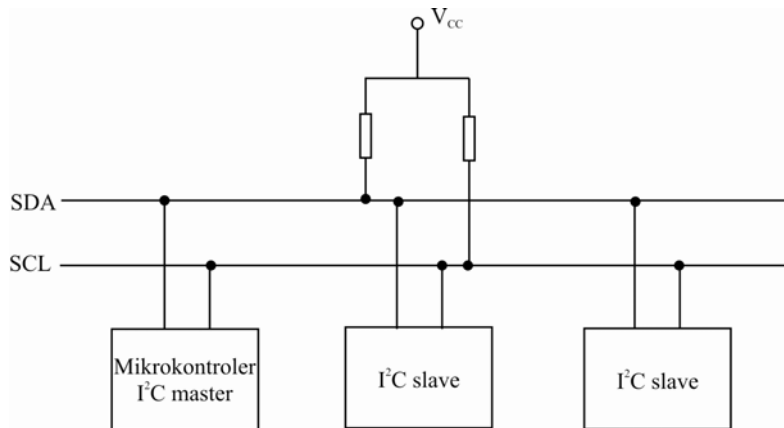


## I2C komunikacija

I2C („Inter-Integrated Circuit“) komunikaciju je razvio Philips. Ovaj vid komunikacije se koristi za mala rastojanja, najčešće za povezivanje integriranih kola na štampanoj ploči. Za ovaj protokol su potrebne dve fizičke linije tipa open drejn. Ove linije se nazivaju SCL (Serial CLock) i SDA (Serial DAta). Na I2C magistralu se mogu vezati 112 kola, a njima se pristupa preko 7-bitne adrese.

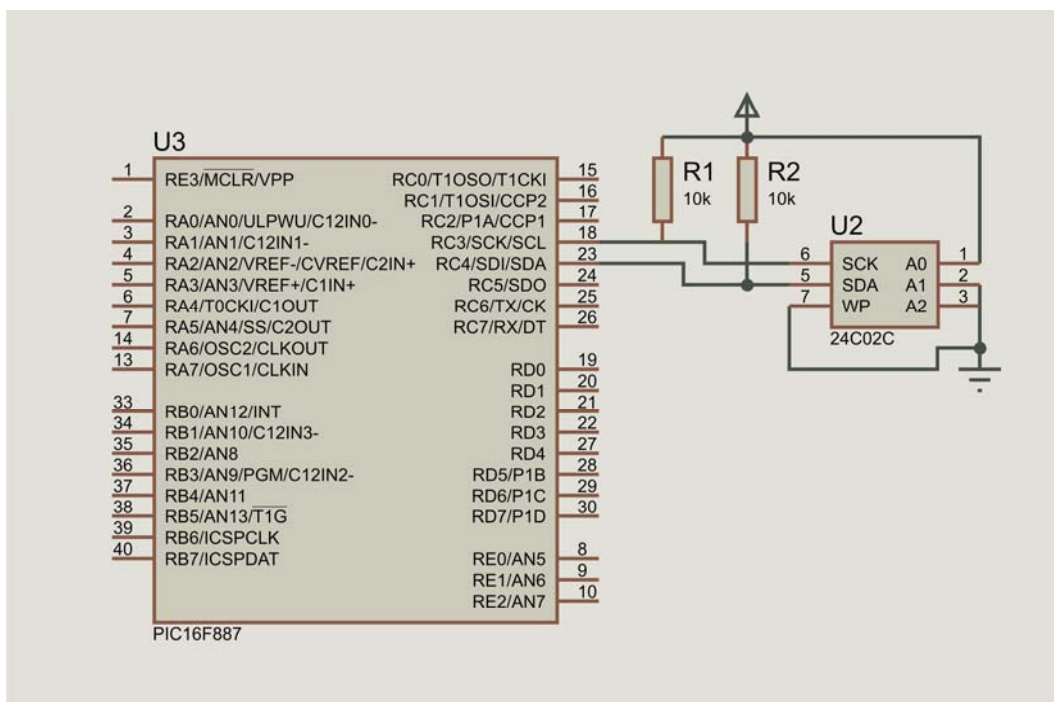


Sl. 1. Način povezivanja uređaja na magistralu.

Komunikacija započinje start bitom (S) kada se SDA linija sa jedinice obori na nulu dok je SCL još na logičkoj jedinici. Zatim dolazi do slanja podataka preko SDA linije tako što se bit na SDA liniji postavlja dok je SCL na nuli, a uzorkuje se kada je SCL na jedinici. Kada se završi prenos podataka tada se šalje stop bit (P) tako što se SDA postavi sa nule na jedinicu dok je SCL na jedinici.

Ukoliko se na istoj magistrali nalaze više master uređaja onda se mora voditi računa o koliziji, odnosno situaciji kada dva ili više mastera pokušava da istovremeno pošalje podatak. U ovom slučaju master uređaj posle postavljenog bita mora i da pročita da li se zaista na magistrali nalazi postavljeni bit.

1. Za početak, će se obraditi primer iz uputstva za MikroC (strana 341 do 345). Primer predstavlja komunikaciju mikrokontrolera sa memorijom 24C02 (tehničke specifikacije su date u materijalu na sajtu).



Sl. 2. Simulaciona šema primera iz materijala za MikroC (strana 345).

Kod ove memorije u kontrolnom bajtu gornja četiri bita su rezervisana dok donja četiri služe za adresu i komandu za čitanje ili upis. Format naredbe treba biti u obliku:

1	0	1	0	A2	A1	A0	R/W
---	---	---	---	----	----	----	-----

Ovde se treba obratiti pažnja na povezivanje adresnih bitova memorijskog čipa. Sa slike 2 se vidi da je A0 na visokom potencijalu, dok su A1 i A2 na niskom potencijalu. Znači da kontrolni bit treba da ima sledeće vrednosti:

10100010 (A2 u heksadecimalnom zapisu) – u slučaju upisa

10100011 (A3 u heksadecimalnom zapisu)- u slučaju čitanja

Pošto poseduje samo tri bita za adresu moguće je samo 8 ovakvih memorija povezati na istu magistralu. Prilikom upisa prvo ide start bit zatim kontrolni bajt nakon toga sledi odgovor prijemnika (ACK – potvrda prijema), nakon toga sledi adresa na koju treba upisati podatak zatim sledi ACK posle koga se šalje podatak i posle ACK ide stop bit.

```

void main()
{
    ANSEL = 0;
    ANSELH = 0;
    PORTB = 0;           //postavljanje porta B na nulu
    TRISB = 0;          // konfiguracija B porta kao izlaznog

    I2C1_Init(100000);   // inicijalizacija I2C komunikacije sa odgovarajućom frekvencijom

    I2C1_Start();        // početak I2C komunikacije
    I2C1_Wr(0xA2);       // slanje kontrolnog bajta (adresa memorijskog čipa sa upisom)
    I2C1_Wr(2);          // slanje bajta podataka (adresa EEPROM memorijske lokacije)
    I2C1_Wr(0xF0);       // podatak koji će biti upisan na predhodno određenu memorijsku lokaciju
    I2C1_Stop();         // zaustavljanje I2C komunikacije

    Delay_100ms();

    I2C1_Start();        // ponovo iniciranje I2C komunikacije
    I2C1_Wr(0xA2);       // slanje kontrolnog bajta (adresa memorijskog čipa sa upisom)
    I2C1_Wr(2);          // slanje bajta podataka (adresa EEPROM memorijske lokacije)
    I2C1_Repeated_Start(); // "resetovanje" I2C komunikacije
    I2C1_Wr(0xA3);       // slanje kontrolnog bajta (adresa memorijskog čipa sa čitanjem)
    PORTB = I2C1_Rd(0u); // čitanje bajta podatka sa predhodno određene memorijske lokacije
    I2C1_Stop();         // zaustavljanje I2C komunikacije
}

```

Prikazani primer je prilično jednostavan. Na određenu memorijsku lokaciju memorije se vrši upis jednog bajta (F0 se upisuje na lokaciju 2). Obratiti pažnju na proceduru I2C komunikacije. Kod upisa u memoriju:

1. Start I2C
2. Slanje kontrolnog bajta (A2)
3. Slanje memorijske lokacije **na koju se vrši upis (2)**
4. Slanje bajta informacije (F0)
5. Stop I2C

Kod čitanja iz memorije:

1. Start I2C
2. Slanje kontrolnog bajta (A2)
3. Slanje memorijske lokacije **iz koje će se čitati podatak (2)**
4. Slanje kontrolnog bajta (A3)
5. Čitanje podatka sa memorijske lokacije (2)
6. Stop I2C

## 2. Za samostalni rad

Za kod koji je dat u nastavku nisu data nikakva dodatna objašnjenja ni komentari. Probajte sami da zaključite šta kod radi i dati odgovarajuća objašnjenja. Kolege koje su u mogućnosti neka izvrše simulaciju zadatog problema. I za ovaj primer se koristi PIC16F887 i memorijski čip 24C02C. Šta se još koristi u zadatku videti iz programskog koda i ranijih primera.

```
sbit LCD_RS at RB4_bit;
sbit LCD_EN at RB5_bit;
sbit LCD_D4 at RB0_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D7 at RB3_bit;

sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;

char dsp[15];
char dsp1[15];

unsigned int temp_res;
float res = 0;
short i = 0;

void main()
{
    TRISD = 0;
    portd = 0;
    Lcd_Init();
    Lcd_Cmd(_LCD_CURSOR_OFF);

    do {
        temp_res = ADC_Read(2);
        res = temp_res*0.00488;
        FloatToStr(res, dsp);
        IntToStr(temp_res, dsp1);

        I2C1_Init(100000);

        I2C1_Start();
        I2C1_Wr(0xA2);
        I2C1_Wr(0);

        for(i = 0; i < 15; i++)
        {
            I2C1_Wr(dsp[i]);
        }
        I2C1_Stop();

        Delay_ms(100);

        for(i = 0; i < 15; i++)
        {
            I2C1_Start();
```

```
        I2C1_Wr(0xA2);
        I2C1_Wr(i);
        I2C1_Repeated_Start();
        I2C1_Wr(0xA3);
        dsp1[i] = I2C1_Rd(0u);
        I2C1_Stop();
    }
    Lcd_Cmd(_LCD_CLEAR);
    Lcd_Out(1, 5, dsp);
    Lcd_Out(2, 5, dsp1);
} while (1);
}
```