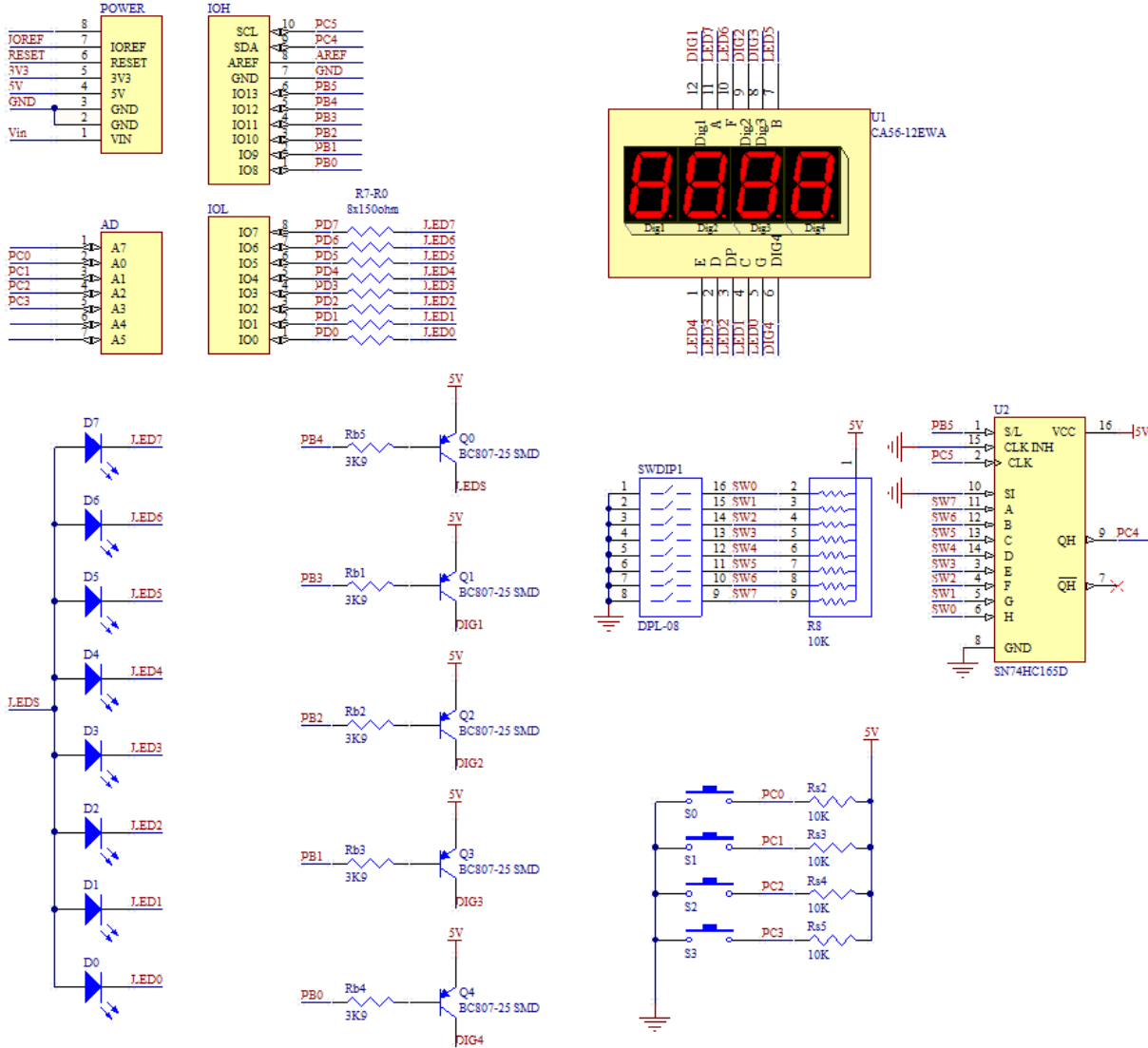


# Arduino PLS7 espansione ploča (4x7SEG + LED+ tasteri + prekidači)

## Šema PLS7 espansione ploče



## LED diode

Niz od 8 LED dioda (*LED7-LED0*) povezan je u spoju sa zajedničkom anodom. Spoj između napona napajanja ( $V_{cc}=5V$ ) i čvora na koji su vezane anode ostvaruje se preko prekidačkog tranzistora *Q0*. Tranzistorom *Q0* se upravlja pomoću pina **PB4**. Pošto je u pitanju PNP tranzistor čiji emiter je spojen na napon napajanja, njegovo uključivanje se vrši dovođenjem niskog napona na bazu (odnosno kada je **PB4=0**), a kada je napon na bazi visok (**PB4=1**), *Q0* će biti isključen, čime se istovremeno prekida proticanje struje kroz ceo niz LED dioda.

Katode su preko otpornika *R7-R0* vrednosti  $150\Omega$  povezane sa pinovima porta D mikrokontrolera *ATMega328p* (**PD7-PD0**). Na ovaj način moguće je kontrolisati stanje svake LED diode pojedinačno. Za uključenje diode na poziciji *n*, pored toga što tranzistor *Q0* mora biti uključen, potrebno je još i postaviti pin **PDn** na nizak logički nivo (**PDn=0**). U suprotnom, kada je **PDn=1**, na katodi se pojavljuje visok napon, čime se dioda isključuje.

### Primer:

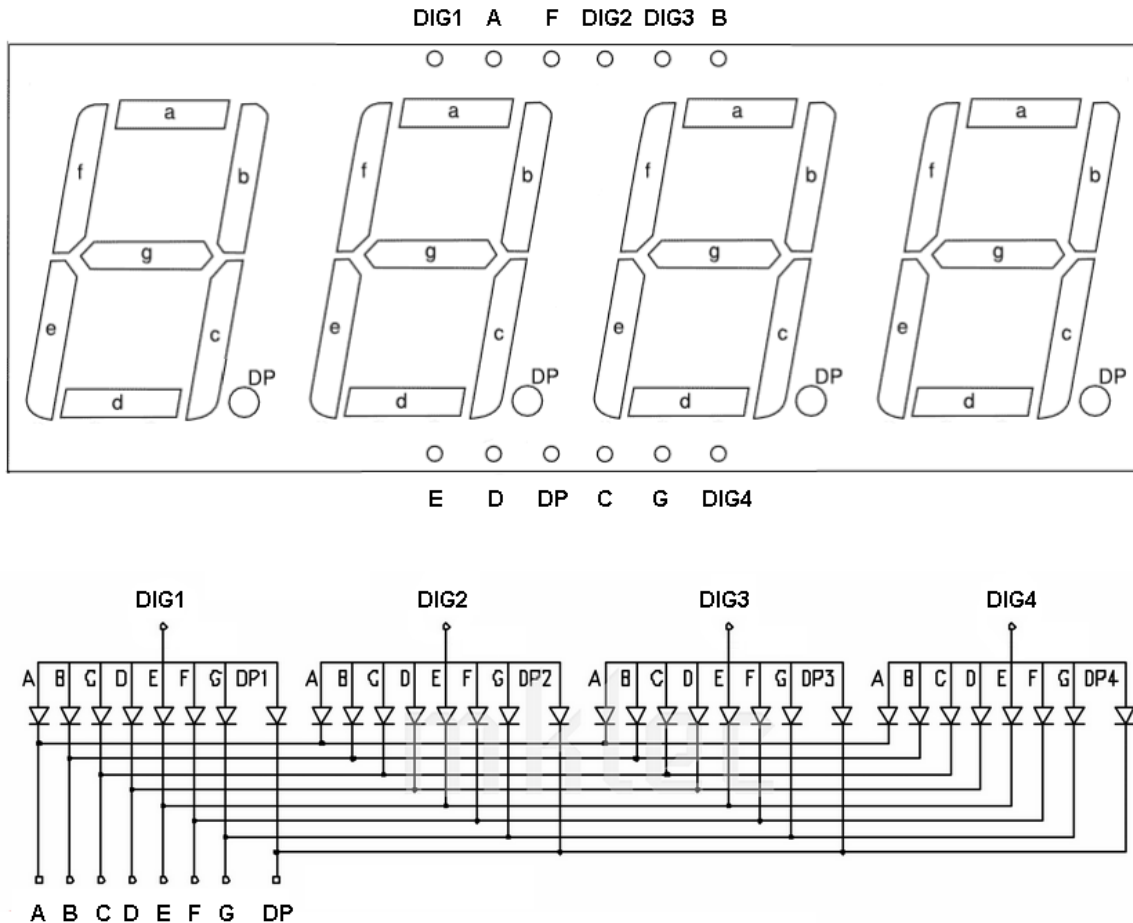
```
#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
    DDRD = 0xff;           //port D -> izlaz
    DDRB |= 1 << 4;       //PB4 -> izlaz
    PORTB &= ~(1 << 4);   //PB4 = 0, cime se ukljucuje tranzistor Q0

    while(1)
    {
        PORTD = 0xaa;     //ukljucuju se LED na pozicijama 6, 4, 2 i 0
        _delay_ms(1000); //pauza 1s
        PORTD = 0x55;     //ukljucuju se LED na pozicijama 7, 5, 3 i 1
        _delay_ms(1000); //pauza 1s
    }
    return 0;
}
```

## LED displej 4x7SEG

Na ploči se nalazi komponenta *U1* koja sadrži četiri sedmosegmentna displeja sa decimalnom tačkom, integrisana u zajedničko kućište. LED diode koje čine segmente displeja povezane su u spoju sa zajedničkom anodom. Katode odgovarajućih segmenata kod sva 4 displeja kratko su spojene, kao što je prikazano na internoj šemi displeja (dole na slici):



Na različitim displejima (*DIG1-DIG4*), katode odgovarajućih segmenata (A-DP<sub>x</sub>) su kratko spojene, a anode su razdvojene. Na red sa anodama displeja *DIG1-DIG4* povezani su prekidački tranzistori *Q1-Q4* pomoću kojih je moguće prekidati struju i time uključivati/isključivati pojedine displeje. Anode su preko otpornika za ograničavanje struje *R7-R0* povezane sa pinovima porta D mikrokontrolera (**PD7-PD0**). Logika upravljanja displejima realizuje se tzv. vremenskim multipleksiranjem: displej se osvežava u pravilnim vremenskim intervalima. Perioda osvežavanja displeja *T* podeljena je na 4 intervala jednakog trajanja:  $T_1 = T_2 = T_3 = T_4 = T/4$ . Na

početku intervala T1, uključuje se tranzistor *Q1* i isključuju tranzistori *Q2-Q4*. Tada stanja pinova porta D (**PD7-PD0**) određuju stanja pojedinih segmenata displeja *DIG1*<sup>1</sup>. Nakon toga, tokom intervala T2 uključuje se *Q2*, ostali tranzistori se isključuju, a pinovi porta D kontrolišu segmente displeja *DIG2*, i tako redom do poslednjeg displeja *DIG4*. Frekvencija osvežavanja treba da bude dovoljno velika (bar 25Hz), kako bi korisnik imao iluziju da je prikaz na displeju nepomičan. Tranzistori *Q1-Q4* kontrolišu se pinovima porta B (**PB4-PB0**)<sup>2</sup>. U narednoj tabeli prikazana je veza između stanja pinova na portovima B i D i segmenata na displeju, odnosno LED dioda:

Aktivan displej	PB4	PB3	PB2	PB1	PB0	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
LEDS	0	1	1	1	1	LED <sub>7</sub>	LED <sub>6</sub>	LED <sub>5</sub>	LED <sub>4</sub>	LED <sub>3</sub>	LED <sub>2</sub>	LED <sub>1</sub>	LED <sub>0</sub>
DIG1	1	0	1	1	1	A <sub>1</sub>	F <sub>1</sub>	B <sub>1</sub>	E <sub>1</sub>	D <sub>1</sub>	DP <sub>1</sub>	C <sub>1</sub>	G <sub>1</sub>
DIG2	1	1	0	1	1	A <sub>2</sub>	F <sub>2</sub>	B <sub>2</sub>	E <sub>2</sub>	D <sub>2</sub>	DP <sub>2</sub>	C <sub>2</sub>	G <sub>2</sub>
DIG3	1	1	1	0	1	A <sub>3</sub>	F <sub>3</sub>	B <sub>3</sub>	E <sub>3</sub>	D <sub>3</sub>	DP <sub>3</sub>	C <sub>3</sub>	G <sub>3</sub>
DIG4	1	1	1	1	0	A <sub>4</sub>	F <sub>4</sub>	B <sub>4</sub>	E <sub>4</sub>	D <sub>4</sub>	DP <sub>4</sub>	C <sub>4</sub>	G <sub>4</sub>

### Primer:

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

const unsigned char simboli[] = {
    0x0c, 0xa4, 0x27, 0xc4
}; //look-up tabela sa simbolima za ispis na displej (A,b,C,d)

int main(void)
{
    unsigned char displej;
    DDRD = 0xff; //port D -> izlaz
    DDRB = 0x0f; //PB3 - PB0 -> izlazi
    while(1)
    {
        for (displej = 1; displej <= 4; displej++)
        {
            PORTB = ~(0x01 << (4-displej)); //uključuje se tranzistor
                                                //na odgovarajućoj poziciji
            PORTD = simboli[displej-1]; //ispis simbola iz tabele
            _delay_ms(2); //pauza 2ms
        }
    }
    return 0;
}
```

<sup>1</sup> Kao i kod niza LED dioda, pojedini segmenti displeja *DIGN* uključuju se postavljanjem pinova porta D na logičku nulu (**PDx = 0**), pod uslovom da je odgovarajući prekidački tranzistor *Qn* uključen.

<sup>2</sup> Poput tranzistora *Q0* i tranzistori *Q1-Q4* su PNP sa emiterom direktno spojenim na *Vcc*. Uključivanje se vrši dovođenjem niskog napona na bazu, odnosno kada je odgovarajući pin **PBn = 0**.

## Tasteri

Tasteri S3-S0 povezani su direktno na pinove porta C (**PC3-PC0**). Svaki od tastera povezan je između ulaznog pina i mase, u kombinaciji sa *pull-up* otpornikom. To znači da kada taster nije pritisnut, stanje pina će biti 1 usled prisustva *pull-up* otpornika, a u pritisnutom stanju taster kratko spaja pin na masu, što se čita kao logička 0.

### Primer:

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

const unsigned char simboli[] = { 0x0c, 0xa4, 0x27, 0xc4 };

void ispis_7SEG (unsigned char karakter, unsigned char pozicija)
{
    PORTB = ~(0x01 << (4-pozicija));    //uključenje displeja na zeljenoj
                                        //poziciji
    PORTD = karakter;                    //prikaz karaktera
    _delay_ms(2);                        //pauza 2ms
}

int main(void)
{
    unsigned char tasteri;
    DDRD = 0xff; //port D -> izlaz
    DDRC = 0x00; //port C -> ulaz
    DDRB = 0x0f; //PB3 - PB0 izlazi
    while (1)
    {
        tasteri = PINC & 0x0f; //ocitavanje stanja tastera
        if (!(tasteri & 0x01)) //provera stanja tastera S0
            ispis_7SEG (simboli[0], 1);
        else
            ispis_7SEG (0xfe, 1);

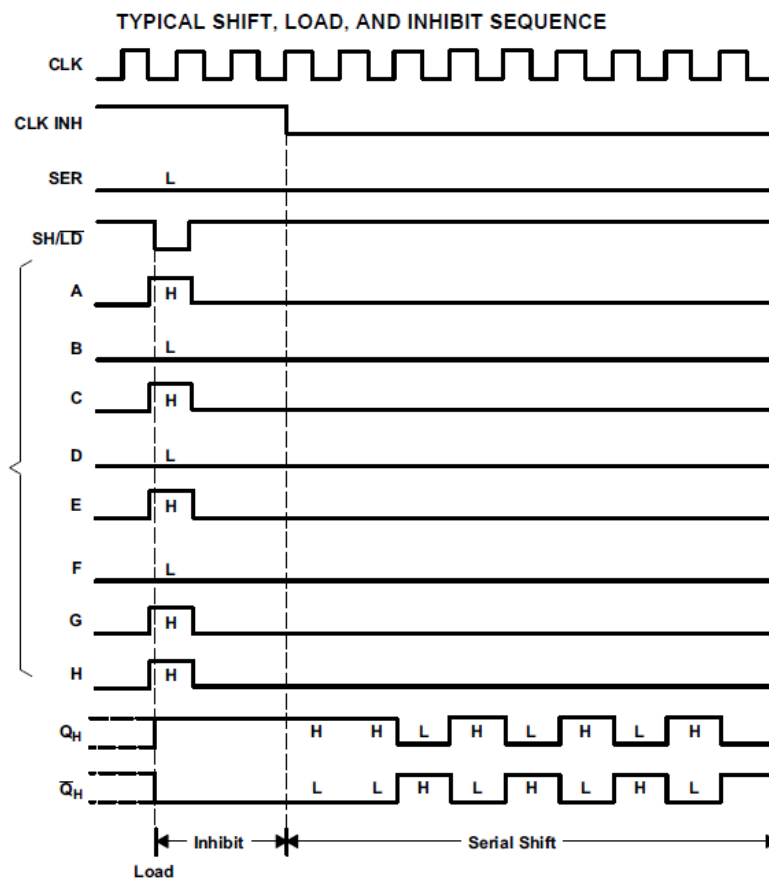
        if (!(tasteri & 0x02)) //provera stanja tastera S1
            ispis_7SEG (simboli[1], 2);
        else
            ispis_7SEG (0xfe, 2);

        if (!(tasteri & 0x04)) //provera stanja tastera S2
            ispis_7SEG (simboli[2], 3);
        else
            ispis_7SEG (0xfe, 3);

        if (!(tasteri & 0x08)) //provera stanja tastera S3
            ispis_7SEG (simboli[3], 4);
        else
            ispis_7SEG (0xfe, 4);
    }
    return 0;
}
```

## Prekidači

Komponenta *SWDIP1* koja sadrži 8 prekidača u zajedničkom kućištu, nije direktno povezana na pinove mikrokontrolera, pošto bi za tako nešto bilo potrebno 8 pinova, a povezivanjem već opisanih komponenti preostala su samo 3 raspoloživa pina. Stoga su prekidači povezani na 8-bitni pomerački registar sa mogućnošću paralelnog upisa *74HC165* (komponenta *U2* na šemi). Pin mikrokontrolera **PB5** povezan je na liniju za dozvolu upisa u registar (*SH/LD*) i upravlja njenim stanjem. Upis tekućeg stanja svih 8 prekidača u registar vrši se postavljanjem ove linije na logičku 0 (**PB5 = 0**). Nakon toga, potrebno je ukinuti dozvolu upisa (postaviti **PB5 = 1**), čime se registar "zaključava" za dalji upis. Čitanje vrednosti registra obavlja se serijskim protokolom, na sledeći način: promenom stanja pina **PC5** mikrokontrolera generiše se takt signal (doveden na *CLK* ulaz) kojim se vrši pomeranje sadržaja registra. Serijski izlaz registra (*QH* izlaz) povezan je na pin **PC4**, preko koga se čita bit po bit tokom 8 perioda takta. Vremenski dijagrami tokom jednog ciklusa učitavanja stanja prekidača u kolo *74HC165* i njihovog serijskog prenosa prikazani su na slici<sup>3</sup>:



<sup>3</sup> Linija *CLK INH* kojom se zabranjuje pomeranje sadržaja registra kratko je spojena na masu, čime je data stalna dozvola pomeranja.

## Primer:

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

//makroi za kontrolu pinova preko kojih je kontroler povezan sa 74HC165:
#define SCL_HI (PORTC |= (1<<5))
#define SCL_LO (PORTC &= ~(1<<5))
#define SDA (PINC & (1 << 4))
#define SHLD_HI (PORTB |= (1<<5))
#define SHLD_LO (PORTB &= ~(1<<5))

unsigned char ocitaj_prekidace()
{
    unsigned char i, tmp = 0, mask = 0x80;

    //impuls za upis stanja prekidaca u registar
    SHLD_HI;
    SHLD_LO;
    SHLD_HI;

    for (i=0; i<8; i++)
    {
        SCL_LO;
        SCL_HI;    //generisanje aktivne ivice takta

        if (SDA)    //provera stanja ulaznog pina
            tmp |= mask;
        mask >>= 1;
    }
    return tmp;
}

int main(void)
{
    DDRD = 0xff;    //port D -> izlaz
    DDRC = 0x20;    //PC5 -> izlaz
    DDRB = 0x30;    //PB5 i PB4 -> izlazi

    PORTB = ~0x10; //uključenje tranzistora Q0

    while (1)
        PORTD = ocitaj_prekidace();

    return 0;
}
```

## PRILOG: Funkcije pinova mikrokontrolera ATmega328p

Port	Pin	Ulaz/Izlaz	Funkcija	Logika
PORTB	PB5	Izlaz	Upis stanja prekidača u 74HC165	0 = UPIS
	PB4	Izlaz	Kontrola tranzistora Q0 (niz 8 LED dioda)	0=ON, 1 = OFF
	PB3	Izlaz	Kontrola tranzistora Q1 (displej DIG1)	0=ON, 1 = OFF
	PB2	Izlaz	Kontrola tranzistora Q2 (displej DIG2)	0=ON, 1 = OFF
	PB1	Izlaz	Kontrola tranzistora Q2 (displej DIG3)	0=ON, 1 = OFF
	PB0	Izlaz	Kontrola tranzistora Q2 (displej DIG4)	0=ON, 1 = OFF
PORTC	PC5	Izlaz	Takt za 74HC165	Pomeranje na rastuću ivicu (↑) Neinvertujuća 0=PRITISNUT, 1=PUŠTEN
	PC4	Ulaz	Serijski ulaz za bite sa 74HC165	
	PC3	Ulaz	Stanje tastera S3	
	PC2	Ulaz	Stanje tastera S2	
	PC1	Ulaz	Stanje tastera S1	
	PC0	Ulaz	Stanje tastera S0	
PORTD	PD7	Izlaz	LED 7, ili segment A na aktivnom displeju	0=ON, 1 = OFF
	PD6	Izlaz	LED 6, ili segment F na aktivnom displeju	0=ON, 1 = OFF
	PD5	Izlaz	LED 5, ili segment B na aktivnom displeju	0=ON, 1 = OFF
	PD4	Izlaz	LED 4, ili segment E na aktivnom displeju	0=ON, 1 = OFF
	PD3	Izlaz	LED 3, ili segment D na aktivnom displeju	0=ON, 1 = OFF
	PD2	Izlaz	LED 2, ili segment DP na aktivnom displeju	0=ON, 1 = OFF
	PD1	Izlaz	LED 1, ili segment C na aktivnom displeju	0=ON, 1 = OFF
	PD0	Izlaz	LED 0, ili segment G na aktivnom displeju	0=ON, 1 = OFF

