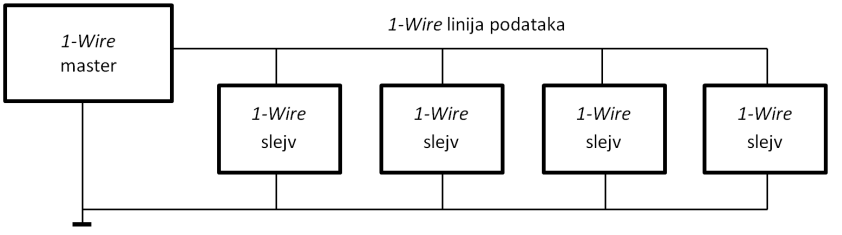
5

Merenje temperature senzorom DS18B20

U okviru ove vežbe ćemo se upoznati sa komunikacionim protokolom *1-Wire* kao i sa radom sa temperaturnim senzorom DS18B20.

5.1. Dallas *1-Wire* serijska komunikacija

Temperaturni senzor DS18B20 koristi *1-Wire* komunikaciju pa ćemo se prvo upoznati sa njom. *1-Wire* serijska komunikacija originalno poznata kao MicroLAN™, definisana od strane kompanije *Dallas Semiconductor* je komunikacioni sistem između električnih uređaja koji koristi jednu žičanu vezu. *1-Wire* se sastoji od tri osnovna elementa: *1-Wire bus master*, *slave* uređaji i električna veza između *master* i *slave* uređaja.



Slika 5.1. Izgled 1-Wire mreže sa jednim bus masterom i četiri slejva.

1-Wire komunikacija je slična starom telefonskom sistemu. U takvom sistemu jedna osoba poziva drugu. Osoba koja zove, tj. osoba koja inicira poziv je u stvari *1-Wire bus master*. Signal mora prvo da se rutira kroz prekidače u centrali odakle se rutira do određene linije. Ovo se može uraditi korišćenjem MicroLan habova u slučaju *1-Wire* sistema. Svi članovi linije primaju signal dolaznog poziva. Broj poziva, ili adresa, kaže koja osoba, tj. slejv uređaj, treba da se javi. Kao i u telefonskoj konverzaciji, jedna

osoba priča dok druga sluša, *bus master* govori slejvu šta želi od njega. Zahteva informaciju ili mu govori šta da radi.

1-Wire sistem komunicira digitalno preko kabla koji je obično upredena parica. Mreža je definisana kao *open drain*, ožičeno I kolo, master-slejv *multi drop* arhitektura koja koristi *pull-up* otpornike na nominalnih 5 V koji su ujedno napajanje mastera. Master uređaj inicira i kontroliše sve aktivnosti na mrežnoj magistrali. *Bus* master se ponaša kao protokol i tajming interfejs između personalnog računara ili mikrokontrolera i *1-Wire* mreže. Ovo su uređaji koji mogu da šalju i primaju podatke na jednoj liniji podataka. Podaci se mogu slati samo kao polu-dupleks. Prenos podataka je bit po bit, gde se najmanje značajni bit prenosi prvi. Sinhronizacija se obavlja pomoću *1-Wire* protokola. Ovo omogućava kontrolu magistrale i samo master može da inicira komunikaciju.

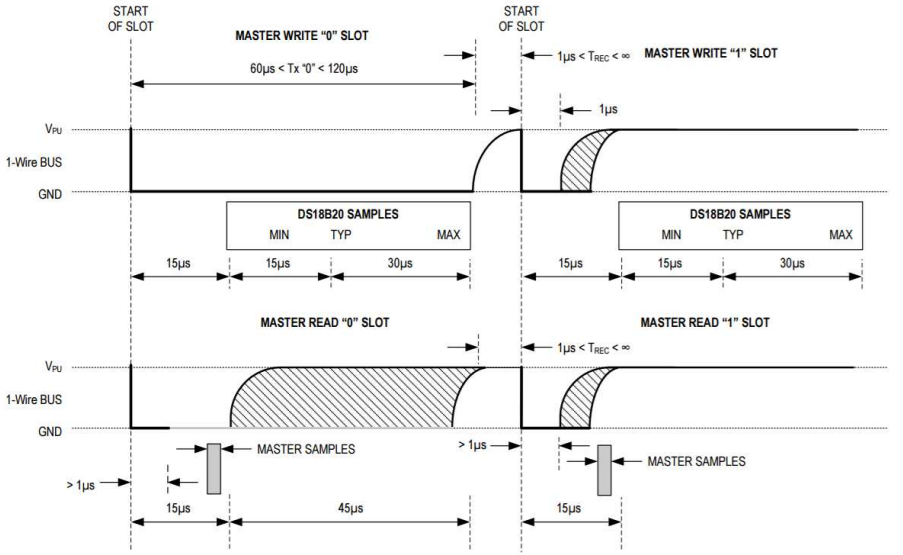
1-Wire logički nivoi su grubo kompatibilni sa konvencionalnim CMOS/TTL logičkim nivoima, sa približno 0,8 V za logičku „0“ i minimum 2,2 V za logičku „1“. Oblast između ova dva napona je nedefinisana.

U stanju mirovanja magistrala se nalazi na logičkoj jedinici zbog veze preko *pull-up* otpornika. Zbog toga svi uređaji moraju biti u stanju da spuste magistralu na nulu i to je moguće samo ako kola poseduju *open drain* ili *open collector* interfejs.

Signali u *1-Wire* komunikaciji su podeljeni u vremenske slotove (eng. *time slots*) od po 60 µs. Jedan bit podatka se prenosi u jednom vremenskom slotu. Slejv uređaji mogu imati vremensku bazu, po kojoj određuju prijem podataka, koja se značajno razlikuje od nominalne vremenske baze. Doduše ovo zahteva da master ima jako precizan tajming, kako bi omogućio ispravnu komunikaciju sa slejv uređajima sa drugačijom vremenskom bazom.

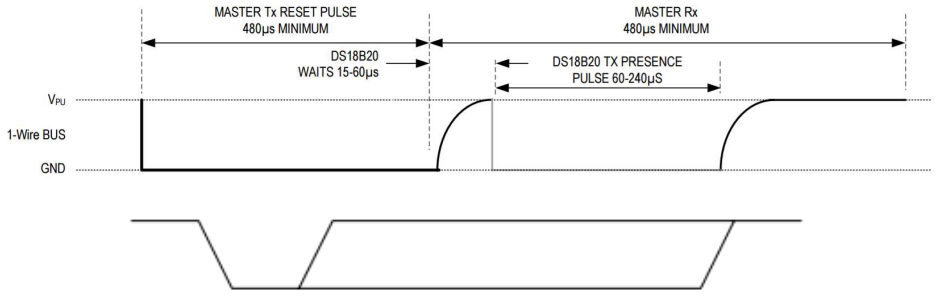
Master inicira svaku komunikaciju na magistrali tako što povlači magistralu na logičku nulu i na taj način vrši sinhronizaciju tajming logike svih uređaja. Postoje ukupno pet osnovnih komandi za komunikaciju na *1-Wire* magistrali i to su: “**write 1**”, “**write 0**”, “**read**”, “**reset**” i “**presence**”.

Ukoliko master želi da obavi „write 1“ operaciju, onda on postavlja magistralu na nisku vrednost u periodu od 1 do 15 µs. Za operaciju „write 0“, master povlači magistralu na nisku vrednost u periodu od 60 µs do maksimalnih 120 µs. Da bi master aktivirao „read“ signal potrebno je da postavi magistralu na nisku vrednost u periodu od 1 do 15 µs. Slejv uređaj odgovara tako što drži magistralu na niskoj vrednosti za logičku nulu, ili jednostavno oslobađa liniju ukoliko ukoliko želi da pošalje logiku „1“. Magistrala treba da bude semplovana 15 µs posle spuštanja na nisku vrednost. Pošto se „write 1“ i „read“ signali definišu na isti način, slejv sam određuje koja je komanda u pitanju.



Slika 5.2 . Izgled „write“ i „read “ signala [16]

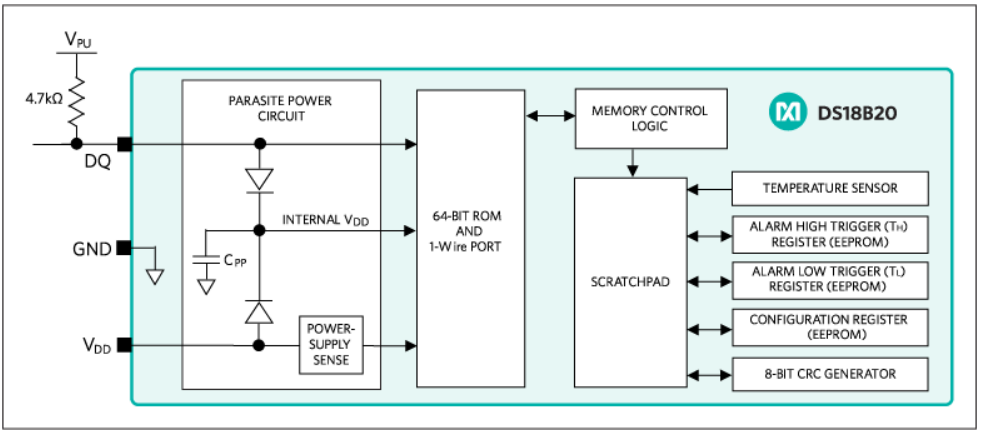
„Reset“ i „presence“ signali prikazani su na slici 5.3. Važno je istaći da je vremenska skala drugačija od prethodnih signala. Master povlači magistralu na nisku vrednost za vremenski period od barem 8 vremenskih slotova, ili za 480 µs. Ovaj vremenski period inicira „reset” signal. Slejv bi onda trebao da povuče magistralu na nisku vrednost u periodu od 60 µs posle „reset“ signala. Ta niska vrednost treba da traje barem 60 µs. Ovaj odziv se naziva „presence” signal. Ukoliko nema „presence“ signala na magistrali, master mora zaključiti da nema slejv uređaja na magistrali.



Slika 5.3. Izgled „reset“ i „presence“ signala [16]

5.2. Temperaturni senzor DS18B20

DS18B20 digitalni temperaturni senzor ima 9-12 bitnu rezoluciju merenja temperature u Celzijusima i alarmnu funkciju koju korisnik može da podesi (i gornji i donji nivo). DS18B20 komunicira sa mikrokontrolerom kao masterom preko *1-Wire* magistrale koja zahteva samo jednu liniju za prenos podataka (i *gnd*). Dodatno, DS18B20 ne mora da ima spoljašnje napajanje, već može da se napaja i preko linije za podatke ("parazitira").



Slika 5.4. Blok šema DS18B20 [16]

Svaki DS18B20 ima jedinstven 64-bit serijski kod, koji omogućava korišćenje više senzora na istoj magistrali. Na taj način moguće je jednim procesorom upravljati sa puno senzora distribuirano na velikom prostoru. Primeri primene ovih senzora su: upravljanje grejanjem i klimatizacijom, monitoring temperature unutar zgrada, opreme, mašina i u upravljačkim i procesnim sistemima.

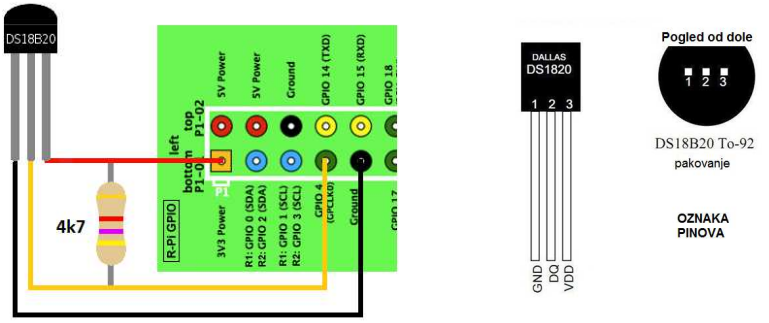
Ključna svojstva:

- Temperaturni opseg je -55°C do +125°C
- Tačnost: ±0.5°C (u opsegu -10°C do +85°C)
- Rezolucija se može podesiti u opsegu od 9 -12 bita
- Nisu potrebne spoljašnje komponente
- U modu parazitnog napajanja je dovoljno 2 pina za rad (DQ i GND)
- Svaki senzor ima jedinstveni 64-bit serijski kod u internoj ROM
- Fleksibilno podešavanje alarma

5.3. Rad sa senzorom DS18B20

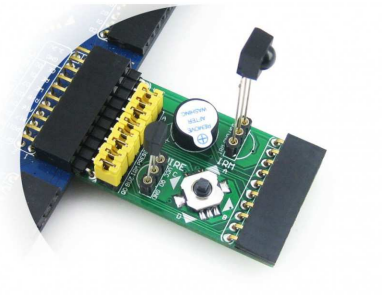
Pre početka rada sa temperaturnim senzorom potrebno je povezati ga sa RPi na jedan od sledeća dva načina:

1) prema narednoj slici koristeći protobord i odgovarajuće kabliće za povezivanje



Slika 5.5. Povezivanje DS18B20 sa RPi

2) koristeći DVK512 pločicu sa *MixBoard* dodatnom pločicom [17] postavljenom u konektor sa oznakom 8I/Os kao i senzorom postavljenim u *1-Wire* 3-pinski priključak na *Mixboard* pločici.



Slika 5.6. Mixboard dodatna pločica sa priključenim DS18B20 i IR senzorom

Da bi se moglo raditi sa temperaturnim senzorom DS18B20 koji koristi *1-Wire* magistralu potrebno je uraditi izmenu u `/boot/config.txt` datoteci na sledeći način:

```
sudo nano /boot/config.txt
```

Na kraju datoteke dodati liniju:

```
dtoverlay=w1-gpio,gpiopin=4
```

i sačuvati promene. Ova promena je neophodna jer novije verzije Raspbian OS koriste *overlay* metodu za učitavanje drajvera koji koriste određene delove hardvera na višestruke načine. Konkretno pin 4, koji je inače GPIO pin, ovde će biti korišćen kao *1-*

Wire pin. Detalji oko mehanizama rada *Device Tree* koncepta i *overlay* metoda mogu se videti na [18].

Ranije verzije Raspbiana i Raspberry Pi-a su koristile sledeći metod pri radu sa temperaturnim senzorom.

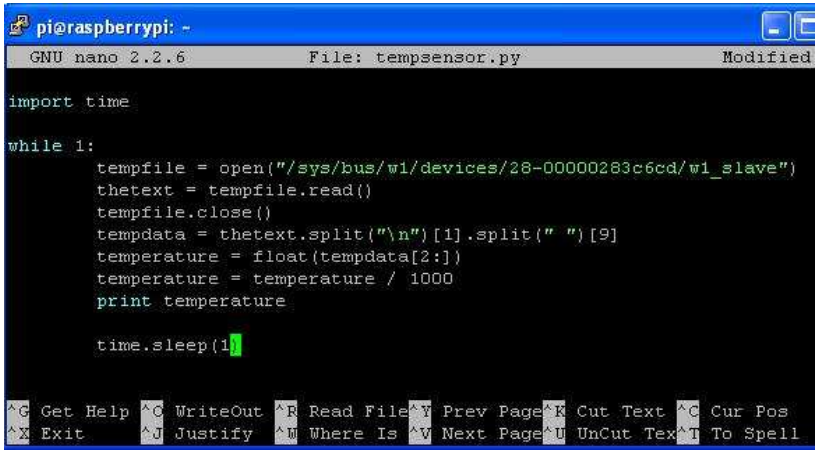
Prvo su učitani drajveri za *1-Wire* na sledeći način:

```
sudo modprobe w1-gpio
sudo modprobe w1_therm
```

Nakon toga se u `/sys/bus/w1/devices/` folderu pojavljuje direktorijum u formatu 28-00000xxxxxxx gde poslednjih 7 cifara (označene sa x) predstavljaju jedinstveni identifikacioni broj (eng. ID) senzora i temperatura se može očitati jednostavnim pozivom u tom direktorijumu: `cat w1_slave`

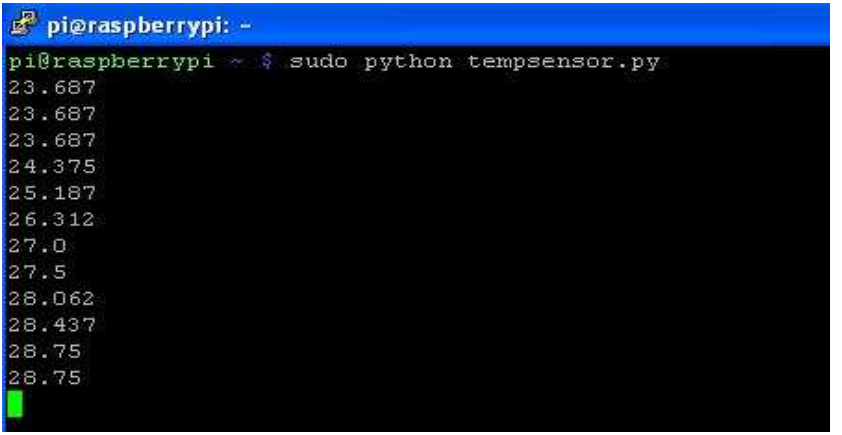
Temperatura je izlistana u 5-cifrenom formatu sa početnim "t=". Npr, t=22575 znači da je očitana temperatura 22.575 stepeni Celzijusa.

Primer Python programa koji na svaku sekundu izlistava temperaturu je dat na slici 5.7.



Slika 5.7. Python program za čitanje temperaturnog senzora

Nakon startovanja programa izgled ekrana u terminalu je kao na slici 5.8.



Slika 5.8. Rezultat programa za čitanje temperaturnog senzora

5.3.1. Program za rad sa senzorom u programskom jeziku C

Naredni program za rad sa temperaturnim senzorom koristi programski jezik C za realizaciju očitavanja temperature.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/fcntl.h>

int main(int argc, char *argv[])
{
     int fd = -1, ret;
     char *tmp1, tmp2[10], ch='t';
     char devname_head[50] =
     "/sys/devices/w1_bus_master1/28-00000";

     char devname_end[10] = "/w1_slave";
     char dev_name[100];
     long value;
     int integer, decimal;
     char buffer[100];
     int i,j;

     strcpy(dev_name, devname_head);
     strcat(dev_name, argv[1]);
     strcat(dev_name, devname_end);
```

```
if ((fd = open(dev_name, O_RDONLY)) < 0)
     {
          perror("Greška pri otvaranju!");
          exit(1);
     }

     ret = read(fd, buffer, sizeof(buffer));

     if (ret < 0)
     {
          perror("Greška pri čitanju!");
          exit(1);
     }

     tmp1 = strchr(buffer, ch);
     sscanf(tmp1, "t=%s", tmp2);
     value = atoi(tmp2);
     integer = value / 1000;
     decimal = value % 1000;

     printf("Temperatura je %d.%d\n", integer,decimal);

     close(fd);
     return 0;
}
```

Program kompajlirati sa:

```
gcc -o ds18b20 ds18b20.c
```

i pokrenuti ga sa: `./ds18b20 xxxxxxx`
gde je xxxxxxx jedinstveni ID senzora o kome je bilo reči ranije.

5.3.2. Očitavanje temperature i beleženje u log datoteku

Sledeći primer omogućava čitanje temperature i logovanje izmerenih vrednosti u datoteku. Karaktere označene sa ??????? zameniti sa odgovarajućim ID brojem za konkretni senzor.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

double temperatura(void)  //očitavanje temperature
{
```

```
      FILE *ft;
      char tekst[100];
      ft=fopen("/sys/bus/w1/devices/28-00000???????/w1_slave","r");

      if(ft==NULL) return 0;

      int i=0;
      for(i=0;i<22;i++)               //samo temperatura
      fscanf(ft,"%s", tekst);

      fclose(ft);

      //obrisati „t=”
      for(i=0;i<10;i++) tekst[i]=tekst[i+2];

      int temp=atoi(tekst);           //prebaci u double
      double tem=(double)temp/1000;

      return tem;
};

void zapis(double temp)         //log metod
{
      FILE * log;

      log=fopen("log","a+");    //otvoriti datoteku
      if(log==NULL) return;

      fprintf(log, "\n%.3f", temp);
      fclose(log);
};

int main(void)
{
      FILE * log;
      int cz=0, j=0;
      struct timespec ts1, ts2;       //merenje vremena
      long czas;

      log=fopen("log","w");

      if(log==NULL) return 0;


      fprintf(log, "\n\t\t***Temperatura***\n
                Izmerena temperatura = %.3f \xC2\xB0 C\t C\n",
                temperatura());

      fclose(log);
```

```
printf("\n\nPočetna Temp = %.3f \xC2\xB0 C", temperatura());

     while(1)
     {
          j++;                        //broj iteracija

          clock_gettime(CLOCK_REALTIME, &ts1);

          printf("\nMerenje br. %d.\tTemp = %.3f
                    \xC2\xB0 C", j, temperatura());

     clock_gettime(CLOCK_REALTIME, &ts2);

     //trajanje pojedinacnog //merenja
     czas = (ts2.tv_nsec - ts1.tv_nsec)/1000000;

     if(czas<=0) czas=czas+999;

     /* Prethodni red koriguje povremenu gresku pri merenju
     kada se dobija rezultat koji je manji za 999ms u odnosu
     na stvarni */

     printf("\tTrajanje merenja: %ld ms.", czas);

     zapis(temperatura());     //upis u datoteku
     }
return 0;
}
```

5.3.3. Zadatak 1

Napisati program u programskom jeziku C koji radi očitavanje vrednosti temperaturnog senzora i ukoliko se dostigne 25 stepeni Celzijusa uključuje LED diodu koja trepće simbolišući alarm. Nakon smanjenja temperature ispod 25 stepeni Celzijusa isključiti diodu.

5.3.4. Zadatak 2

Napisati program u programskom jeziku C koji ima mogućnost podešavanja praga temperaturnog alarma i nakon toga radi očitavanje vrednosti temperaturnog senzora. Ukoliko se dostigne podešen prag alarma uključuje se LED dioda koja trepće simbolišući alarm. Nakon smanjenja temperature ispod praga isključiti diodu.

5.3.5. Primer projekta - temperaturni *loger/viewer*

Napisati program u programskom jeziku po izboru koji omogućava logovanje temperaturnih podataka u datoteku na svaku sekundu i koji grafički prikazuje promenu temperature u određenom vremenskom intervalu (*real-time* promene i *history* promene iz datoteke). Omogućiti donji i gornji temperaturni prag. Podešavanje pragova izvršiti tasterima na DVK512 pločici prikazom na LCD. Pored pragova, na displeju treba da bude ispisana i trenutna vrednost temperature, a ako je temperatura u alarmnoj zoni onda treba LED0 na DVK512 da trepće.