

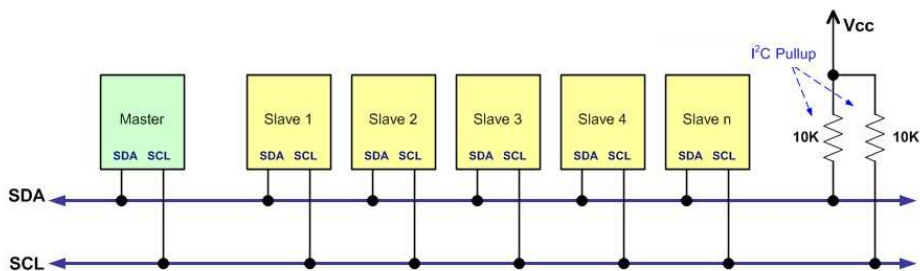
Rad sa satom/kalendarom realnog vremena

Obzirom da rad sa satom/kalendarom realnog vremena (eng. *real time clock/calendar*; RTC) zahteva razumevanje I²C magistrale i načina komunikacije, rad ove magistrale će u ovoj vežbi biti prvo objašnjen [22]. Potom će biti pokazan primer rada sa RTC kolom PCF8563 kompanije NXP [23]. Iskustva sa radom ovog kola u sprezi sa RPi su data u [24].

8.1. I²C komunikacija

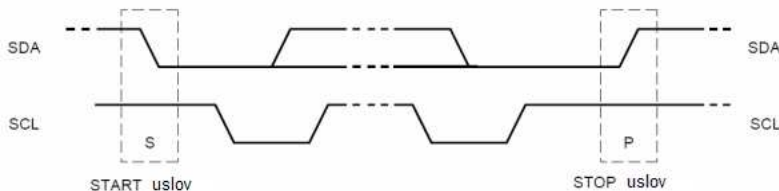
I²C (eng. *Inter-Integrated Circuit*, I „na kvadrat“ C, I dva C) komunikaciju je razvila kompanija Philips. Ovaj vid komunikacije koristi se za mala rastojanja, najčešće za međusobno povezivanje integrisanih kola na štampanoj pločici. Za ovaj protokol potrebne su dve fizičke linije tipa *open drain*. Ove linije nazivaju se SCL (eng. *Serial CLock*) i SDA (eng. *Serial DAta*). Linijom SCL prenosi se signal takta, a SDA služi za prenos podataka. Komunikacija je *master-slave* tipa i uvek je inicira master. Na jednu I²C magistralu može se povezati do 112 kola, a njima se pristupa preko 7-bitne adrese. Na slici 8.1 može se videti način povezivanja na I²C magistralu.

Kada je magistrala slobodna obe linije su na visokom naponskom nivou (zbog *pull-up* otpornika, uobičajenih vrednosti od 1,8 k do 47 k). Podaci na magistrali se mogu prenositi brzinama do 100 kb/s u standardnom modu, do 400 kb/s u *fast*-modu ili do 3,4 Mbit/s u *high-speed* modu.

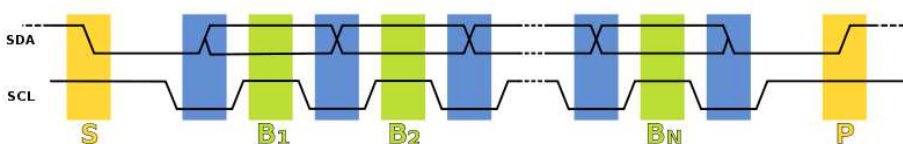


Slika 8.1. Način povezivanja uređaja na I²C magistralu

U okviru I²C komunikacije postoje dva stanja koja su definisana kao START (S) i STOP (P) stanja. Izgled tih signala može se videti na slici 8.2, a primer jednog komunikacionog transfera od N bita dat je na slici 8.3.



Slika 8.2. Start i stop stanja



Slika 8.3. Primer I²C komunikacije

Komunikacija započinje start bitom (S) kada SDA linija prelazi iz stanja logičke jedinice u stanje logičke nule dok je SCL još na logičkoj jedinici. Zatim dolazi do stanja podataka preko SDA linije tako što se bit na SDA liniji postavlja dok je SCL na nuli (plavo), a uzima se kada je SCL na jedinici (zeleno). 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. Komunikacija se obično obavlja u paketima od po 8 bita nakon čega uređaj koji je primio podatke šalje signal potvrde (ACK).

Pored START i STOP stanja postoji još i RESTART stanje koje restartuje slejv uređaj. RESTART stanje se postiže tako što se prvo pošalje start signal, pa se opet pošalje start signal koji je praćen sa stop signalom.

8.2. Povezivanje i adresiranje više I²C kola na zajedničku magistralu

Kao što je već rečeno na I²C magistralu moguće je povezati 112 uređaja (7-bitna adresa). Teorijski je moguće adresirati 128 uređaja, ali postoje rezervisane adrese pa je taj broj zato nešto niži. Postoji i opcija 10-bitnog adresiranja. Prilikom slanja adrese kao osmi bit šalje se bit koji definiše smer tj. da li je potrebno izvršiti čitanje ili pisanje (R/W). Logička nula predstavlja operaciju pisanja, a logička jedinica operaciju čitanja. Kada je adresa poslata, svaki uređaj u sistemu upoređuje prvih sedam bita posle start komande sa svojom adresom. Ukoliko se poklapaju, slejv zna da master želi da komunicira sa njim na način koji zavisi od R/W bita. Adresa slejv uređaja može biti sačinjena od predefinisano i programabilnog dela. S obzirom da u sistemu mogu da se nađu više istih uređaja, programabilni deo adrese određuje maksimalni mogući broj tih uređaja koji se povezuju na magistralu. Broj programabilnih adresa zavisi od broja slobodnih pinova. Na primer, ukoliko uređaj ima 4 fiksna i 3 programabilna adresna bita, ukupno 8 identičnih uređaja može biti povezano.

I²C komisija određuje alokaciju adresa. Dve grupe od osam adresa (0000XXX i 1111XXX) su zauzete. Njihova uloga data je u tabeli na slici 8.4. Kombinacija bita slejv adrese 11110XX je zauzeta za 10-bitno adresiranje.

<i>Slave adresa</i>	R/ bit	Opis
0000 000	0	Generalna adresa poziva
0000 000	1	START bajt
0000 001	x	CBUS adresa
0000 010	x	Adresa rezervisana za različite formate magistrala
0000 011	x	Rezervisano za buduće potrebe
0000 1xx	x	
1111 1xx	x	
1111 0xx	x	10-bit <i>slave</i> adresiranje

Slika 8.4. Uloga zauzetih adresa u I²C komunikaciji

General call adresa služi za adresiranje svakog uređaja na I²C magistrali. Ukoliko podatak koji se šalje nije potreban nekom slejv uređaju, on ga može jednostavno ignorisati tako što neće generisati ACK signal.

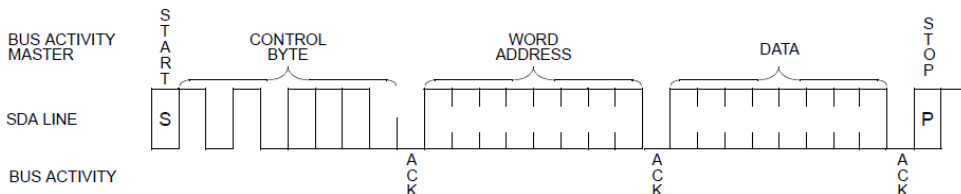
8.3. I²C komunikacija sa memorijom 24C04A

Integrirano kolo 24C04A predstavlja serijski EEPROM od 4Kbit-a, a organizovan je interno kao 512 x 8 bita memorije. U kontrolnom bajtu gornja četiri bita su rezervisana, dok donja četiri služe za adresu i komandu za čitanje ili upis što je prikazano u tabeli 8.1.

Tabela 8.1.

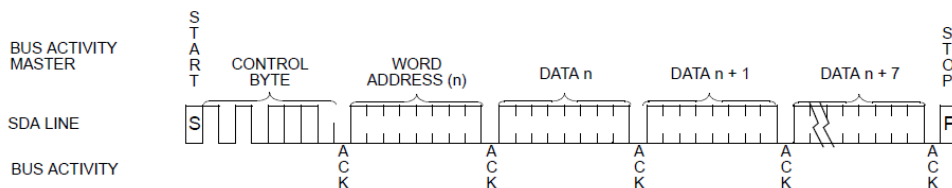
1	0	1	0	A2	A1	A0	R/W
---	---	---	---	----	----	----	-----

Na osnovu tri adresna bita 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.



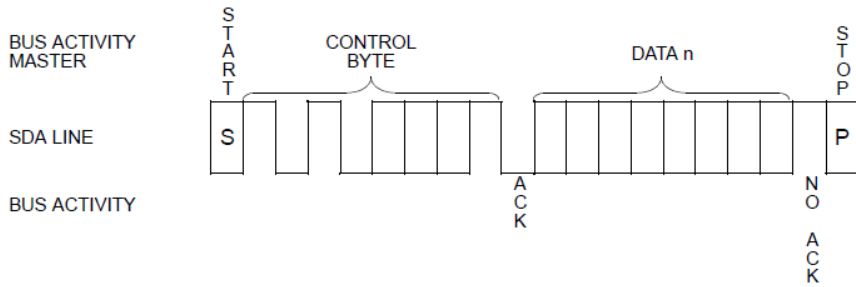
Slika 8.5. Dijagram upisa bajta [23]

Razlika između slanja jednog bajta i niza bajtova je u tome što se posle ACK za prvi bajt nastavlja komunikacija.



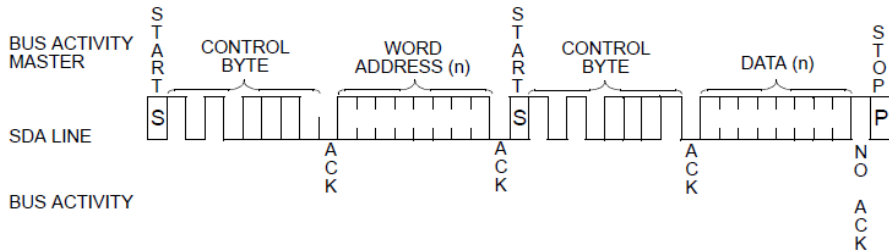
Slika 8.6. Dijagram upisa niza bajtova [23]

Na slici 8.7 prikazana je procedura čitanja sa početne adrese u memoriji, pa se iz tog razloga ne šalje adresa. Kad se podatak pročita, adresni brojač se automatski uveća za jedan tako da automatski pokazuje na sledeću memorijsku lokaciju. Master na kraju čitanja neće proslediti ACK, ali će generisati stop bit.



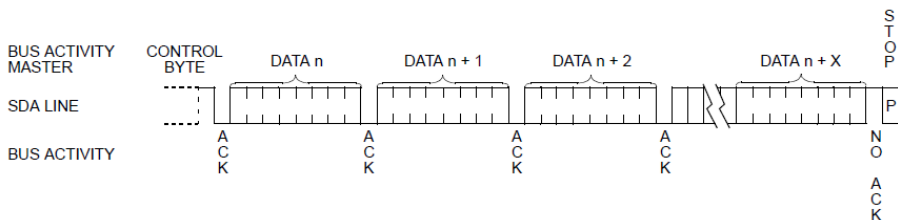
Slika 8.7. Čitanje bajta sa početne adrese [23]

Kod čitanja sa proizvoljne adrese prvo se upiše adresa sa koje se čita pa se ponovo pošalje start bit i kontrolni bajt i tek onda se dobija željeni podatak što je pokazano na slici 8.8.



Slika 8.8. Čitanje bajta sa proizvoljne adrese [23]

Kod čitanja niza bajtova sa proizvoljne adrese procedura je ista kao i kod čitanja jednog bajta, s tim da se nastavlja čitanje podataka postavljanjem ACK signala, osim posle poslednjeg čitanja. Ovo je prikazano na slici 8.9.

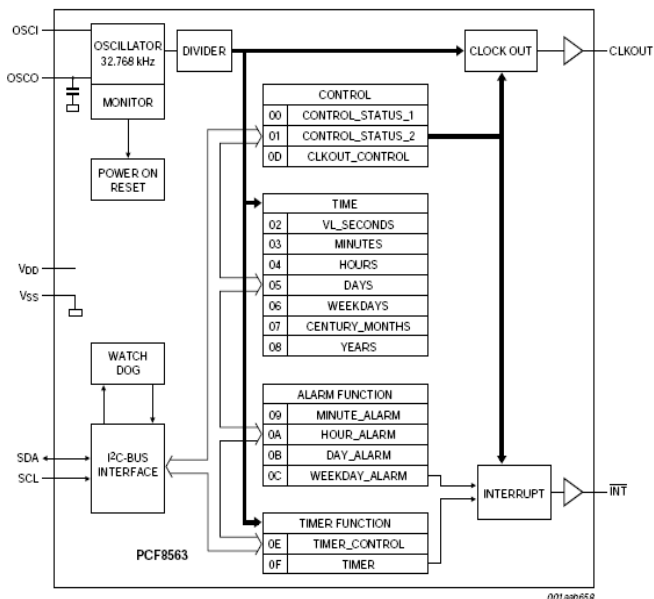


Slika 8.9. Čitanje niza bajtova sa proizvoljne adrese [23]

8.4. PCF8563 kolo

Integrirano kolo PCF8563 je CMOS sat/kalendar realnog vremena optimizovan za aplikacije gde je važno obezbediti nisku potrošnju. Komunikacija se odvija preko I²C magistrale. Maksimalna brzina je 400 kb/s. Ugrađeni *word* adresni registar se uvećava automatski nakon svakog pročitanoog ili upisanog bajta podataka. PCF8536 daje podatke

o godini, mesecu, danu, danu u nedelji, satima, minutima i sekundama na bazi 32.768 kHz kvarcnog kristala. Sve vrednosti su kodovane u BCD formatu. Posедуje alarmne i tajmerske funkcije. Adrese slejv uređaja na I²C magistrali su za čitanje A3h, a za upis A2h. PCF8563 radi kao slejv uređaj i ima fiksnu adresu 51h. Blok šema je data na slici 8.10.



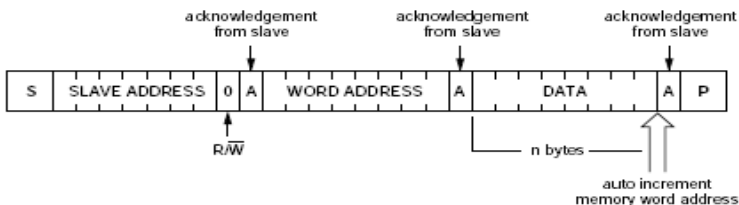
Slika 8.10. Blok dijagram PCF8563 [23]

PCF8563 ima 16 (adrese od 00h do 0Fh) 8-bitnih registara koji su prikazani na slici 8.11. Pojedini biti u nekim od registara nisu implementirani.

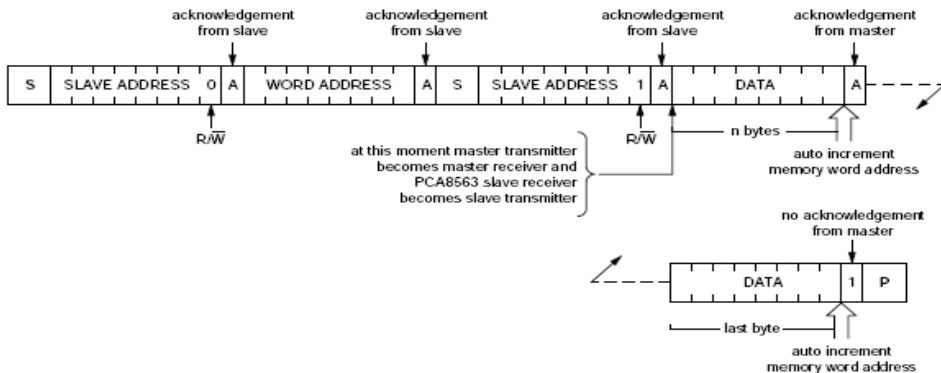
Address	Register name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
00h	control_status_1	TEST1	0	STOP	0	TESTC	0	0	0
01h	control_status_2	0	0	0	TI_TP	AF	TF	AIE	TIE
02h	VL_seconds	VL	<seconds 00 to 59 coded in BCD>						
03h	minutes	x	<minutes 00 to 59 coded in BCD>						
04h	hours	x	x	<hours 00 to 23 coded in BCD>					
05h	days	x	x	<days 01 to 31 coded in BCD>					
06h	weekdays	x	x	x	x	x	<weekdays 0 to 6 in BCD>		
07h	century_months	C	x	x	<months 01 to 12 coded in BCD>				
08h	years	<years 00 to 99 coded in BCD>							
09h	minute_alarm	AE	<minute alarm 00 to 59 coded in BCD>						
0Ah	hour_alarm	AE	x	<hour alarm 00 to 23 coded in BCD>					
0Bh	day_alarm	AE	x	<day alarm 01 to 31 coded in BCD>					
0Ch	weekday_alarm	AE	x	x	x	x	<weekday alarm 0 to 6 in BCD>		
0Dh	CLKOUT_control	FE	x	x	x	x	x	FD1	FD0
0Eh	timer_control	TE	x	x	x	x	x	TD1	TDO
0Fh	timer	<timer countdown value>							

Slika 8.11. Registri PCF8563 [23]

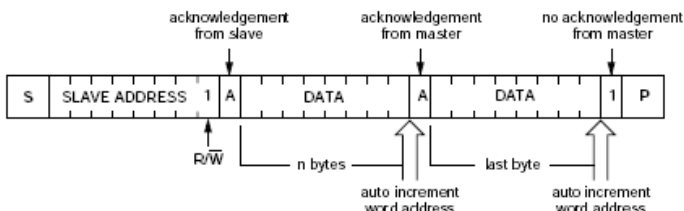
Ciklus čitanja i upisa je prikazan na slikama 8.12, 8.13 i 8.14. *Word* adresa je 4-bitna i određuje registar sa kojim se radi. Gornja 4 bita adrese se ne koriste. Primer jedne primene ovog integrisanog kola je dato na slici 8.15.



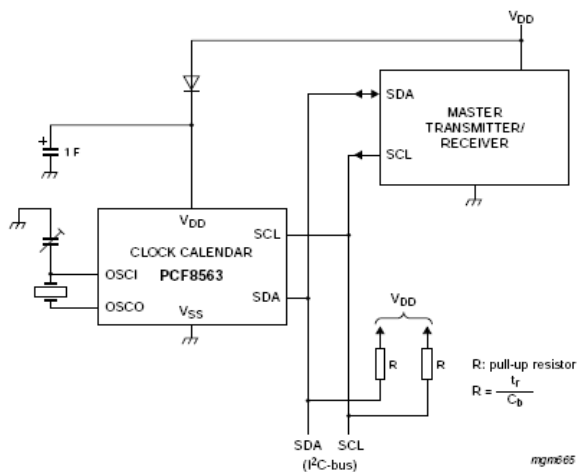
Slika 8.12. Master šalje podatke (write mod) [23]



Slika 8.13. Master čita nakon postavljanja word adrese [23]



Slika 8.14. Master čita neposredno nakon prvog bajta (read mod) [23]



Slika 8.15. Primer aplikacije [23]

8.5. Podešavanje I²C na RPi

Prvo je potrebno aktivirati I²C magistralu pozivom konfiguracionog alata:

```
sudo raspi-config
```

U *Interfacing options*, odabrati I²C. Potom odabrati da interfejs bude aktivan kao i da drajver (modul) bude učitani biranjem *Yes*, *Ok* i *Finish*.

Nakon toga podesiti učitavanje modula pozivom:

```
sudo nano /etc/modules
```

i dodavanjem sledeća dva reda:

```
i2c-bcm2708
i2c-dev
```

Dalje instalirati i2c alate sa:

```
sudo apt-get install i2c-tools
```

Restartovati RPi sa: `sudo reboot`

Proveriti da li je učitani i2c modul sa:

```
lsmod | grep i2c_
```

što bi trebalo da ispiše tekst koji sadrži `i2c_bcm2708`.

Dalje proveriti hardversku adresu priključenog RTC-a pozivom:

```
sudo i2cdetect -y 1
```


što bi trebalo da da tabelu u kojoj se na adresi 0x51 pojavljuje oznaka da postoji neki uređaj. Ovo je prikazano na slici 8.16.

```
pi@raspberrypi ~ $ sudo i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  51  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

Slika 8.16. Adrese I²C uređaja

8.6. Primer rada sa PCF8563 upotrebom BCM2835 biblioteke

U narednom primeru je data realizacija ispisa postavljenog vremena upotrebom BCM2835 biblioteke tako da radi ispis tekućeg vremena u terminalu na svakih 5 sekundi. Nakon prevođenja, primer pozivati sa sudo.

```
// primer rada sa PCF8563 koji ispisuje tekuće vreme
// u terminalu svakih 5 sekundi

#include <bcm2835.h>
#include <stdio.h>
#include <unistd.h>

#define changeHexToInt(hex)      (((hex)>>4) *10 ) +
                                  ((hex)%16) )

//adrese registara
#define SEK      0x02
#define MIN      0x03
#define SAT      0x04

unsigned char WriteBuf[2];
unsigned char ReadBuf;
unsigned char g8563_Store[3];
// sec,min,sat
unsigned char init8563_Store[3]={0x00,0x59,0x08};
```

```

void P8563_settime()
{
    WriteBuf[0] = SEK;
    WriteBuf[1] = g8563_Store[0];
    bcm2835_i2c_write(WriteBuf,2);

    WriteBuf[0] = MIN;
    WriteBuf[1] = g8563_Store[1];
    bcm2835_i2c_write(WriteBuf,2);

    WriteBuf[0] = SAT;
    WriteBuf[1] = g8563_Store[2];
    bcm2835_i2c_write(WriteBuf,2);
}

void P8563_init()
{
    unsigned char i;

    for(i=0;i<=3;i++)
        g8563_Store[i]=init8563_Store[i];

    P8563_settime();
    printf("Postavi početno tekuće vreme\n");

    //inicijalizacija RTC-a
    WriteBuf[0] = 0x0;
    WriteBuf[1] = 0x00; //normalni rezim rada
    bcm2835_i2c_write(WriteBuf,2);
}

void P8563_Readtime()
{
    unsigned char time[7];

    WriteBuf[0] = SEK;
    bcm2835_i2c_write_read_rs(WriteBuf, 1, time, 7);

    g8563_Store[0] = time[0] & 0x7f;
    g8563_Store[1] = time[1] & 0x7f;
    g8563_Store[2] = time[2] & 0x3f;

    g8563_Store[0] = changeHexToInt(g8563_Store[0]);
    g8563_Store[1] = changeHexToInt(g8563_Store[1]);
    g8563_Store[2] = changeHexToInt(g8563_Store[2]);
}

```

```

int main(int argc, char **argv)
{
    if (!bcm2835_init())
        return 1;

    bcm2835_i2c_begin();

    // adresa PCF8563 na I2C magistrali
    bcm2835_i2c_setSlaveAddress(0x51);
    bcm2835_i2c_set_baudrate(10000);
    printf("start.....\n");
    P8563_init() ;

    while(1)
    {
        P8563_Readtime();

        printf("Sati:%d Minuti:%d Sekunde:%d\n",
            g8563_Store[2], g8563_Store[1],
            g8563_Store[0]);

        bcm2835_delay(5000);
    }

    bcm2835_i2c_end();
    bcm2835_close();

    return 0;
}

```

8.7. Zadaci

8.7.1. Zadatak 1

Proširiti prethodni primer tako da se ispisuje i datum.

8.7.2. Zadatak 2

Prethodni primer uraditi pomoću *wiringPi* funkcija za rad sa I²C magistralom.

8.7.3. Zadatak 3

Povezati LCD sa RPi i napisati program koji ispisuje u prvom redu datum, a u drugom tekuće vreme.

8.7.4. Zadatak 4

Koristeći dodatne tastere (na DVK512 pločici ili na dodatnoj pločici) proširiti prethodni zadatak sa mogućnošću unosa datuma i tekućeg vremena. Jedan taster treba da inicira unos datuma ili vremena, drugi potvrđuje, a treći i četvrti služe za promenu vrednosti gore-dole.