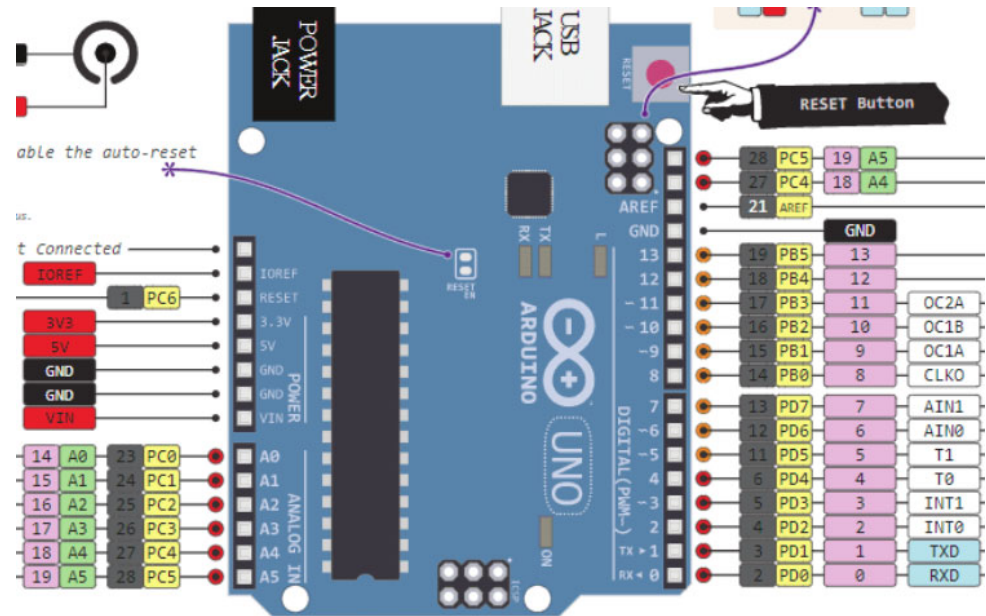


# Pin upotrijebljen kao izlazni

- Uključenje LED
  - `digitalWrite(3,HIGH);`
- Isključenje LED
  - `digitalWrite(3,LOW);`



# I što?



- 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 Atmega328 su 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)	<b>DDB7</b>	<b>DDB6</b>	<b>DDB5</b>	<b>DDB4</b>	<b>DDB3</b>	<b>DDB2</b>	<b>DDB1</b>	<b>DDB0</b>	<b>DDRB</b>
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](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 bita 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.

# ATmega328 registri za rad sa portovima

- Vidijeti ATmega328 data sheet, pp. 76-94
- Za digitalne IO, važni registri 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

ATmega328 datasheet, pp. 76-94

Figure 13-2. General Digital I/O<sup>(1)</sup>

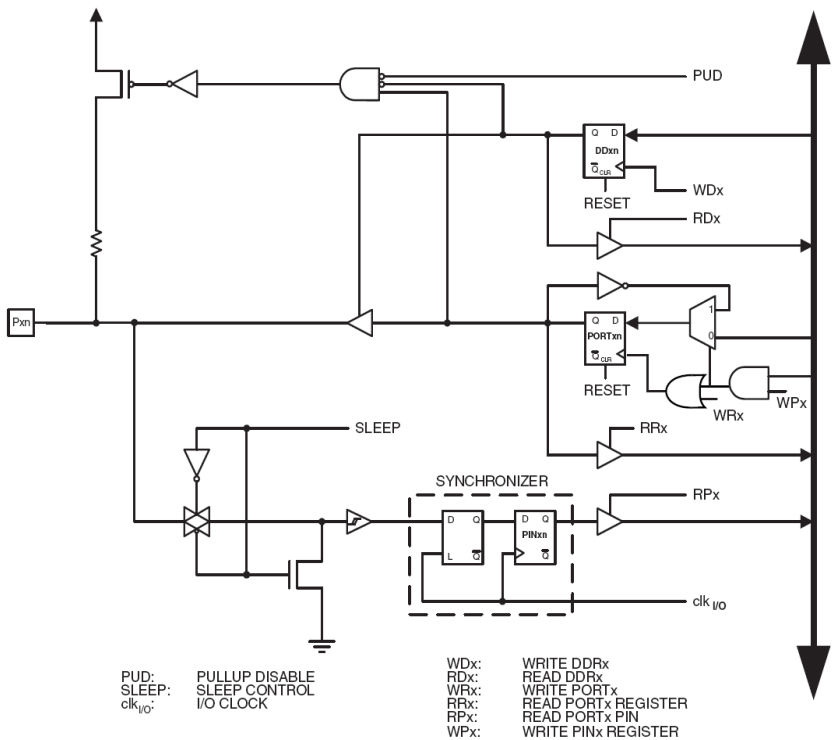
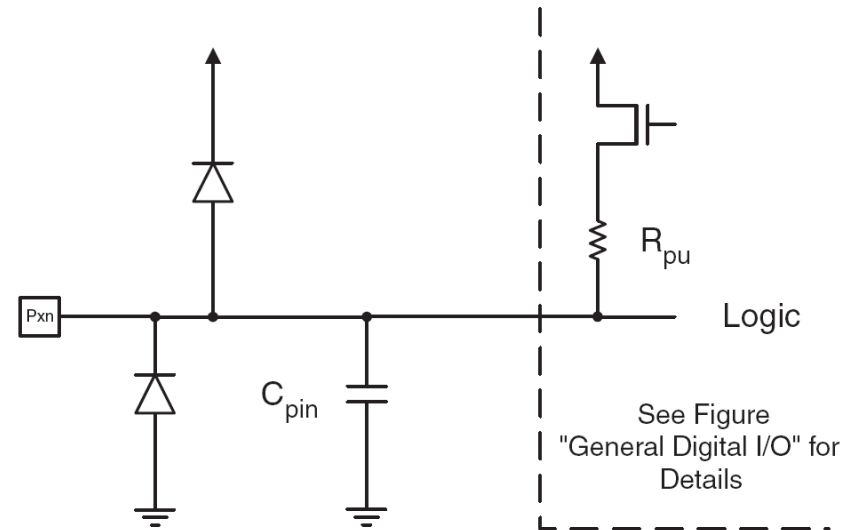


Figure 13-1. I/O Pin Equivalent Schematic



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 (PD3, PD5, i PD7) kao izlazne

- Arduino pristup

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

Ili ako je upotrijebljena me106.h:

```
pinMode(PIN_D3, OUTPUT);  
pinMode(PIN_D5, OUTPUT);  
pinMode(PIN_D7, OUTPUT);
```

- Alternativni pristup

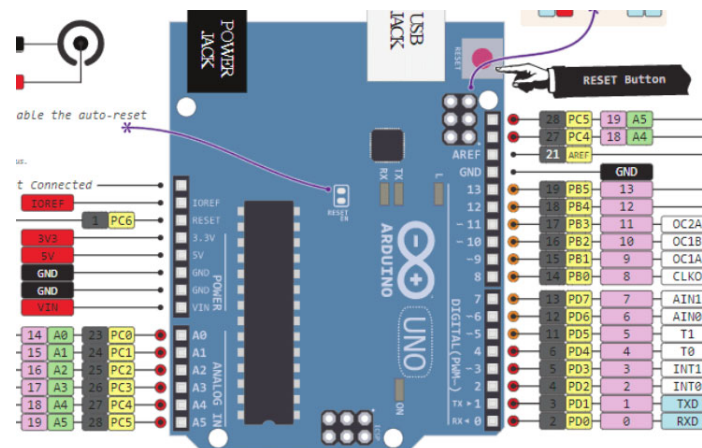
```
DDRD = 0b10101000;
```

ili

```
DDRD = 0xA8;
```

ili

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



# Arduino C je izveden iz C++

## ■ Ovaj program radi treperenje LED na pinu 13

- avr-libc

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

```
int main(void) {
    while (1) {
        PORTB = 0x20;
        _delay_ms(1000);
        PORTB = 0x00;
        _delay_ms(1000);
    }
    return 1;
}
```

- Arduino C

```
void setup( ) {
    pinMode(13, OUTPUT);
}
```

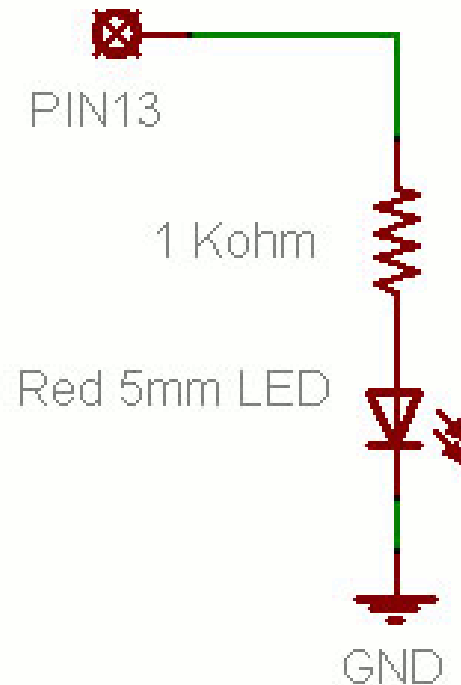
```
void loop( ) {
    digitalWrite(13, HIGH);
    delay(1000);
    digitalWrite(13, LOW);
    delay(1000);
}
```





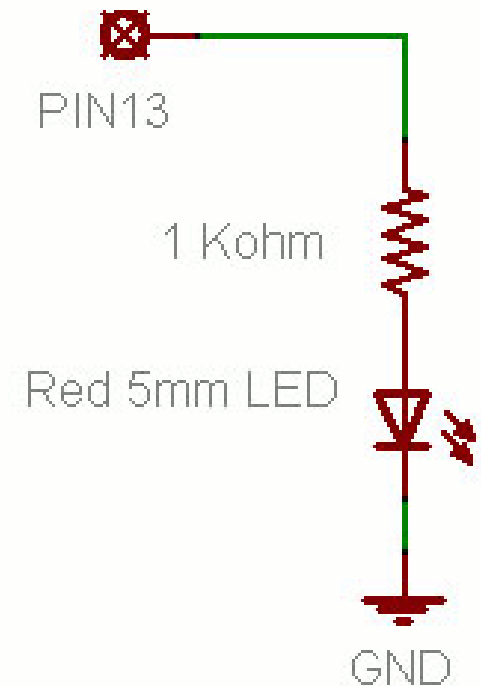
# Prosto elektronsko kolo

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



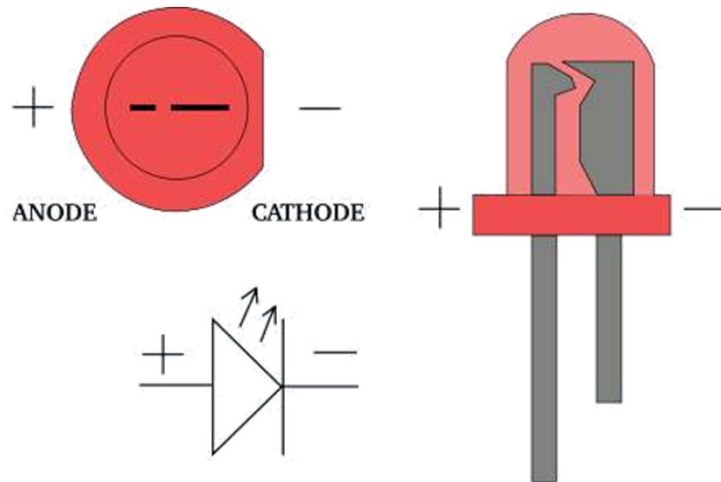
# Osnovno LED kolo

- Povežite pin 13 mikrokontrolera na jedan kraj otpornika.
- Drugu nožicu otpornika spojite na dužu nožicu LED.
  - Veća otpornost znači slabije svjetlo.
  - Manja otpornost znači jače svjetlo.
  - Bez otpornosti znači pregorijevanje LED ili preopterećenje porta.
- Kraću nožicu LED spojite na negativni priključak napajanja (masu).



# Blink Skeč (Treperenje)

- **File > Examples > Digital > Blink**
- LED ima polaritet
  - Negativni je indikovani zasječenim obodom tijela diode i kraćom nožicom.

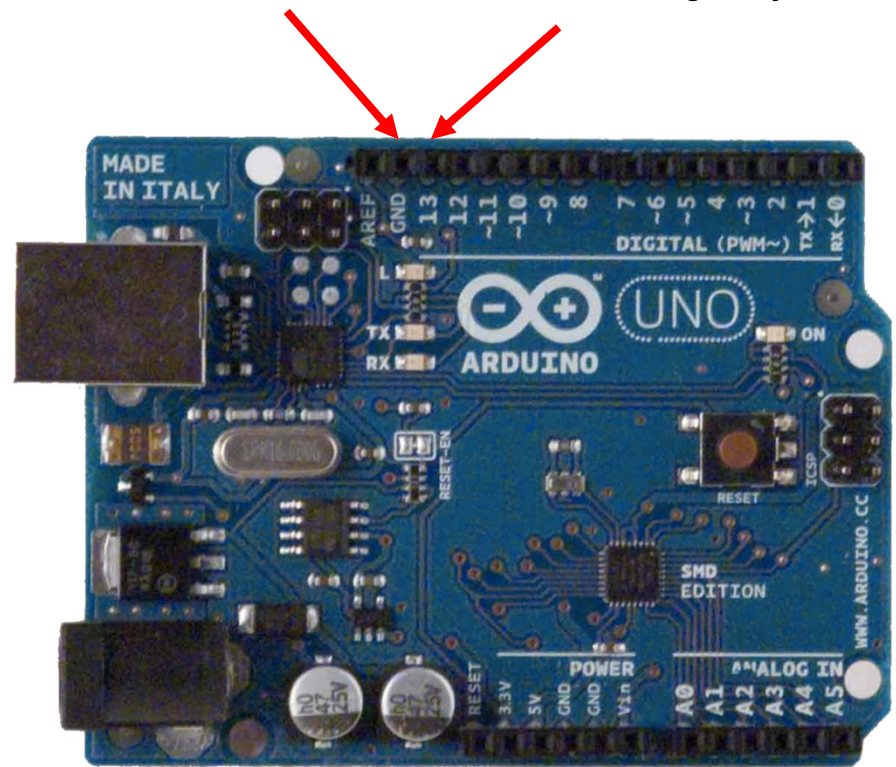


# Blink Skeč (Treperenje)

```
void setup( ) {  
  pinMode(13, OUTPUT);  
}  
void loop( ) {  
  digitalWrite(13, HIGH);  
  delay(1000);  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```

Povežite jedan kraj kola

Povežite drugi kraj kola



# Struktura Arduino programa

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

```
/* Blink - turns on an LED for DELAY_ON msec,
then off for DELAY_OFF msec, and repeats
BJ Furman rev. 1.1 Last rev: 22JAN2011
*/
#define LED_PIN 13 // LED on digital pin 13
#define DELAY_ON 1000
#define DELAY_OFF 1000

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

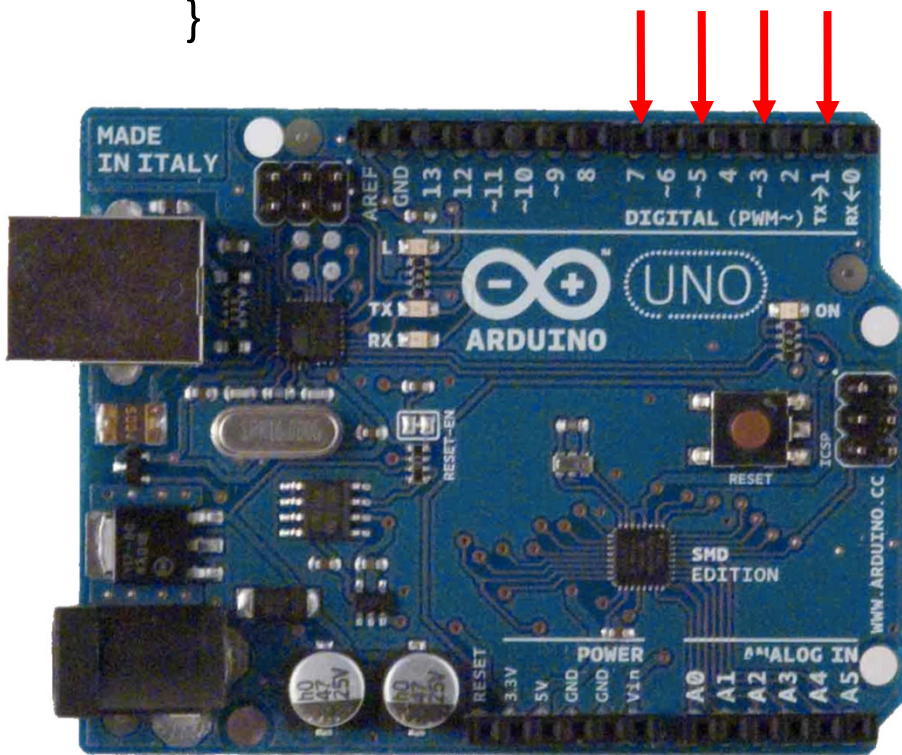
// loop() method runs forever,
// as long as the Arduino has power

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

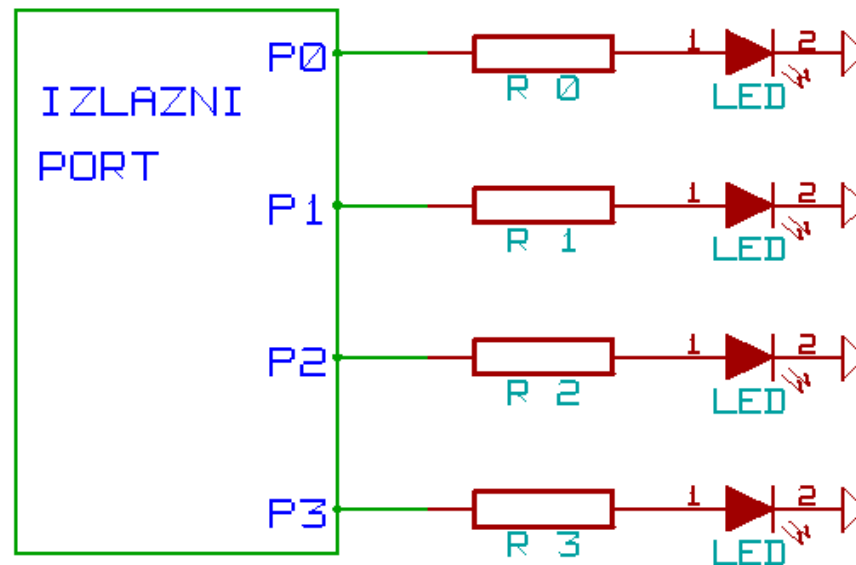


# I što?

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

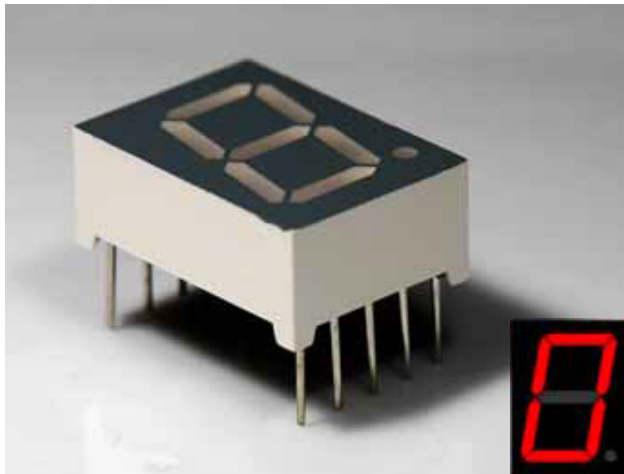
# Zadaci za vježbu 1

1. Pomocu 4 LED, u binarnom obliku prikazati vrijednost promjenjive BROJAC. Vrijednost promjenjive brojac se inkrementira svake sekunde. **(2 poena)**
2. Trcece svjetlo upotrebom 4 LED. Uvijek je samo jedna dioda uključena. **(3 poena)**





3. Napisati program koji broji od 0 do 9, s ponavljanjem, i prikazuje rezultat na sedmosegmentnom LED displeju. (4 poena)



Pomoć:

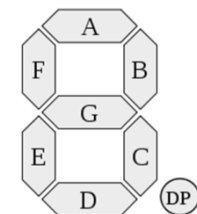
Razmotrite pisanje funkcije:

```
void writeDigit(int n)
```

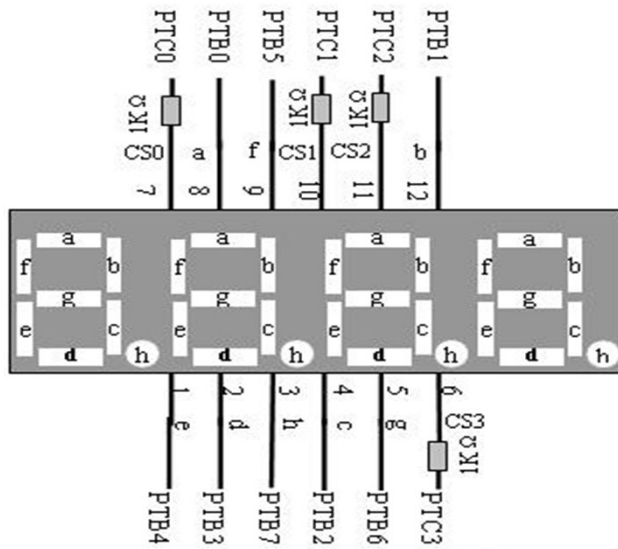
koja ispisuje jednu cifru

Trasformaciona tabela

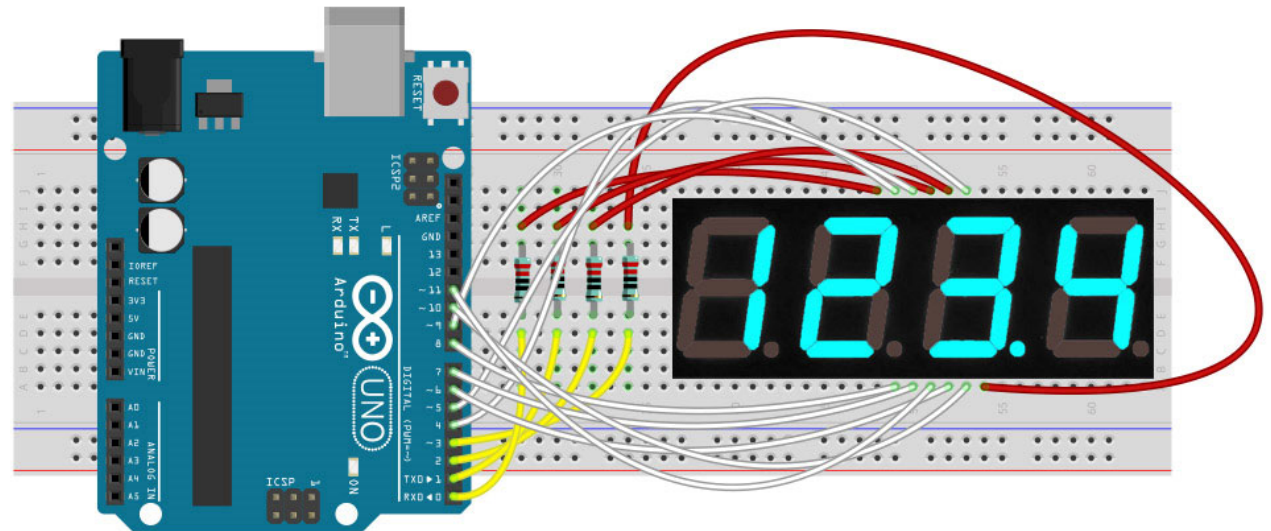
Digit	ABCDEFG	A	B	C	D	E	F	G
0	0x7E	on	on	on	on	on	on	off
1	0x30	off	on	on	off	off	off	off
2	0x6D	on	on	off	on	on	off	on
3	0x79	on	on	on	on	off	off	on
4	0x33	off	on	on	off	off	on	on
5	0x5B	on	off	on	on	off	on	on
6	0x5F	on	off	on	on	on	on	on
7	0x70	on	on	on	off	off	off	off
8	0x7F	on	on	on	on	on	on	on
9	0x7B	on	on	on	on	off	on	on



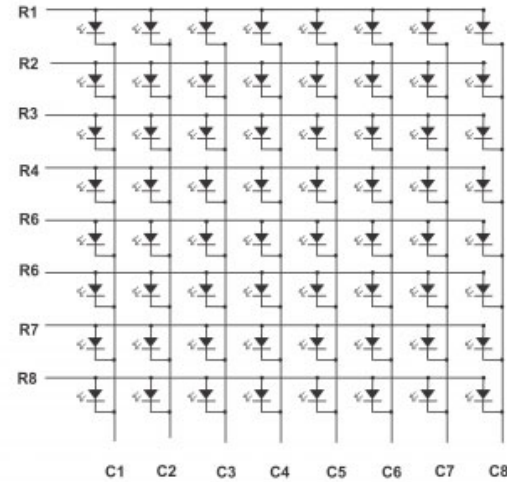
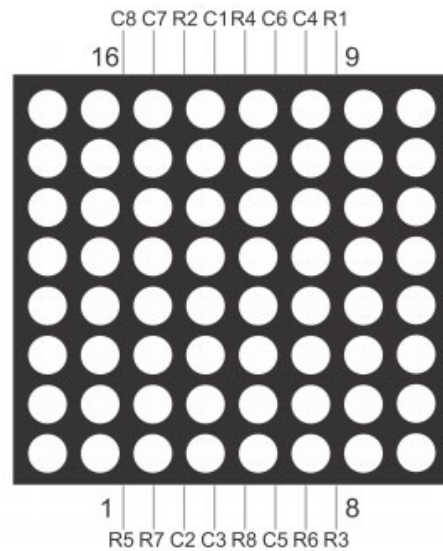
4. Napisati program koji na četvorocifarskom sedmosegmentnom LED displeju ispisiše: '432.1', i to nasljedeći način: najprije ispise samo prvu cifru slijeva, zadrži dva sekunda, ispise prvu i drugu , zadrži 2 sekunda, itd. Na kraju ispise sve 4 cifre i zadrži 5 sekundi. (6 poena)



MCU与4连排8段数码管的连接



5. Napisati program koji na 8X8 matrix LED displeju ispisi cife od 0 do 9, stalno u krug. Pojedina cifra se na displeju zadržava dvije sekunde. (8 poena)



# Kuraj