

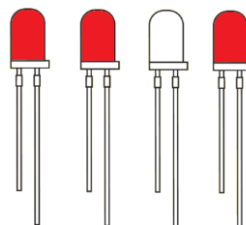
14. Operacije nad bitovima, pointeri, registri

BITWISE OPERACIJE I POINTERI

Zajedno će se objasniti na ploči ☺ a zatim naučeno primjeni na sljedeći zadatak.

ZADATAK 1. Spoji 4 LED-ice (ili po izboru više) na Arduino. Njih ćemo koristiti za binarni prikaz broja. Na *Serial Monitor* je potrebno javiti korisniku da unese jedan broj koji želi pretvoriti u binarni zapis. Pošto imamo četiri LED-ice na raspolaganju, neka taj broj bude u rasponu od 0 do 15.

Ako korisnik unese broj izvan raspona, na *Serial Monitor* se ispisuje poruka korisniku da ponovno upiše broj iz zadanog raspona. Uneseni broj pretvori u binarni (koristi operacije nad bitovima) te zatim prikaži na LED-icama (1 svijetli, 0 ne svijetli).



PRIMJER IZLAZA ZA BROJ 13

Za operacije nad bitovima googlaj „bitwise operators arduino“, a posebno „bitwise AND“ i „bitshift LEFT“. Koristan ti može biti i „Bit Math Tutorial“ (prva stvar koju dobiješ kad to upišeš u google).

ZADATAK 2. Modificiraj prethodni program tako da se pretvaranje broja u stanja 4 ledice radi u funkciji `toLedSignals` koja je definirana na način:

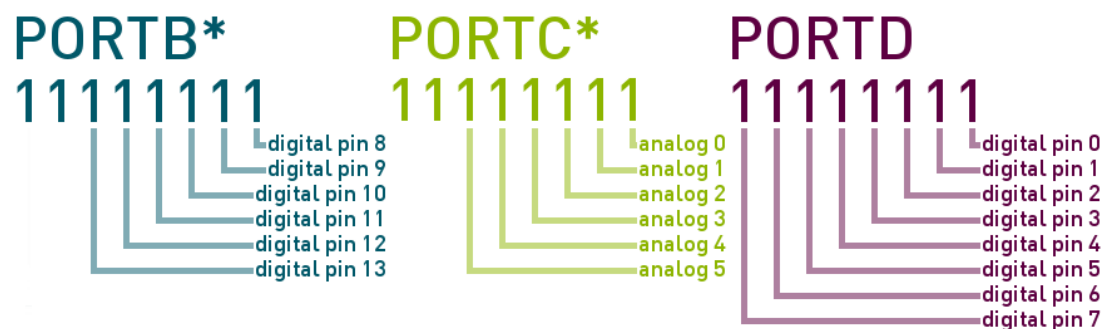
```
void toLedSignals (int num, int* led1, int* led2, int* led3,
int* led4) {}
```

Unutar funkcije stanja `led1`, `led2` itd mijenjaš pristupajući im koristeći pointerne:

```
*led1=...;
```

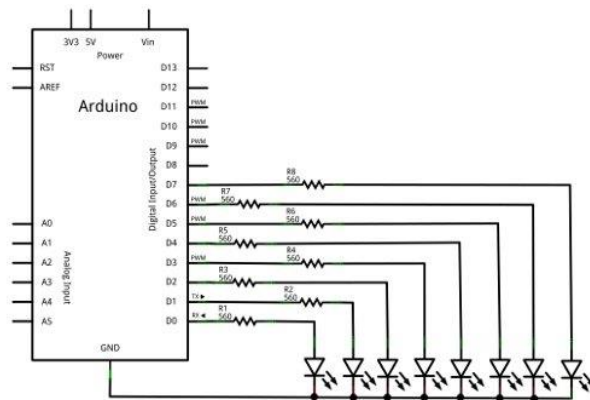
Fukciju `toLedSignals` iz loopa ćeš pozvati predajući adrese `led1`, `led2`, `led3`, i `led 4` na način:

```
toLedSignals (inputNum, &led1, &led2, &led3, &led4);
```



REGISTRI

ZADATAK 3. Spoji komponente prema shemi (ne moraš koristiti otpornike)



i prouči idući primjer koda (bit će detaljno objašnjeno na ploči, od pomoći ti može biti *dodatak 1* na prethodnoj stranici) te ga isprobaj:

(kod možeš naći i ovdje <https://goo.gl/FB3iHa>)

```
void setup()
{
  DDRD = B11111111; // postavi PORTD (digitalni 7~0)
                    // na izlaz
}

byte a = B11111111;
byte b = B00000001;
byte c = B10000000;
byte e = B10101010;

void gasi_pali()
{
  for (int k=0; k<10; k++)
  {
    PORTD = a;
    delay(100);
    PORTD = 0;
    delay(100);
  }
}

void blinkaj()
{
  for (int z=0; z<10; z++)
  {
    PORTD = e;
    delay(100);
    PORTD = ~e;
    delay(100);
  }
}

void broji_binarno()
{
}
```

```

    for (int z=0; z<256; z++)
    {
        PORTD = z;
        delay(100);
    }
    PORTD=0;
}

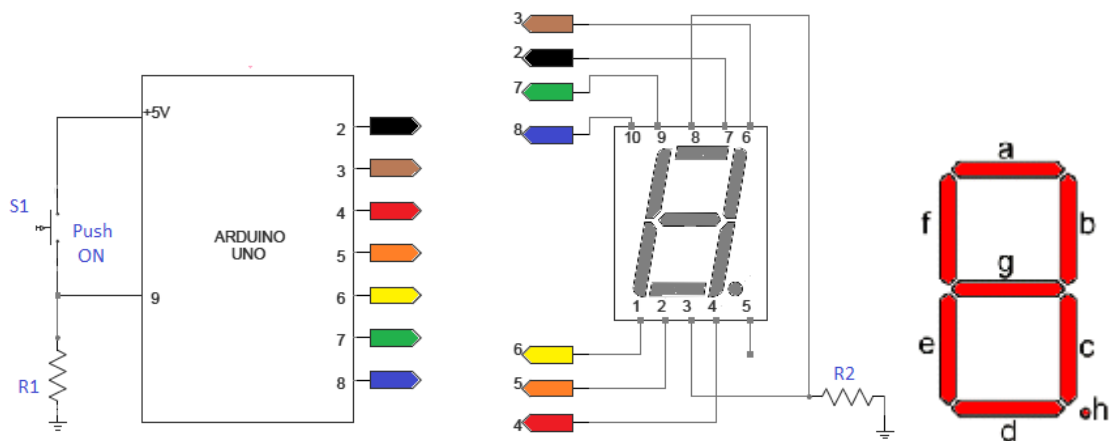
void ld()
{
    for (int k=0; k<5; k++)
    {
        for (int z=0; z<8; z++)
        {
            PORTD = b << z;
            delay(100);
        }

        for (int z=0; z<8; z++)
        {
            PORTD = c >> z;
            delay(100);
        }
    }
}

void loop()
{
    blinkaj();
    delay(500);
    broji_binarno();
    delay(500);
    ld();
    delay(500);
    gasi_pali();
}

```

ZADATAK 4. Spoji komponente prema shemi (R1=10 kΩ, R2=330 Ω):



Koristeći registre (direktan pristup *portovima*) treba napisati „brojilo“ koje će brojati kada je gumb pritisnuti, a kada nije zaustaviti se na zadnjoj vrijednosti. Brzina brojanja je proizvoljna.

Hint: koristi polje „segmenti“, njime je definirano koji segmenti su za koje znamenke (0, 1, 2...)

```

byte segmenti[] = { B00111111, B00000110, B01011011, B01001111, B01100110, B01101101, B01111101, B00000111,
B01111111, B01101111 };

```