



Aleš Hojak

MIKROKRMILNIŠKI REGULATOR TEMPERATURE Z DALJINSKIM NADZOROM

Diplomska naloga

Maribor, maj 2009

Diplomsko delo univerzitetnega študijskega programa

MIKROKRMILNIŠKI REGULATOR TEMPERATURE Z DALJINSKIM NADZOROM

Študent: Aleš Hojak
Študijski program: UN ŠP - Elektrotehnika
Smer: Telekomunikacije
Mentor: doc.dr.Mitja Solar
Somentor: doc.dr Iztok Kramberger

Maribor, maj 2009



FERI

Fakulteta za elektrotehniko,
računalništvo in informatiko

Smetanova ulica 17
2000 Maribor

Številka: TK-30

Datum in kraj: 11. 05. 2009, Maribor

Na osnovi 330. člena Statuta Univerze v Mariboru (Ur. l. RS, št. 90/2008)

SKLEP O DIPLOMSKEM DELU

1. **Alešu Hojaku**, študentu univerzitetnega študijskega programa Telekomunikacije, se dovoljuje izdelati diplomsko delo pri predmetu Meritve v elektroniki.
2. **MENTOR:** doc. dr. Mitja Solar
SOMENTOR: doc. dr. Iztok Kramberger
3. **Naslov diplomskega dela:**
MIKROKRMILNIŠKI REGULATOR TEMPERATURE Z DALJINSKIM NADZOROM
4. **Naslov diplomskega dela v angleškem jeziku:**
TEMPERATURE REMOTE CONTROL WITH MICROCONTROLLER
5. Diplomsko delo je potrebno izdelati skladno z "Navodili za izdelavo diplomskega dela" in ga oddati v treh izvodih ter en izvod elektronske verzije do 11. 05. 2010 v referatu za študentske zadeve.

Pravni pouk: Zoper ta sklep je možna pritožba na senat članice v roku 3 delovnih dni.



Obvestiti:

- kandidatko,
- mentorja,
- somentorja,
- odložiti arhiv.

ZAHVALA

Zahvaljujem se mentorju za pomoč in vodenje pri opravljanju diplomskega dela. Prav tako se zahvaljujem somentorju.

MIKROKRMILNIŠKI REGULATOR TEMPERATURE Z DALJINSKIM NADZOROM

Ključne besede: temperatura, rele, I²C, zaporedna komunikacija, mikrokrmilnik

UDK: 681.536.5:621.398(043.2)

Povzetek

Vsebina diplomskega dela obsega načrtovanje strojne in programske opreme za prototip izdelka. Opisujemo uporabljeni mikrokrmilnik, krmilnik svetlečih diod in pretvornik A/D, ki so najvažnejši gradniki prototipa izdelka. V nadaljevanju podajamo nekaj osnovnih nastavitvev programskega razvojnega okolja Bascom-8051 ter princip delovanja programa s pomočjo odločitvenega diagrama. Podani so tudi izračuni in načrti za električno vezje.

TEMPERATURE REMOTE CONTROL WITH MICROCONTROLLER

Key words: temperatura, rele, I²C, zaporedna komunikacija, mikrokrmilnik

UDK: 681.536.5:621.398(043.2)

Abstract

The content of this diploma work includes hardware and software description for circuit prototype. We also describe the used microcontroller, LED driver and a A/D converter, which are the most important components of this circuit prototype. We give the description of some basic settings of Bascom-8051 integrated development environment and principles of our software.. We present calculations and circuit diagrams for circuit prototype.

UPORABLJENI SIMBOLI	VIII
UPORABLJENE KRATICE	IX
1. UVOD	1
1.1 Opredelitev oziroma opis problema, ki je predmet raziskovanja	1
1.2 Namen in cilj diplomske naloge	1
1.3 Metode raziskovanja	1
1.4 Vsebina	2
2. OPIS SISTEMA	3
3. VEZJE IN IZRAČUNI	5
3.1 Prototip vezja s krmilnikom diod	5
3.2 Prototip vezja za komunikacijo, vzorčenje in vklop releja.	7
3.3 Določitev potrebnih napajalnih napetosti za vezje.	12
4. PRETVORNIK A/D	13
4.1 Splošni opis	13
4.2 Pretvorba A/D	13
4.3 Naslavljanje naprave	14
4.4 Nadzorni zlog naprave	14
5. KRMILNIK DIOD LED	16
5.1 Splošen opis krmilnika diod LED	16
5.3 Prenos podatkov	19
5.4 Registri znakov	22
5.5 Nastavitveni register	24
5.6 Tabela ASCII znakov	25
5.7 Register svetilnosti	25
6. MIKROKRMILNIK	27
6.1 Splošni opis mikrokrmilnika	27
6.2 Naslavljanje	28
6.3 Pomnilniki mikrokrmilnika	30
6.4 Vrata	31
6.6 EEPROM	33
7. KOMUNIKACIJA RS 232C	36

8. PROGRAMIRANJE MIKROKRMILNIKA	39
8.1 Razvojno okolje	39
8.2 Nastavitve mikrokrmilnika v razvojnem okolju	41
8.3 Programska programa	45
9. REZULTATI	47
10. SKLEP	50

UPORABLJENI SIMBOLI

$C_{\text{PARAZITNA}}$ - parazitna kapacitivnost
 C_{SET} - kapacitivnost za nastavljanje frekvence notranjega oscilatorja
 f_{OSC} - frekvenca oscilatorja
 I_{ADJ} – nastavitveni tok
 I_{SEG} - segmentni tok
 I – skupni tok
 I_1 - tok v veji 1
 I_2 – tok v veji 2
 K_F - konstanta za izračun frekvence oscilatorja
 K_I - konstanta za izračun segmentnega toka
 k_T – temperaturni faktor
 R_{SET} - upornost za nastavljanje frekvence notranjega oscilatorja
 R_T - upornost pri temperaturi
 R_{25} - upornost pri temperaturi 25°C
 R_1, R_2, R_3, R_4 – upori
 T – temperatura
 T_A – temperature
 U - napetost
 U_0 - napetost v veji 2 na uporu R_4
 U_1 - napetost v veji 1 na uporu R_3
 V_{AGND} – napetost analogne mase za pretvornika A/D
 $V_{\text{AIN+}}$ – pozitivna napetost za diferenčni vhod pretvornika A/D
 $V_{\text{AIN-}}$ – negativna napetost za diferenčni vhod pretvornika A/D
 V_{IN} - vhodna napetost
 V_{LSB} – napetost stopnice
 V_{OUT} - izhodna napetost
 V_{REF} – referenčna napetost za pretvornik A/D

UPORABLJENE KRATICE

kratica	pomen kratice	prevod ali razlaga
ADC	Analog to Digital Conversion	Analogno digitalna pretvorba
AGND	Analog GrouND	analogna masa
AIN	Analog Input	analogni vhod
AOUT	Analog OUTput	analogni izhod
ACALL	Absolute jump to subroutine	absolutni skok v podprogram
BRGCON	Baud Rate Generator CONtrol	nadzorni register generatorja prenosne hitrosti
BRGR0	Baud Rate Generator Rate Low	spodnji zlog generatorja za določanje prenosne hitrosti
BRGR1	Baud Rate Generator Rate High	zgornji zlog generatorja za določanje prenosne hitrosti
CCU	Capture Compare Unit	lovilno primerjalna enota
CMOS	Complementary Metal Oxide Semiconductor	tehnika izdelave polprevodniških elementov
CPE	Centralna Procesna Enota	-
DAC	Digital to Analog Conversion	Digitalno analogna pretvorba
DEEADR	Data EEPROM ADdRess register	naslovni register podatkovnega EEPROM-a
DEECON	Data EEPROM CONtrol register	kontrolni register podatkovnega EEPROM-a
DEEDAT	Data EEPROM DATA register	podatkovni register podatkovnega EEPROM-a
DPTR	Data PoinTeR	Podatkovni kazalec
EA	EnAble	bit registra IEN0
EADR	EEPROM ADdRess	MSB naslova EEPROM-a
ECTL	EEPROM ConTroL	bit kontrolnega registra EEPROM-a
EEIF	EEPROM Interrupt Flag	prekinitvena zastavica EEPROM-a
EEPROM	Electrically Erasable Programmable Read Only Memory	programirljiv, električno zbrisljiv bralni pomnilnik
F	Fast	hiter
FE	Framing Error flag	zastavica za napako okvirja
GND	GrouND	masa
HEX	HEXadecimal	heksadecimalno
HS	High Speed	visoka hitrost
HVERR	High Voltage ERRor	napaka visoke napetosti
IEN0	Interrupt ENable 0	prekinitveni register 0
IEN1	Interrupt ENable 1	prekinitveni register 1
IP0	Interrupt Priority 0	spodnji zlog prekinitvene prioritete 0
IP0H	Interrupt Priority 0 High	zgornji zlog prekinitvene prioritete 0
IP1	Interrupt Priority 1	spodnji zlog prekinitvene prioritete 1

IP1H	Interrupt Priority 1 High	zgornji zlog prekinitvene prioritete 1
IRAM	Internal RAM	notranji RAM
ISR	Interrupt Service Routine	prekinitvena rutina
I2C	Integrated2Circuit	protokol za serijsko komunikacijo
I/O	Input/Output	vhod/izhod
LCD	Liquid Crystal Display	prikazovalnik s tekočimi kristali
LCALL	Long jump to subroutine	dolg skok v podprogram
LED	Light Emitting Diode	svetleča dioda
LJMP	Long JuMP	dolgi skok
LSB	Less Significant Bit	bit z najmanjšo težo
MOV	MOVe	premakni
MOVC	MOVe Code byte	premakni programski zlog
MOVX	MOVe eXternal RAM	premakni zunanji RAM
MSB	Most Significant Bit	bit z največjo težo
OSC	OSCillator	oscilator
PCON	Power CONtrol Register	nadzorni register za obratovanje mikrokrmilnika
RAM	Random Access Memory	pomnilnik z naključnim dostopom
RB8	Recived Bit 8	bit nadzornega registra SCON
REN	Reception ENable	omogoči sprejem
RI	Receive Interrupt flag	sprejemna prekinitvena zastavica
ROM	Read Only Memory	bralni pomnilnik
RST	ReSeT	ponastavitev
RS232	Recommended Standard 232	priporočen standard 232 za serijsko komunikacijo
RxD	Receive data	sprejem podatkov
R/W	Read/Write	branje/pisanje
S	Standard	standardno
SADDR	Serial port ADDRESS Register	naslovni register serijskih vrat
SADEN	Serial port Address ENable	omogoči naslovni register serijskih vrat
SBUF	Serial port data BUFFer register	register serijskega podatkovnega pomnilnika serijskih vrat
SCL	Serial Clock Line	serijska urina linija
SCON	Serial port CONtrol register	register za nadzor serijskih vrat
SDA	Serial Data Line	serijska podatkovna linija
SFR	Special Function Register	register za posebne funkcije
SM0	Select Mode 0	bit registra za nadzor serijskih vrat SCON
SM1	Select Mode 1	bit registra za nadzor serijskih vrat SCON
SM2	Select Mode 2	bit registra za nadzor serijskih vrat SCON
SMOD0	Select MODE 0	bit nadzornega registra PCON
SMOD1	Select MODE 1	bit nadzornega registra PCON
SP	Stack Pointer	kazalec sklada

SPI	Serial Peripheral Interface	zaporedni vmesnik za periferijo
SPTR	Stack Pointer Register	register kazalca sklada
SSTAT	Serial port extended STATUS register	statusni register serijskih vrat
TB8	Transmitted Bit 8	bit nadzornega registra SCON
TI	Transmit Interrupt flag	oddajna prekinitvena zastavica
TxD	Transmit Data	oddaja podatkov
UART	Universal Asynchronous Receiver / Transmitter	univerzalni asinhroni sprejemnik / oddajnik

1. UVOD

1.1 Opredelitev oziroma opis problema, ki je predmet raziskovanja

Diplomska naloga se dotika področij programiranja mikrokrmilnikov in načrtovanja vezij. Mikrokrmilniki se danes pojavljajo praktično v vseh vezjih, njihova prednost je vsestranskost, ki jo dosežemo s programiranjem. Mikrokrmilnik bomo uporabili za izdelavo prototipa vezja za črpanje segrete vode iz sončnega zbiralnika v vodni zalogovnik. Pogoj za vklop črpalke je razlika med merjenima temperaturama. Vezje želimo povezati z računalnikom, za izpis temperatur in vnos razlike med merjenima temperaturoma, ki sta pogoj za vklop črpalke.

1.2 Namen in cilj diplomske naloge

Namen diplomske naloge je izdelati prototip elektronskega vezja, ki vsebuje mikrokrmilnik. S pomočjo prototipa razložimo delovanje programa, in mikrokrmilnika ter ostalih elementov, ki jih vsebuje vezje.

Cilj je izdelati vezje za vklopjanje črpalk, ki bo ustrezal sledečemu opisu. Vezje naj ima štiri znakovni prikazovalnik. Prikazovati mora temperaturo v sončnem kolektorju in zalogovniku vode. Vezje mora omogočati nastavljanje temperaturne razlike, ki določa pogoj za vklapljanje črpalke. Imeti mora malo porabo energije ter vsebovati čim manjše število gradnikov.

1.3 Metode raziskovanja

Končni izdelek nastaja postopoma, načrtovanja se lotimo modularno s testiranjem strojne opreme s pomočjo programske opreme. Metode smo izvedli po naslednjem zaporedju:

- Načrtovanje in modularna izgradnja izdelka na preizkusni plošči.
- Preizkušanje različnih programskih rutin s pomočjo simulatorja, ki je del programskega paketa Bascom - 8051.
- Sestavljanje posameznih modulov električnega vezja z mikrokrmilnikom na preizkusni plošči ter testiranje posameznih programskih aplikacij, potrebnih za delovanje izdelka.

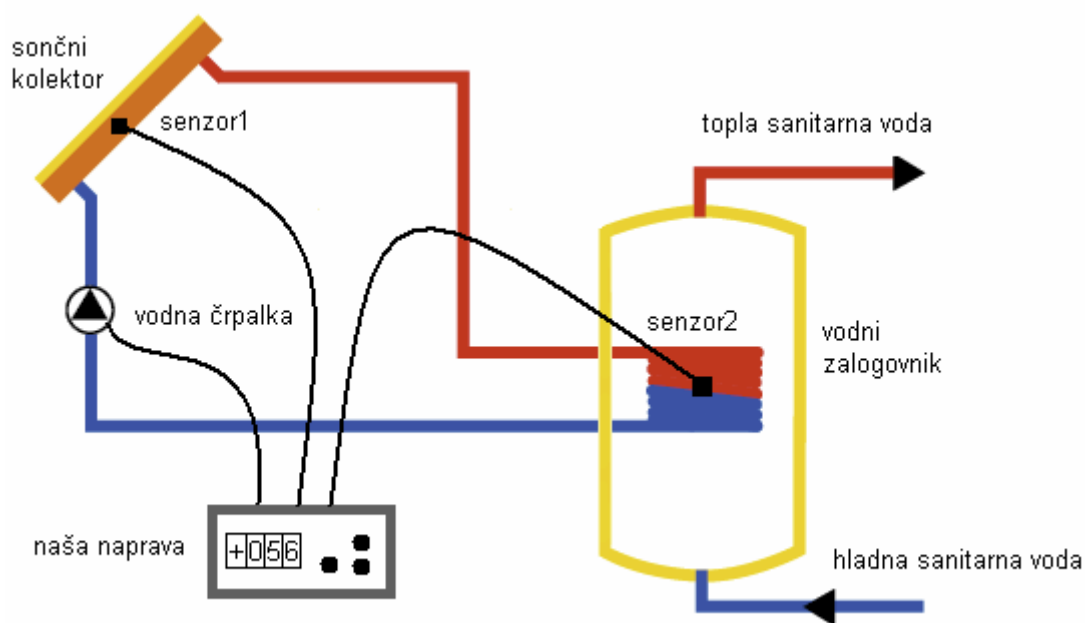
- Uporaba merilnih instrumentov.

1.4 Vsebina

Diplomska naloga ima deset poglavij. Drugo poglavje predstavlja idejno rešitev problema. V tretjem poglavju je predstavljeno prototipno vezje s potrebnimi izračuni in meritvami vmesniškega protokola I²C. V četrtem poglavju je opisan pretvornik A/D. V petem poglavju je opisan krmilnik svetlečih diod LED. V šestem poglavju opisujemo mikrokrmilnik, podrobneje so opisane periferne enote, ki smo jih za izdelek rabili, ostalih enot ne opisujemo. Sedmo poglavje opisuje komunikacijski protokol, ki smo ga uporabili za povezavo z računalnikom. Osmo poglavje vsebuje opis programske opreme ter opis programa s pomočjo odločitvenega diagrama. V devetem poglavju predstavimo merilne rezultate. Deseto poglavje je sklep v katerem ocenjujemo prednosti in slabosti izdelka ter naštejemo nekaj možnih izboljšav.

2. OPIS SISTEMA

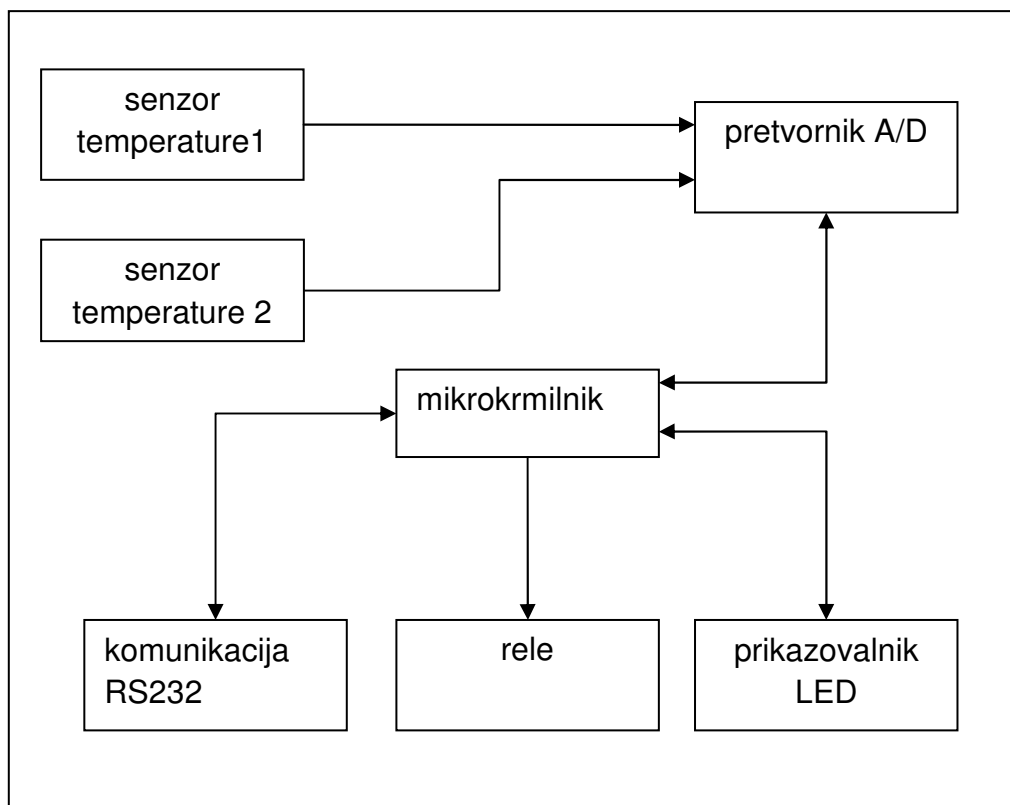
Za sistem na sliki 2.1 bomo izdelali napravo, ki bo vklopljala črpalko v odvisnosti od izmerjene temperature vode v sončnem kolektorju in izmerjene temperature v vodnem zalogovniku ter nastavljene razlike temperatur. Naprava bo imela štiri znakovni prikazovalnik LED, tri tipke za izbiro prikaza temperatur in vnos razlike temperatur. Napravo bomo



Slika 2.1: Slika sistema

lahko povezali na osebni računalnik z vmesnikom RS232. Na sliki 2.2 je blokovna shema naprave. Za vzorčenje temperature bomo uporabili pretvornik A/D. Pretvornik A/D pošilja vzorce po zaporednem vodilu I²C, po istem vodilu prejema tudi ukaze za nastavljanje pretvornika. Ukaze pošiljamo z mikrokrmilnikom pretvorniku A/D in prikazovalniku LED. Mikrokrmilnik prejete podatke obdela in jih pošilja na prikazovalnik ali komunicira preko računalnika ter vključi in izključi rele in s tem vodno črpalko. V diplomski nalogi bomo najprej predstavili potrebne izračune za zajemanje temperature senzorjev in vezji za prikazovalnik LED in vezje za komunikacijo, vzorčenje in vklop releja. Opišemo še kako določimo potrebne vire napetosti v vezju. Sledi opis uporabljenih gradnikov vezja: pretvornik

A/D, krmilnik LED diod, mikrokrmilnik. Opisan je tudi protokol UART, ki ga uporabljamo za komunikacijo z računalnikom. Na koncu podamo še opis programske opreme, ki je predstavljena z odločitvenim diagramom in komentiramo rezultate meritev na vodilu I²C.



Slika 2.2: Blokovna shema naprave

3. VEZJE IN IZRAČUNI

3.1 Prototip vezja s krmilnikom diod

Nastavitev frekvence oscilatorja in segmentnega toka za krmilnik diod

Krmilnik diod nam omogoča enostavno prikazovanje štirih znakov, ki so že zapiani v pomnilniku ROM krmilnika [5,7,9,10]. Vse kar potrebujemo je nastavitev frekvence notranjega oscilatorja in segmentnega toka, ter priključitev diod na izhodne priključke. Notranji RC oscilator uporablja zunanji upor R_{SET} in zunanji kondenzator C_{SET} , s katerima nastavljamo frekvenco oscilatorja F_{OSC} , glej formulo 3.2. Dovoljeno območje F_{OSC} je med 1 MHz do 8 MHz. R_{SET} nam tudi določa največji segmentni tok, formula 3.1. Priporočljivi vrednosti za R_{SET} in C_{SET} nam določita f_{OSC} 4 MHz, kar tudi določi frekvenco utripanja na 0,5 Hz. Priporočena vrednost R_{SET} določi maksimalni segmentni tok na 40 mA in tako lahko določamo segmentni tok v korakih po 2,5 mA od 2,5 do 40 mA.

$$I_{SEG} = K_I / R_{SET} \text{ (mA)} \quad (3.1)$$

$$F_{OSC} = K_F / (R_{SET} \times (C_{SET} + C_{PARAZITNA})) \text{ (MHZ)} \quad (3.2)$$

Pri tem:

K_I = konstanto podaja proizvajalec krmilnika (2144),

K_F = konstanto podaja proizvajalec krmilnika (6003),

R_{SET} = vrednost upora v $k\Omega$,

C_{SET} = vrednost kondenzatorja v pF,

$C_{PARAZITNA}$ = parazitna kapacitivnost iz priključka OSC na priključek GND v pF, tipično 2pF. Priporočena vrednost za R_{SET} je 53,6 $k\Omega$, za C_{SET} pa 26pF. R_{SET} je minimalna dovoljena vrednost, saj ta določi maksimalni segmentni tok krmilnika. Če želimo, lahko izberemo višjo vrednost za R_{SET} , ter s tem določimo nižjo vrednost segmentnega toka konice. Zagotoviti moramo, da ne presežemo maksimalnega toka v konici, ki ga LED diode še lahko prenesejo.

Efektivna kapacitivnost vsebuje kapacitivnost zunanjega kondenzatorja C_{SET} ter tujo kapacitivnost $C_{PARAZITNA}$ iz priključka OSC na priključek GND. Vrednost te kapacitivnosti je odvisna od oblike povezav vezja in je tipično med 1 pF in 5 pF. Izračun segmentnega toka z vrednostmi, ki so uporabljene v vezju:

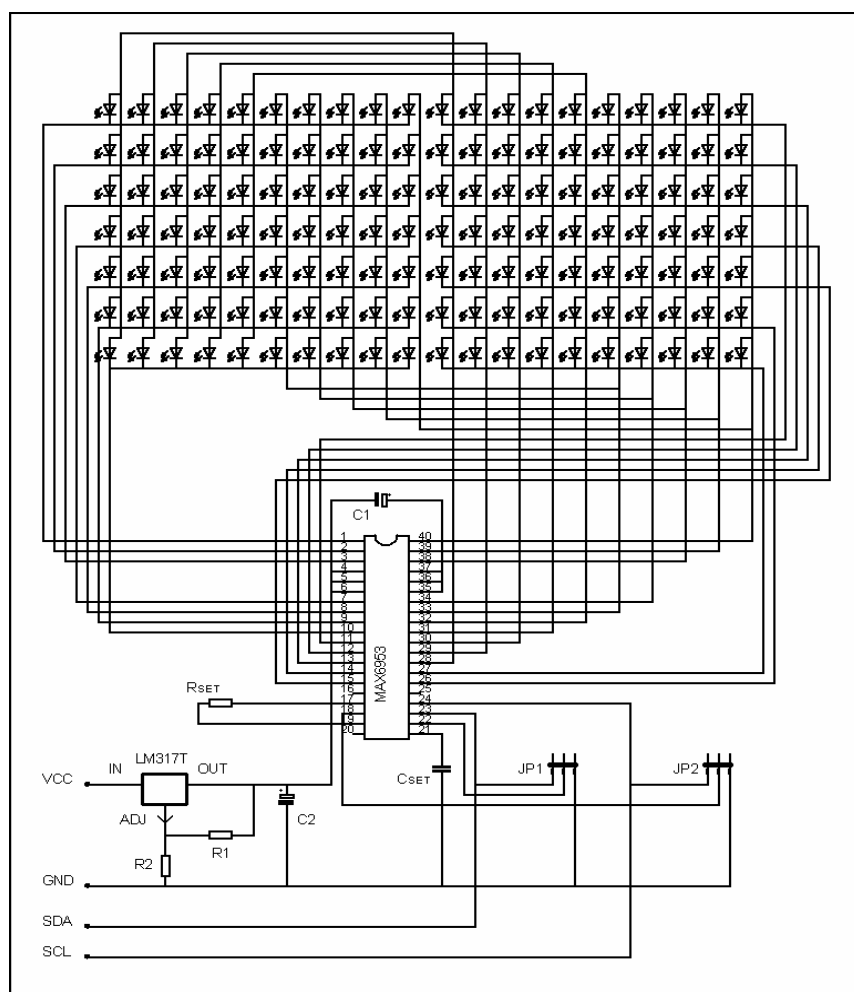
$$I_{SEG} = K_I / R_{SET} \text{ (mA)}$$

$$I_{SEG} = 2144 / 56 = 38,3 \text{ mA}$$

Izračun frekvence notranjega oscilatorja z vrednostmi, ki so uporabljene v vezju:

$$F_{OSC} = K_F / (R_{SET} \times (C_{SET} + C_{PARAZITNA})) \text{ (MHZ)}$$

$$F_{OSC} = 6003 / (56 \times (27 + 2)) = 3,7 \text{ MHz}$$



Slika 3.1 Vezje s krmilnikom diod LED

3.2 Prototip vezja za komunikacijo, vzorčenje in vklop releja.

Izračuni za zajemanje vzorcev temperature.

Temperaturo bomo merili s temperaturnim senzorjem KTY 10-6 v merilnem mostiču. Napetost vzorčimo s pretvornikom A/D, zajete vrednosti pošljemo po vodilu I²C v mikrokrmilnik. Najprej si oglejmo temperaturni senzor KTY 10-6. S tem senzorjem lahko merimo temperaturo zraka, plinov in tekočin v temperaturnem območju med -50 °C in 150 °C. Temperaturo zaznava n-prevodniški element izdelan iz silikonskega kristala v planarni tehnologiji. Rahlo ukrivljena krivulja karakteristike elementa opisujejo sledeče formule, ki veljajo v temperaturnem območju od -30°C do 130 °C.

$$R_T = R_{25} \times (1 + \alpha \times \Delta T_A + \beta \times \Delta T_A^2) = f(T_A) \quad (3.3)$$

kjer je $\alpha = 7.88 \times 10^{-3} K^{-1}$ in $\beta = 1.937 \times 10^{-5} K^{-2}$,

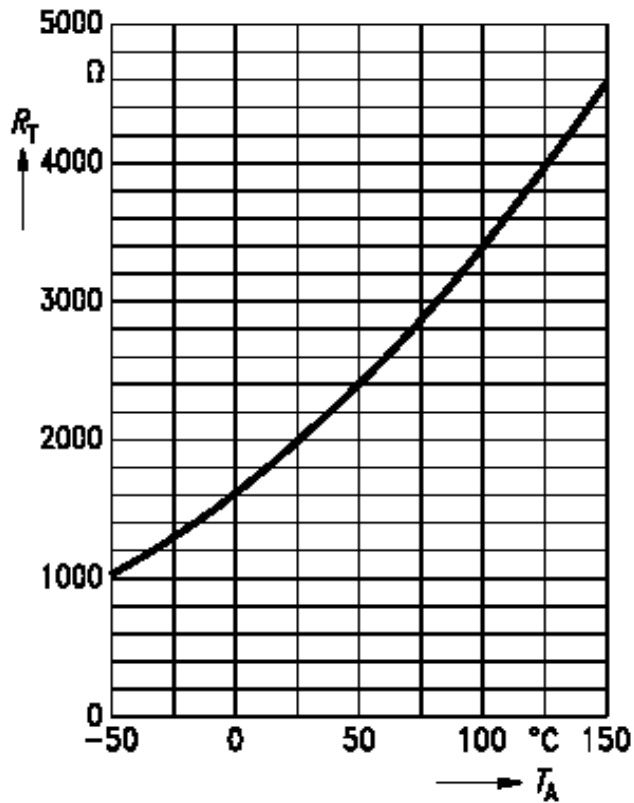
temperaturni faktor k_T izrazimo tako:

$$k_T = \frac{R_T}{R_{25}} = 1 + \alpha \times \Delta T_A + \beta \times \Delta T_A^2 = f(T_A), \quad (3.4)$$

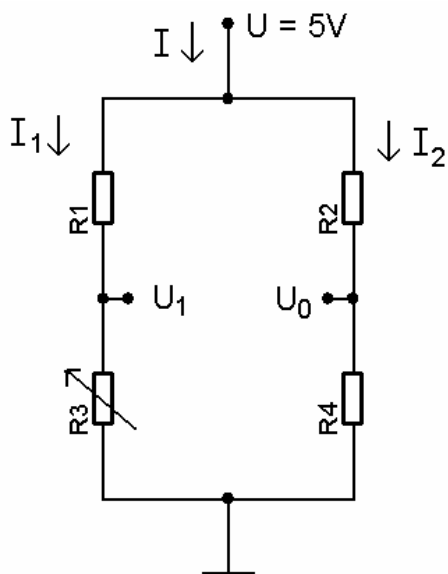
Temperaturo na senzorju izračunamo s spremembo upornosti po naslednji formuli:

$$T = \left(25 + \frac{\sqrt{\alpha^2 - 4 \times \beta + 4 \times \beta \times k_T} - \alpha}{2 \times \beta} \right) ^\circ C \quad (3.5)$$

Iz zgornjih formul dobimo karakteristiko odvisnosti med upornostjo senzorja in temperaturo, ki jo prikazuje slika 3.2. Senzor vstavimo v merilni mostiček, kot prikazuje slika 3.3. Mostiček je sestavljen iz treh uporov in senzorja, upora R_1 in R_2 imata enako vrednost 6,2 kΩ. Ko imata senzor R_3 in upor R_4 enako vrednost, bo napetost med sponkama U_0 in U_1 enaka 0. Rečemo, da je mostiček v ravnovesju. S spreminjanjem temperature na senzorju se spreminja upornost R_3 , posledično se spreminja tudi napetost U_1 . S segrevanjem se večja upornost in



Slika 3.2 Karakteristika odvisnosti med temperaturo in upornostjo senzorja



Slika 3.3 Merilni mostiček.

raste napetost U_1 in obratno. Mostiček smo uravnesili z uporom R_4 vrednosti $2,7\text{k}\Omega$. Izračunajmo koliko stopinj mora biti na senzorju R_3 , da je mostiček v ravnovesju:

$$k_T = \frac{R_T}{R_{25}} = \frac{2,7k\Omega}{2000k\Omega} = 1,35$$

$$\begin{aligned} T &= \left(25 + \frac{\sqrt{\alpha^2 - 4 \times \beta + 4 \times \beta \times k_T - \alpha}}{2 \times \beta}\right) \\ &= \left(25 + \frac{\sqrt{(7,88 \times 10^3 K^{-1})^2 - 4 \times (1,937 \times 10^5 K^{-2}) + 4 \times (1,937 \times 10^5 K^{-2}) \times 1,35 - 7,88 \times 10^3 K^{-1}}}{2 \times (1,937 \times 10^5 K^{-2})}\right) \\ &= \left(25 + \frac{\sqrt{62,09 \times 10^6 K^{-2} - 77,48 \times 10^6 K^{-2} + 1046 \times 10^6 K^{-2} - 7,88 \times 10^3 K^{-1}}}{3874 \times 10^6 K^{-2}}\right) \\ &= \left(25 + \frac{\sqrt{62,09 \times 10^6 K^{-2} - 77,48 \times 10^6 K^{-2} + 1046 \times 10^6 K^{-2} - 7,88 \times 10^3 K^{-1}}}{3874 \times 10^6 K^{-2}}\right) \\ &= 25 + 404 = 654^\circ C \end{aligned}$$

Pretvornik A/D vzorči razliko napetosti med sponkama U_1 in U_0 . Napetost U_0 se ne spreminja in znaša:

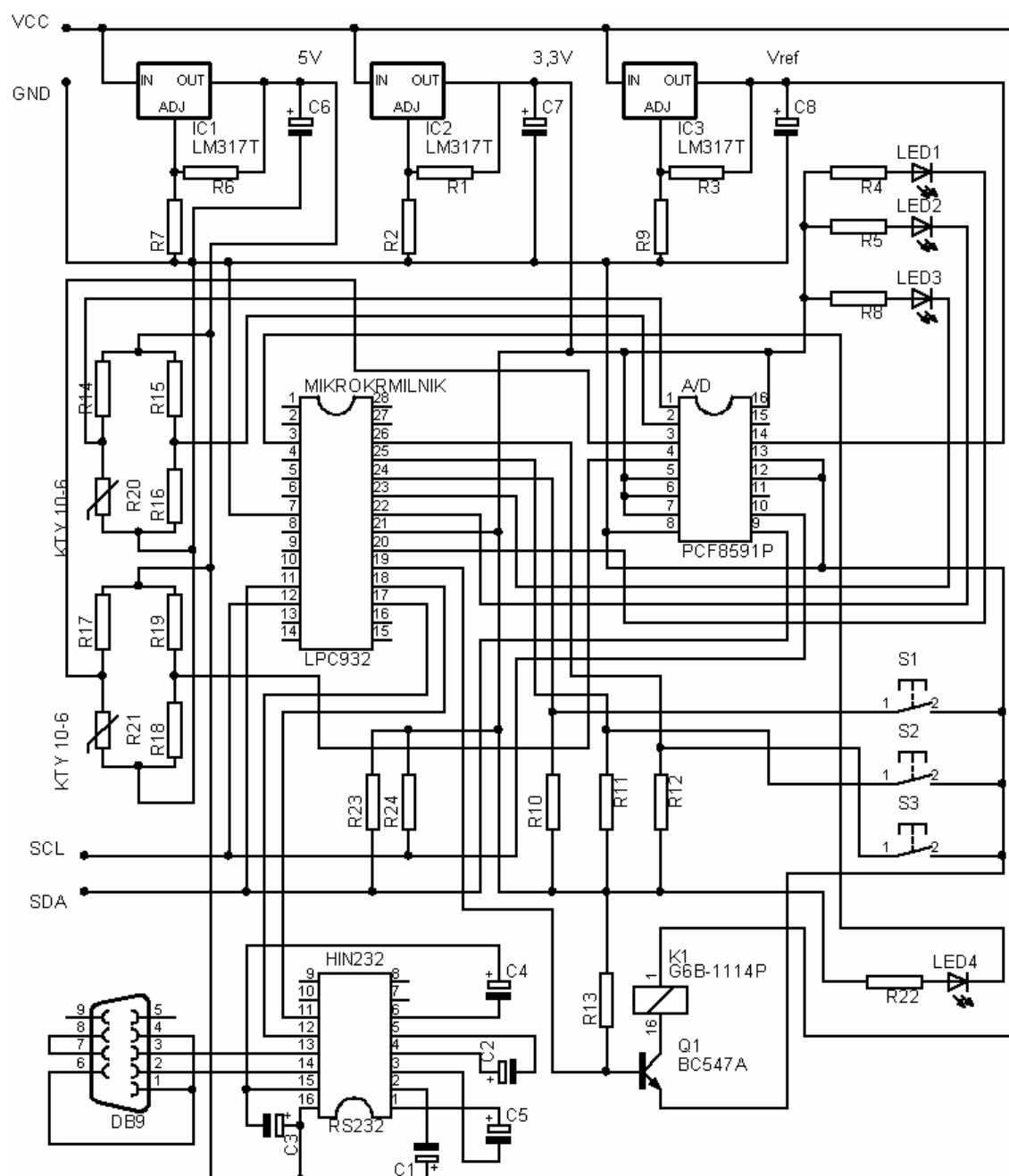
$$U_0 = \frac{U \times R_4}{R_2 + R_4} = \frac{5V \times 2,7k\Omega}{6,2k\Omega + 2,7k\Omega} = 1,52V$$

Napetost U_1 je odvisna od spreminjanja upornosti R_3 senzorja in jo izračunamo:

$$U_1 = \frac{U \times R_3}{R_1 + R_3}$$

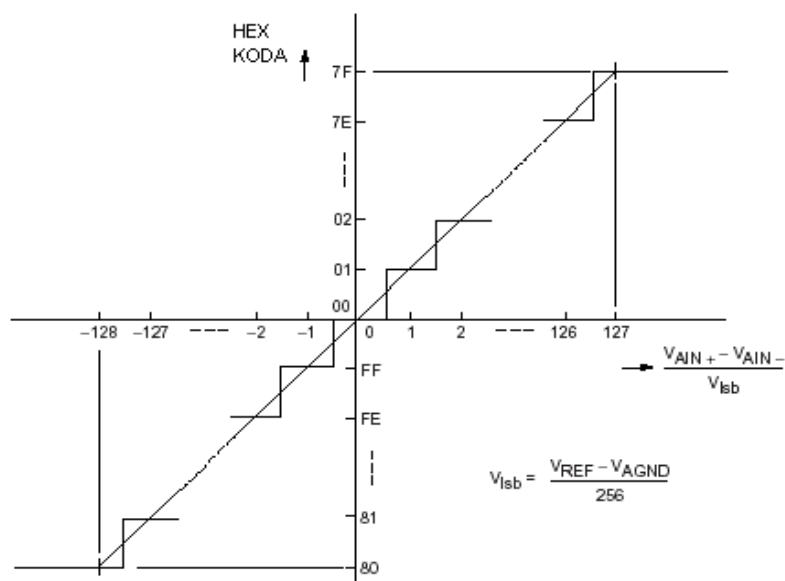
V prilogi je tabela z izračunanimi vrednostmi U_1 pri stopinjah med $-20^\circ C$ in $130^\circ C$. V tabeli je izračunana tudi napetostna razlika, ko se temperatura poveča ali zniža za eno stopinjo. Ta razlika znaša $7,3mV$ kar ustreza višini vzorčne stopnice V_{LSB} . Sedaj lahko izračunamo referenčno napetost za pretvornik A/D, ki znaša:

$$V_{REF} = 256 \times V_{LSB} = 256 \times 0,0073 = 1,87V$$



Slika 3.4 Vezje za komunikacijo, vzorčenje in vklop releja.

Slika 3.5 prikazuje, kako se spremeni binarna koda v odvisnosti od vhodne napetosti. Recimo, da na vhodu ni napetosti, mostič je v ravnovesju, pretvornik A/D vrne vrednost 00 heksadecimalno. Izračunajmo še za primer da je v okolici 25 °C in je upornost senzorja $R_3 = 2k\Omega$:



Slika 3.5 Pretvorba napetosti v binarno kodo.

$$U_1 = \frac{U \times R_3}{R_1 + R_3} = \frac{5V \times 2k\Omega}{6,2k\Omega + 2k\Omega} = 1,22V ,$$

na vhodu pretvornika A/D je napetostna razlika:

$$U_1 - U_0 = 1,22V - 1,52V = -0,3V ,$$

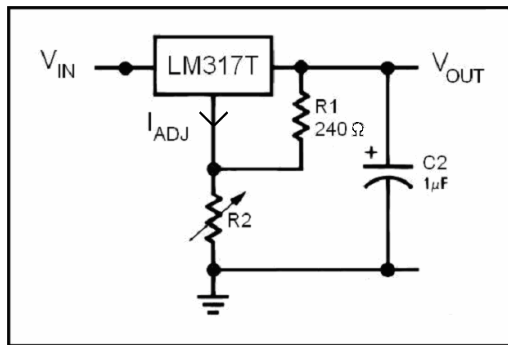
ki jo razdelimo z V_{LSB} :

$$\frac{-0,3V}{0,0073V} \approx -41,1 ,$$

pretvornik A/D vrne binarno vrednost, ki ustreza D6 heksadecimalno. Z drugimi besedami, če imamo uravnotežen mostič pri $65,4^\circ\text{C}$ nam pri temperaturi 25°C in upornosti $R_3 = 2k\Omega$, pretvornik A/D vrača desetiško vrednost $65,4 - 41,1 = 24,3$ kar ustreza 24°C .

3.3 Določitev potrebnih napajalnih napetosti za vezje.

Slika 3.6 [7] prikazuje, kako povežemo priključke vezja LM317T.



Slika3.6 : Vezje z napetostnim regulatorjem LM317T

Naše vezje potrebuje tri različne napajalne napetosti. Pretvornik A/D, mikrokrmilnik in krmilnik LED napajamo z $V_{OUT} = 3,3V$. Merilni mostič in integrirano vezje za RS232 napajamo z $V_{OUT} = 5V$. Za referenčno napetost na pretvorniku A/D potrebujemo $V_{OUT} = 1,87V$. Proizvajalec podaja formulo 3.6 za izračun izhodne napetosti V_{OUT} :

$$V_{OUT} = 1,25 \text{ V} \cdot \left(1 + \frac{R_2}{R_1} \right) + I_{ADJ} \cdot R_2 \quad (3.6)$$

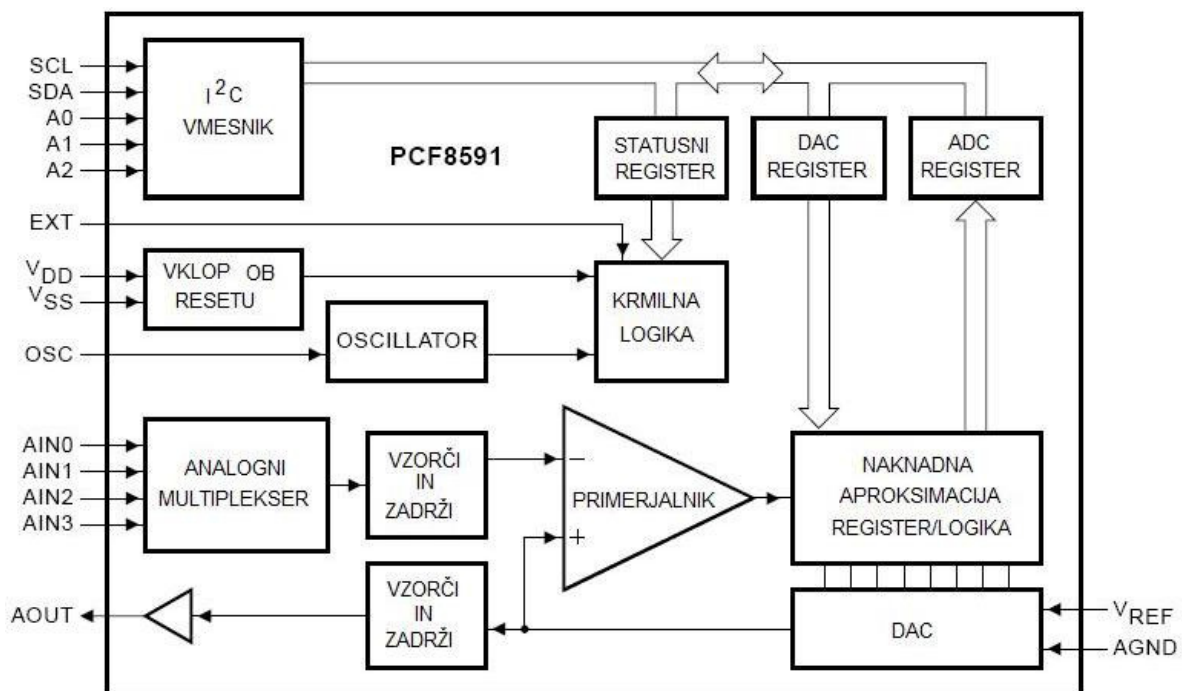
Napajalne napetosti V_{OUT} smo izbrali, medtem ko sta vrednosti za $R_1 = 240\Omega$ in $I_{ADJ} = 50 \mu A$ že določeni. Preostane samo to, da izračunamo vrednost za spremenljivi upor R_2 . Zgornjo enačbo preuredimo in dobimo:

$$R_2 = \frac{V_{OUT} - 1,25 \text{ V}}{\frac{1,25 \text{ V}}{R_1} + I_{ADJ}}$$

4. PRETVORNIK A/D

4.1 Splošni opis

Naprava PCF8591 je izdelana v CMOS tehnologiji, ima štiri analogne vhode za zajemanje podatkov[10]. Za komunikacijo uporablja vmesnik I²C, priključimo lahko največ osem naprav brez dodatne strojne opreme. Funkcije naprave so multipleksiranje analognih vhodov, zadrževalna funkcija, 8-bitna analogno digitalna in 8-bitna digitalno-analoga pretvorba. Blokovna shema je na sliki 4.1.



Slika 4.1: Blokovna shema pretvornika A/D

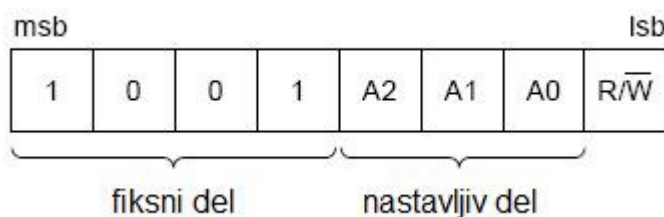
4.2 Pretvorba A/D

Pretvorba A/D se vedno začne po pošiljanju veljavnega bralnega naslova napravi. Pretvorbeni cikel A/D se sproži pri prehodu 1 na 0 potrditvenega urinega impulza in se izvaja med

pošiljanjem vzorca prejšnje pretvorbe. Ko je pretvorbeni cikel sprožen, se zajame vzorec vhodne napetosti na izbranem kanalu. Vzorec se pretvori v ustrezno 8-bitno binarno kodo. Vzorci vzeti z diferenčnih vhodov se pretvorijo v 8-bitno kodo z dvojiškim komplementom. Rezultat pretvorbe se shrani v ADC register in čaka na prenos. Prvi poslan zlog v bralnem ciklu vsebuje rezultat predhodno zajetega vzorca. Prvi brani zlog takoj po vklopu je 80 heksadecimalno. Maksimalna frekvenca vzorčenja je določena s hitrostjo vodila I²C.

4.3 Naslavljanje naprave

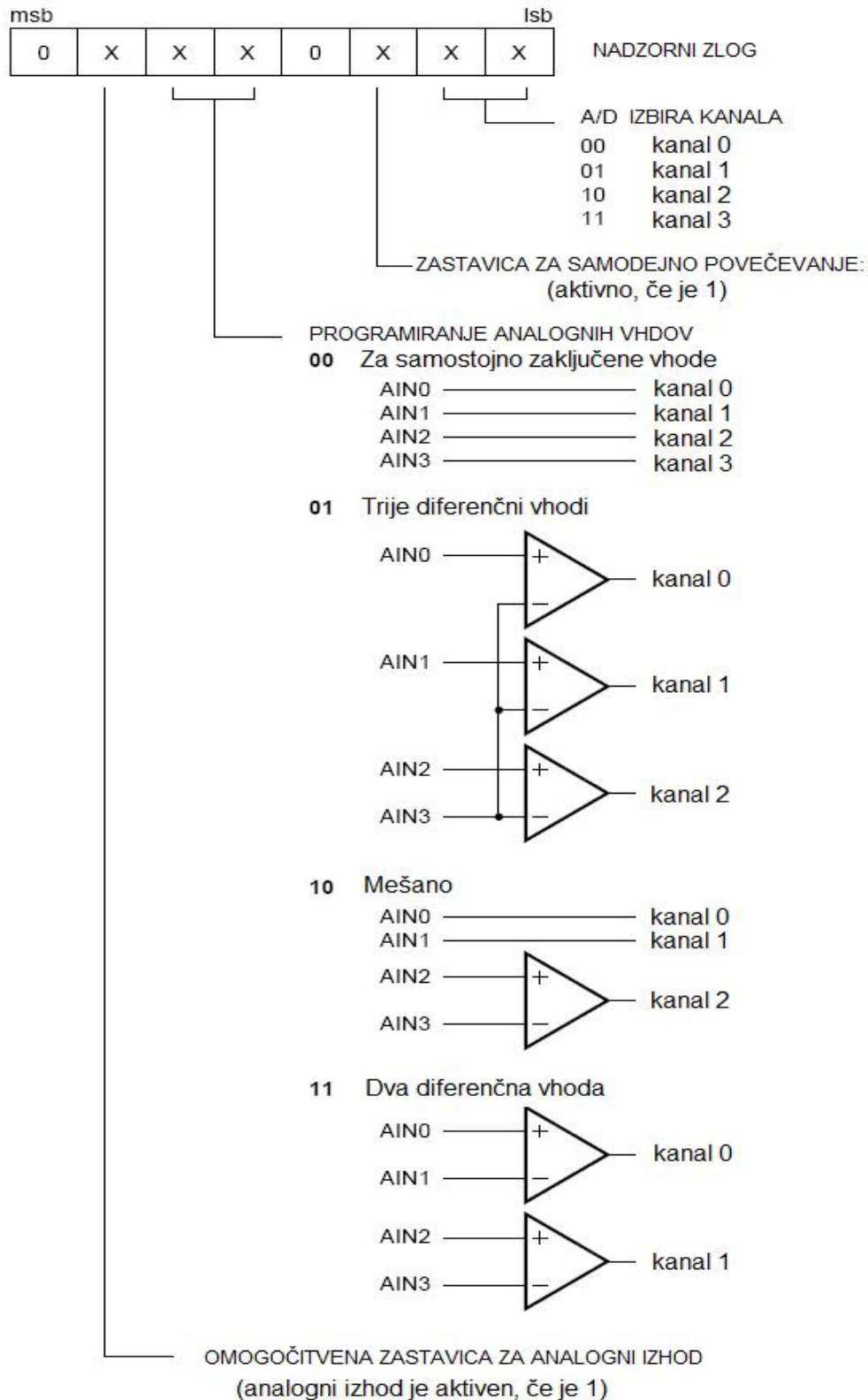
Vsako napravo PCF8591 aktiviramo s pošiljanjem veljavnega naslova na vodilo I²C. Naslov vsebuje fiksni in nastavljeni del. Nastavljeni del nastavimo s priključki A0, A1 in A2. Naslov je vedno prvi poslan zlog, po start pogoju na vodilu I²C. Zadnji bit naslovnega zloga določa smer podatkov, ki jih prenašamo.



Slika 4.2: Naslovni zlog

4.4 Nadzorni zlog naprave

Drugi poslani zlog napravi se shrani nadzorni register in je potreben za nadzor funkcij naprave. Zgornji del nadzornega registra se uporablja za omogočanje analognega izhoda in za nastavljanje analognih vhodov. Analogni vhodi so lahko enostransko zaključeni ali diferenčni. Spodnji del nadzornega registra izbere en analogni vhod, ki je določen z zgornjim delo registra. Če je postavljena zastavica za samodejno povečevanje, se zaporedna številka kanala samodejno povečuje po vsaki A/D pretvorbi. Bit 0 in 7 sta rezervirana za prihodnje funkcije in sta vedno postavljena na logično 0.

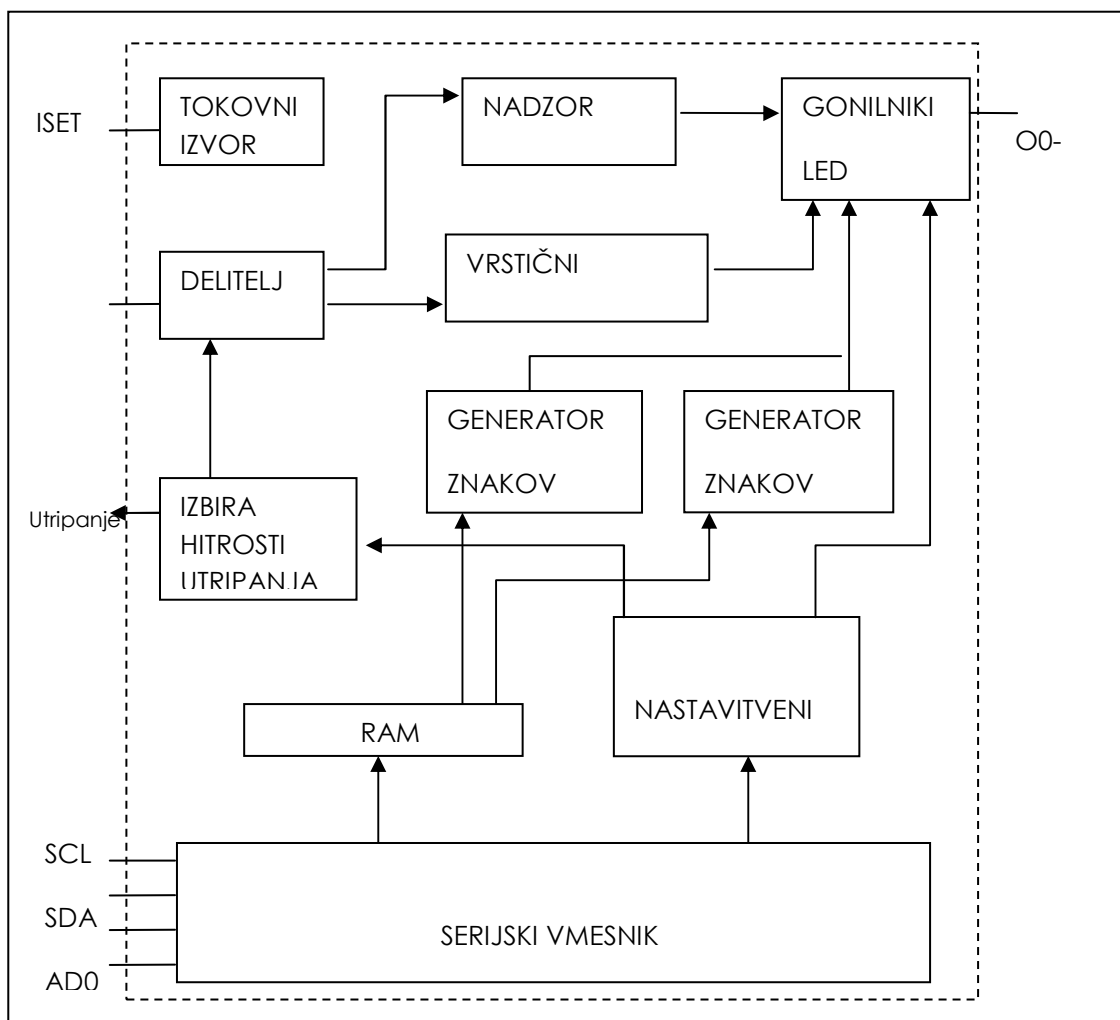


Slika 4.3: Nadzorni zlog

5. KRMILNIK DIOD LED

5.1 Splošen opis krmilnika diod LED

Na kratko bomo opisali blokovno shemo krmilnika svetlečih diod, ki je prikazana na sliki 5.1 [5]. Serijski vmesnik I²C ima naslednje priključke: podatkovna linija SDA, urina linija SCL ter priključka za nastavljanje naslova sužnja ADO in AD1. Preko zaporednega vmesnika uporabnik dostopa do RAM-a in nastavitvenih registrov. Generator znakov RAM iz vsebine RAM-a generira znake in jih posreduje gonilniku svetlečih diod. Gonilnik LED poskrbi za prikaz štirih znakov velikosti 5 x 7 diod. Krmilnik svetlečih diod ima v ROM-u že



Slika 5.1: Blokovna shema krmilnika LED diod

vnešene znake ASCII. Generator znakov ROM iz vsebine ROM-a zelo hitro generira znake in jih posreduje gonilniku LED diod, ker so znaki že v pomnilniku ROM. V nastavitvene registre vnesemo podatke, ki upravljajo s krmilnikom svetlečih diod. Taki podatki nso na primer: jakost svetilnosti znakov, uporabniški znaki, prikaži test itd. Podatki o nastavitvah se uporabljajo pri gonilnikih LED in v primeru uporabe funkcije za utripanje. Na krmilnik lahko priključimo tudi zunanji oscilator ali pa uporabimo kar notranjega, ki ga nastavimo z uporomo in kondenzatorjem. Frekvenco oscilatorja uporabljamo za nadzor svetilnosti in vrstično multipleksiranje.

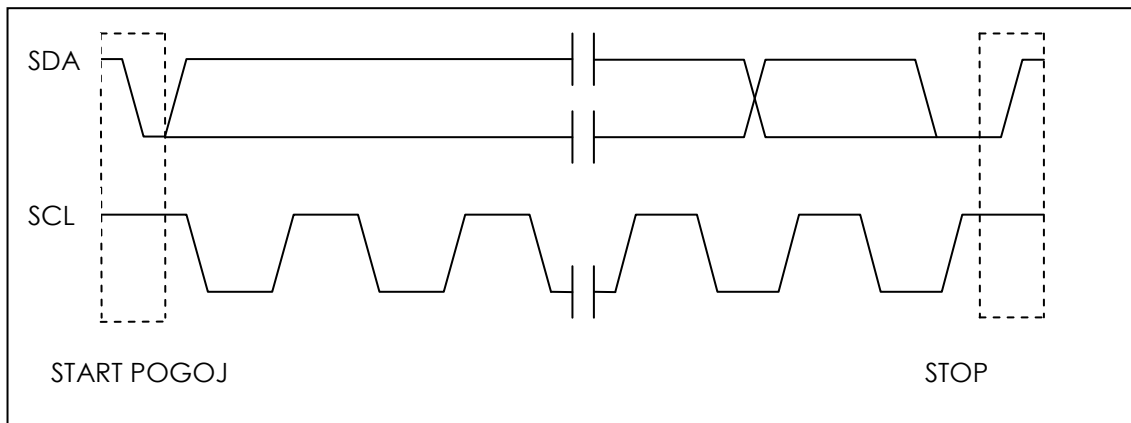
5.2 Komunikacijski vmesnik

Zaporedno naslavljanje

Zaporedni vmesnik MAX6953 deluje kot suženj, ki pošilja in prejema podatke preko vmesnika I²C. Vmesnik uporablja podatkovno linijo SDA in urino linijo SCL za doseg dvostranske komunikacije med gospodarjem in sužnjem. Gospodar (v našem primeru P89LPC932) inicializira vse podatkovne prenose proti in od integriranega vezja MAX6953, prav tako tudi generira uro SCL za sinhronizacijo prenosa. Linija SDA integriranega vezja MAX6953 deluje tako kot vhod kot tudi ponorni izhod. Na liniji SDA potrebujemo upor tipične vrednosti 4,7 k Ω . Linija SCL deluje samo kot vhod. Na njej potrebujemo upor tipične vrednosti 4,7 k Ω samo tedaj ko, uporabljamo več gospodarjev na vmesniku I²C ali če uporabljamo samo enega gospodarja s ponornim izhodom SCL. Vsak prenos, ki ga generira gospodar, sestavlja: pogoj START, 7-bitni suženjski naslov z bitom R/W, naslov registra, en ali več podatkovnih zlogov ter pogoj STOP.

Pogoji start in stop

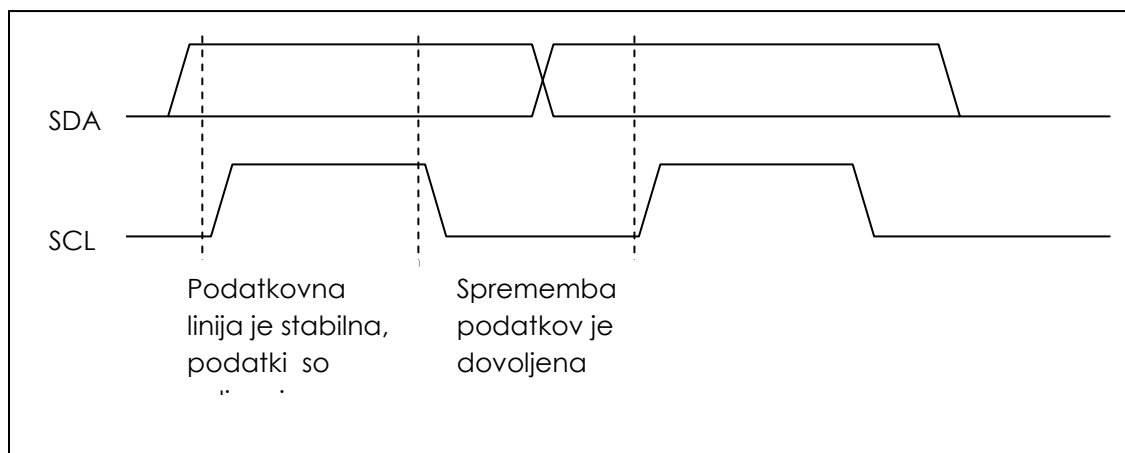
Ko vmesnik ni v uporabi, sta liniji SDA in SCL v visokem stanju. Gospodar začne prenos s pogojem START (S), tako da povleče linijo SDA z visokega na nizek nivo, medtem ko je linija SCL na visokem nivoju. Gospodar konča prenos s pogojem STOP, tako da povleče linijo SDA z nizkega na visok nivo, medtem ko je linija SCL na visokem nivoju. Vodilo je nato prosto in pripravljeno za naslednji prenos. Dogajanje ponazarja slika 5.2.



Slika 5.2: Pogoji start in stop

Prenos bitov

V vsakem urinem ciklu je prenesen en podatkovni bit, kot kaže slika 5.3. Podatki na liniji SDA morajo biti stabilni, medtem ko je linija SCL visoka.

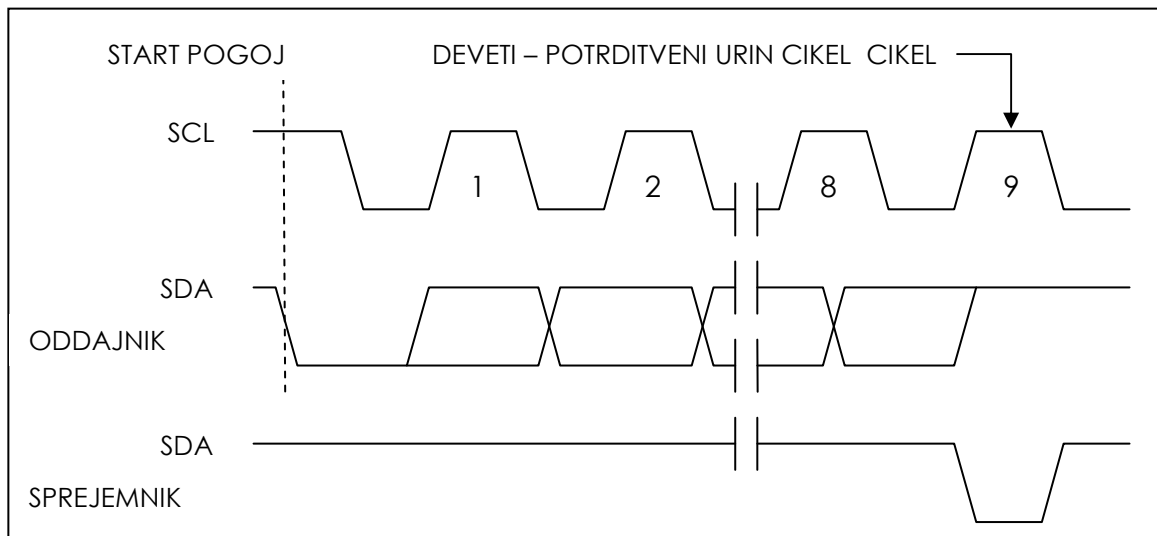


Slika 5.3: Prenos bitov

Potrditev

Potrditveni bit je vsak deveti bit ure, ki ga sprejemnik koristi za potrditev vsakega sprejetega podatkovnega zloga. Tako se za vsak pravilno prenesen zlog porabi devet bitov, prikazanih na sliki 5.4. Gospodar generira deveti urin impulz, sprejemnik pa povleče linijo SDA na nizek nivo med potrditvenim impulzom, tako da je linija SDA stabilna v času, ko je urin impulz v visokem stanju. Ko pošilja podatke gospodar MAX6953-u, MAX6953 generira

potrditveni bit, ker je sprejemnik. Ko pošilja podatke MAX6953 gospodarju, gospodar generira potrditveni bit, ker je sprejemnik.



Slika 5.4: Potrditveni bit

5.3 Prenos podatkov

Naslov sužnja

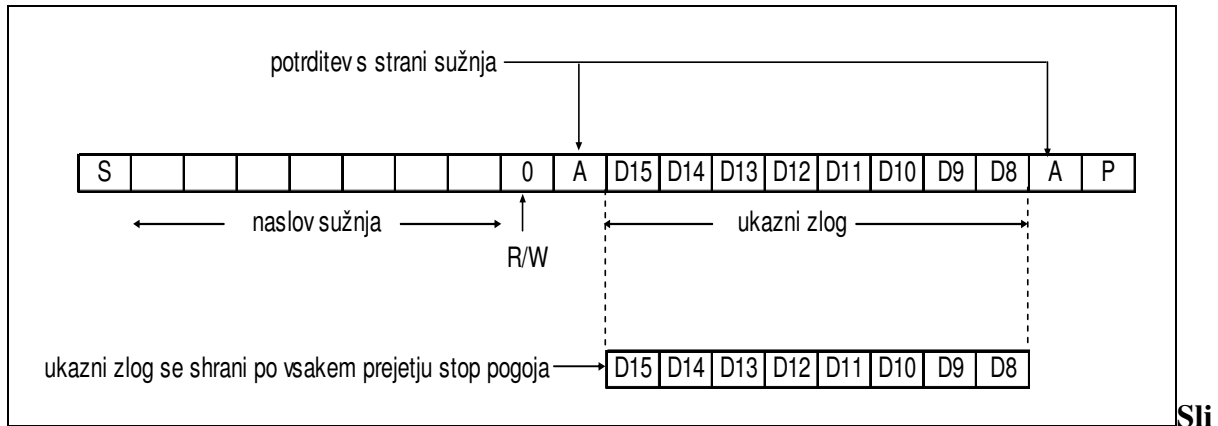
MAX6953 ima 7-bitni suženjski naslov, ki mu sledi osmi bit R/W. Bit R/W je enak 0, če pošiljamo ukaz za pisanje, in enak 1, če pošiljamo ukaz za branje. Prvi trije biti v suženjskem naslovu so vedno 101. Bite A3, A2, A1 in A0 v suženjskem naslovu določimo z vhodoma AD0 in AD1. Ta dva priključka sta lahko priključena na GND, V+, SDA ali SCL. MAX6953 ima lahko največ 16 suženjskih naslovov, kar nam določa maksimalno število možnih vključitev teh naprav na en vmesnik I²C. Tabela 5.1 kaže možne naslove, ki so odvisni od tega, kaj priključimo na priključka AD0 in AD1.

Tabela 5.1: Možni suženjski naslovi krmilnika svetlečih diod

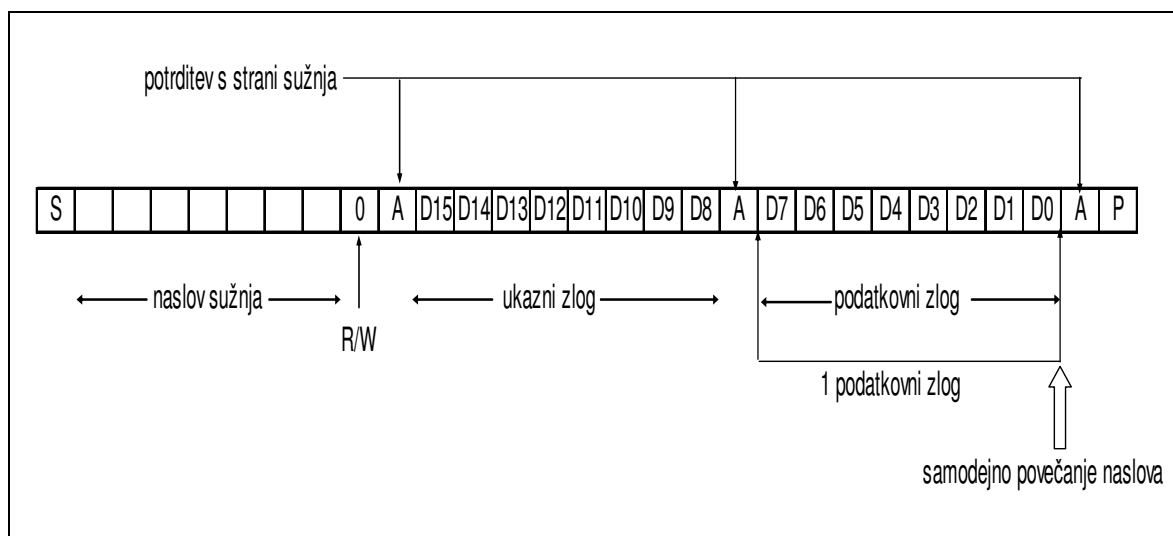
PRIKLJUČEK		NASLOV NAPRAVE						
AD0	AD1	A6	A5	A4	A3	A2	A1	A0
GND	GND	1	0	1	0	0	0	0
GND	V+	1	0	1	0	0	0	1
GND	SDA	1	0	1	0	0	1	0
GND	SCL	1	0	1	0	0	1	1
V+	GND	1	0	1	0	1	0	0
V+	V+	1	0	1	0	1	0	1
V+	SDA	1	0	1	0	1	1	0
V+	SCL	1	0	1	0	1	1	1
SDA	GND	1	0	1	1	0	0	0
SDA	V+	1	0	1	1	0	0	1
SDA	SDA	1	0	1	1	0	1	0
SDA	SCL	1	0	1	1	0	1	1
SCL	GND	1	0	1	1	1	0	0
SCL	V+	1	0	1	1	1	0	1
SCL	SDA	1	0	1	1	1	1	0
SCL	SCL	1	0	1	1	1	1	1

Format sporočila za vpis

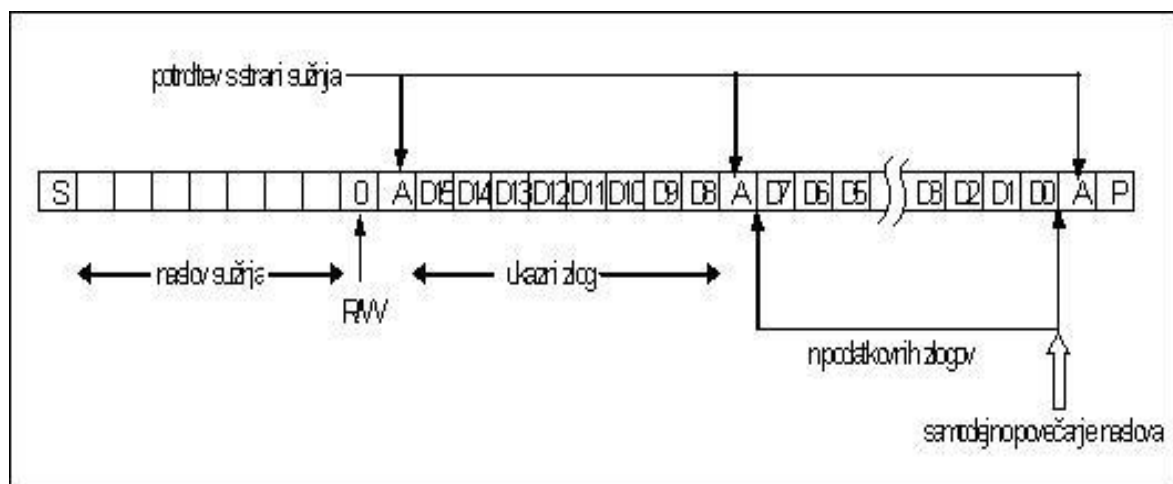
Pisanje v MAX6953 vsebuje prenos suženjskega naslova z bitom R/W, postavljenim na 0, ki mu sledi najmanj en zlog informacije. Prvi podatkovni zlog je ukazni, ki določa, v kateri register se vpisuje naslednji podatkovni zlog. Če je za ukaznim zlogom sprejet pogoj STOP, MAX6953 shrani le ukazni zlog. Prenos ukaznega zloga prikazuje slika 5.5. Vsak zlog, sprejet po ukaznem zlogu, je podatkovni zlog. Prvi podatkovni zlog se shrani v notranjem registru, določenem z ukaznim zlogom. Prenos ukaznega zloga in enega podatkovnega zloga prikazuje slika 5.6. Če pošiljamo več podatkovnih zlogov pred pogojem STOP, se ti običajno shranjujejo v po vrsti si sledeče notranje registre (naslovi za ukazne zloge se običajno samodejno povečajo za 1). Prenos ukaznega zloga in n podatkovnih zlogov prikazuje slika 5.7.



Slika 5.5: Prenos ukaznega zloga



Slika 5.6: Prenos enega podatkovnega zloga



Slika 4.7: Prenos n-podatkovnih zlogov

Format sporočila za branje

Podatke beremo iz MAX6953 z uporabo notranje shranjenega ukaza kot naslovnega kazalca na enak način kot shranjujemo ukazni zlog za vpisovanje. Kazalec se običajno samodejno poveča za 1 vsakokrat, ko preberemo podatkovni zlog na enak način kot pri vpisovanju. Branje začnemo tako, da najprej vpišemo nastavljen ukazni zlog. Gospodar lahko sedaj prebere n zaporednih zlogov, s tem da je prvi zlog prebran iz registra, ki smo ga določili z vpisom na začetku. Ko izvajamo branje po vpisu za preverjanje vpisa, moramo resetirati naslov ukaznega zloga, ker se običajno ta poveča za ena po vsakem vpisu.

Obnašanje ukaznega naslova

Samodejno povečevanje naslova za 1 omogoča nastavljanje MAX6953 z minimalnim številom prenosov ukaznih zlogov. Ukazni naslov za kazalec črk se običajno poveča za 1 z vsakim vpisom ali branjem podatkovnega zloga. Tabela 5.2 nam pove, kako je s samodejnim povečevanjem naslova v odvisnosti od območja ukaznih zlogov.

Tabela 5.2: Obnašanje ukaznega naslova

Območja ukaznih zlogov	Obnašanje samodejnega povečevanja naslova
x0000000 do x0000100	Naslov ukaznega zloga se poveča po vsakem branju ali pisanju.
x0000101	Naslov ukaznega zloga ostane x0000101 po vsakem branju ali pisanju, vendar pa se povečuje naslov znakovnega kazalca.
x0000110	Tovarniško rezerviran, v ta register ne smemo pisati.
x0000111 do x1111110	Naslov ukaznega zloga se poveča po vsakem branju ali pisanju.
x1111111	Naslov ukaznega zloga ostane x1111111 po vsakem branju ali pisanju,

5.4 Registri znakov

MAX6953 uporablja osem registrov za shranjevanje znakov, ki jih želimo prikazati na štirimestnem LED prikazovalniku. En znak predstavlja matriko velikosti 5 x 7 LED diod. Ti registri so izvedeni z dvema nivojema po štiri zloge, ki se imenujeta nivo P0 in nivo P1. Vsak LED znak je predstavljen z dvema zlogoma pomnilnika, enim zlogom nivoja P0 in drugim nivoja P1. Registri znakov so razvrščeni tako, da lahko črkovne podatke osvežimo v nivoju

P0 ali nivoju P1, lahko pa oba nivoja P0 in P1 osvežimo hkrati. Če je funkcija utripanja onemogočena z bitom E v nastavitvenem registru, se podatkovna vsebina v nivoju P0 uporablja za multipleksiranje prikazovalnika. Podatkovna vsebina v nivoju P1 se ne uporablja. Če je funkcija utripanja omogočena, je podatkovna vsebina v obeh nivojih P0 in P1 uporabljena izmenično za multipleksiranje prikazovalnika. Utripanje dosežemo z multipleksiranjem LED prikazovalnika in uporabo obeh nivojev v nasprotnih fazah ure za utripanje. Podatki v registrih znakov ne nadzirajo segmentov znakov neposredno. Podatki v registru se uporabljajo za naslavljanje črkovnega generatorja, ki prikaže podatke iz tabele 128 znakov. Spodnjih 7 bitov znaka (D6 do D0) izbere črko iz tabele 3.8. MSB (D7=0) določi, da so podatki prebrani neposredno ali pa so biti invertirani (D7=1). To nam omogoča, da lahko uporabimo učinek, kjer je ozadje osvetljeno in so črke temne ali obratno. V tabeli 5.3 [5] so podani vsi možni ukazni naslovi, s katerimi dostopamo do registrov znakov. S podatki, ki jih vpisujemo v registre, upravljamo prikaz znakov na prikazovalniku svetlečih diod.

Tabela 5.3: Ukazni naslovi registrov

REGISTER	UKAZNI NASLOVI								HEX KODA
	D15	D14	D13	D12	D11	D10	D9	D8	
brez operacije	X	0	0	0	0	0	0	0	0x00
jakost10	X	0	0	0	0	0	0	1	0x01
jakost32	X	0	0	0	0	0	1	0	0x02
preletna meja	X	0	0	0	0	0	1	1	0x03
nastavitev	X	0	0	0	0	1	0	0	0x04
uporabniški znaki	X	0	0	0	0	1	0	1	0x05
tovarniško rezerviran, prepovedano vpisovanje.	X	0	0	0	0	1	1	0	0x06
prikaži test	X	0	0	0	0	1	1	1	0x07
znak0 nivo P0	X	0	1	0	0	0	0	0	0x20
znak1 nivo P0	X	0	1	0	0	0	0	1	0x21
znak2 nivo P0	X	0	1	0	0	0	1	0	0x22
znak3 nivo P0	X	0	1	0	0	0	1	1	0x23
znak0 nivo P1	X	1	0	0	0	0	0	0	0x40
znak1 nivo P1	X	1	0	0	0	0	0	1	0x41
znak2 nivo P1	X	1	0	0	0	0	1	0	0x42
znak3 nivo P1	X	1	0	0	0	0	1	1	0x43
v nivoja P0 in P1 znaka0 zapiše enake podatke	X	1	1	0	0	0	0	0	0x60
v nivoja P0 in P1 znaka1 zapiše enake podatke	X	1	1	0	0	0	0	1	0x61
v nivoja P0 in P1 znaka2 zapiše enake podatke	X	1	1	0	0	0	1	0	0x62
v nivoja P0 in P1 znaka3 zapiše enake podatke	X	1	1	0	0	0	1	1	0x63

5.5 Nastavitveni register

Nastavitveni register se uporablja za vstop in izstop iz mirovanja, izbiro hitrosti utripanja, omogočitev in onemogočitev funkcije utripanja, za počiščenje vsebine v nivojih P0 in P1 znaka in resetiranje tempiranja utripanja. Biti D6 in D1 sta neuporabljena, njuna vsebina je lahko logična 0 ali 1. Tabela 5.4 prikazuje razvrstitev bitov v nastavitvenem registru, kjer vsaka črka pomeni neko nastavitvev za krmilnik svetlečih diod.

Tabela 5.4: Vsebina nastavitvenega registra

	VSEBINA REGISTRA							
	D7	D6	D5	D4	D3	D2	D1	D0
nastavitveni register	P	X	R	T	E	B	X	S

Bit S v nastavitvenem registru izbere med zaustavitvenim načinom ali normalnim delovanjem.

Bit B v nastavitvenem registru izbere hitrost utripanja. To je hitrost, s katero se segment izmenjuje med nivojema P0 in P1. Hitrost utripanja je določena s frekvenco multipleksne ure OSC, bit B omogoča še izbiro med hitrim in počasnim utripanjem. Pri počasnem utripanju se segmenti osvežujejo z uporabo nivojev P0 in P1 na 1 sekundo, pri hitrem utripanju se segmenti osvežujejo na 0.5 sekunde. Obe hitrosti veljata pri nastavljeni frekvenci OSC=4 kHz.

Bit E v nastavitvenem registru omogoči ali onemogoči utripanje. Če je utripanje omogočeno, potem so uporabljeni podatki v obeh nivojih P0 in P1 za prikaz na prikazovalniku. Če je utripanje onemogočeno, se uporabijo samo podatki v nivoju P0, medtem ko so podatki v nivoju P1 ignorirani.

Bit T omogoča sinhronizacijo učasovanja utripanja za več naprav MAX6953. Pri izbiri logične 1 omogočimo sinhronizacijo, števniki za učasovanje utripanja se med potrditvijo I²C ponastavijo. Pri izbiri logične 0 onemogočimo sinhronizacijo, števniki za učasovanje utripanja se ne ponastavijo.

Bit R zbríše vsebino digita v obeh nivojih P0 in P1 za čas potrditve I²C. Logična 0 ne vpliva na vsebino obeh nivojev P0 in P1. Logična 1 zbríše vsebino znaka v obeh nivojih med potrditvijo I²C.

Bit P odseva stanje izhoda za utripanje v trenutku, ko beremo nastavitveni register. Če smo v fazi utripanja nivoja P1, je izhod logična 1, v fazi nivoja P0 je izhod logična 0.

5.6 Tabela ASCII znakov

Tabela 5.6 prikazuje naslove ASCII znakov v ROM pomnilniku in prazne RAM lokacije.

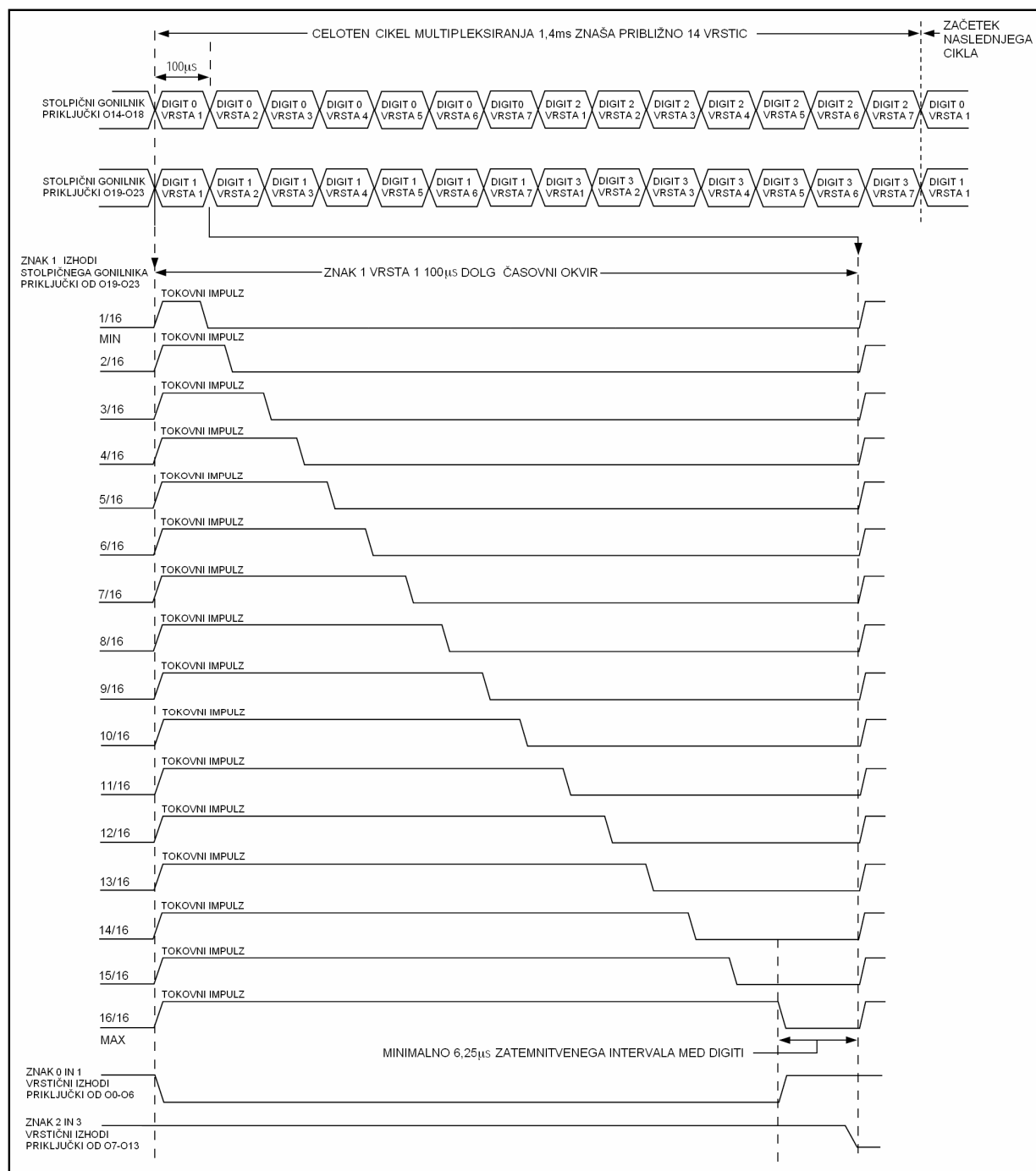
Tabela 5.6: Naslovi ASCII znakov v ROM spominu in prazne RAM lokacije

MSB LSB	X000	X001	X010	X011	X100	X101	X110	X111
0000	RAM00	RAM16		0	@	P	`	p
0001	RAM01	RAM17	!	1	A	Q	a	q
0010	RAM02	RAM18	"	2	B	R	b	r
0011	RAM03	RAM19	#	3	C	S	c	s
0100	RAM04	RAM20	\$	4	D	T	d	t
0101	RAM05	RAM21	%	5	E	U	e	u
0110	RAM06	RAM22	&	6	F	V	f	v
0111	RAM07	RAM23	'	7	G	W	g	w
1000	RAM08	£	(8	H	X	h	x
1001	RAM09	¥)	9	I	Y	i	y
1010	RAM10	€	*	:	J	Z	j	z
1011	RAM11	°	+	;	K	[k	{
1100	RAM12	μ	,	<	L	\	l	
1101	RAM13	±	-	=	M]	m	}
1110	RAM14	↑	.	>	N	^	n	~
1111	RAM15	↓	/	?	O	_	o	□

5.7 Register svetilnosti

Svetilnost prikazovalnika je regulirana s štirimi impulznimi modulatorji, za vsak znak po enim. Vsak znak je krmiljen z delom enega izmed dveh jakostnih registrov, "intensity10" in "intensity32" [5]. Modulator stopnjuje povprečni segmentni tok v korakih od maksimalno

16/16 do najmanj 1/16 maksimalnega toka. Slika 5.8 prikazuje multipleksni časovni okvir in širino impulza pri različnih nastavitvah jakostnega registra.

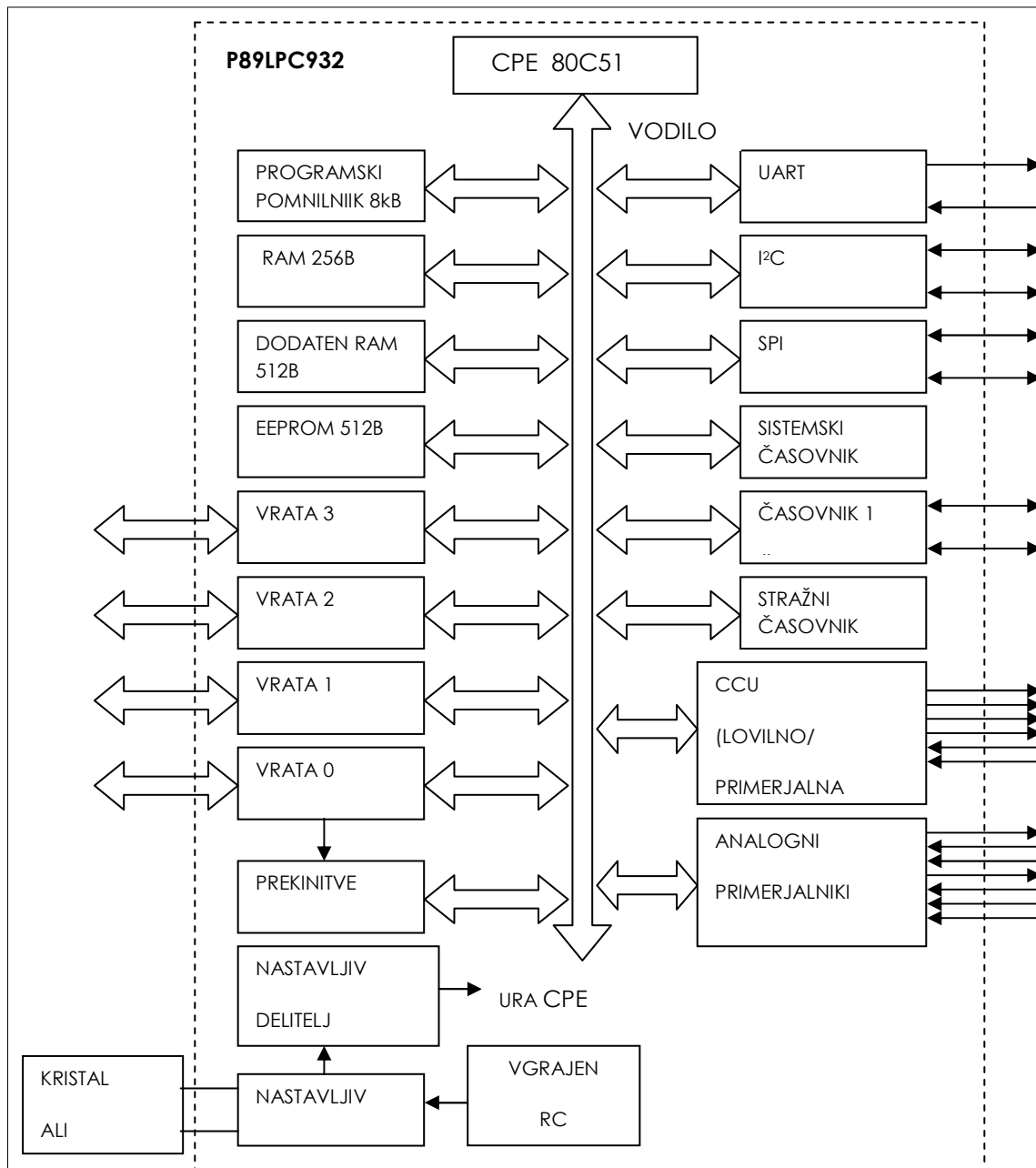


Slika 5.8: Multipleksni časovni okvir.

6. MIKROKRMILNIK

6.1 Splošni opis mikrokrmilnika

Za razlago mikrokrmilnika uporabimo blokovno shemo, ki je na sliki 6.1[3,4,8].



Slika 6.1: Blokovna shema mikrokrmilnika P89LPC932

Začnimo v zgornjem levem kotu in se pomikajmo navzdol. V programski pomnilnik shranimo program, ki ga bo mikrokrmilnik izvajal. Naslednji enoti predstavljata dve vrsti pomnilnika RAM. Razlikujeta se po načinu dostopa, to je z različnim naslavljanjem. Naslednji okvir predstavlja notranji EEPROM. To je trajni pomnilnik, vanj spravljamo podatke, ki se ne spreminjajo, na primer nastavitvene parametre nekega stroja. Naslednji štirje okvirji predstavljajo nastavljljiva vhodno/izhodna vrata. Vrata 0 do 2 imajo osem priključkov, medtem ko imajo vrata 3 le dva priključka, ki sta namenjena priključitvi zunanjega oscilatorja. Na vrata 0 lahko neposredno priključimo tipkovnico. Z debelejšo puščico prikazujemo dvosmerno komunikacijo med mikrokrmilnikom in zunanjim svetom. S tanjšo puščico smo prikazali komunikacijo notranjih enot mikrokrmilnika, z njimi komuniciramo samo preko vrat 0 do 2. Te enote so CCU, analogni primerjalniki, dva časovnika in več protokolov za serijsko komunikacijo. V spodnjem levem kotu so okvirji, ki ponazarjajo uporabo notranjega ali zunanjega oscilatorja ter nastavljanje ure za CPE. Na levi strani imamo še stražni časovnik in sistemski časovnik. Stražni časovnik ščiti sistem pred nepravilnim izvajanjem programske kode. Sistemski časovnik je 23-bitni števec, ki ga uporabljamo kot izvor ure, medtem ko je delovanje mikrokrmilnika v zaustavljenem načinu. Je tudi vir prekinitve ali izvor ponovnega zagona mikrokrmilnika.

6.2 Naslavljanje

Načini naslavljanja povedo, kako naslavljammo dano pomnilniško lokacijo. Za CPE tipa 80C51 se uporabljajo naslednji načini naslavljanja:

Takojšnje naslavljanje se imenuje tako zato, ker se operand že nahaja v sami inštrukciji. Inštrukcija sama pove, katero vrednost bomo shranili v pomnilnik. Primer:

```
MOV A,#20h
```

Ta inštrukcija shrani heksadecimalno vrednost 20 v akumulator. Takojšnje naslavljanje je zelo hitro, ker pa imamo fiksne vrednosti v programu, je hkrati zelo togo.

Neposredno naslavljanje se tako imenuje zato, ker dobimo vrednost, ki jo želimo shraniti, neposredno iz neke druge lokacije v pomnilniku. Na primer:

MOV A,30h

Ta inštrukcija bere podatek iz notranjega IRAM-a na lokaciji 30h in ga shrani v akumulator. Ponavadi je takšno naslavljanje hitro, čeprav se shranjevana vrednost ne nahaja v inštrukciji sami, vendar pa se nahaja v notranjem pomnilniku IRAM-u. Takšno naslavljanje je precej bolj prilagodljivo kot takojšnje naslavljanje, ker je vrednost, ki jo želimo naložiti, lahko tudi spremenljivka.

Pomembno je še povedati, da se z uporabo neposrednega naslavljanja vsaka inštrukcija nanaša na naslove notranjega RAM-a med 00h in 7Fh (IRAM). Neposredno naslavljanje lokacij na naslovih med 80h in FFh se nanaša na nadzorne registre SFR, ki nadzirajo delovanje periferije mikrokrmilnika. Mikrokrmilnik ima v notranjem RAM-u 256 zlogov pomnilnika. Če želimo dostopati do 128 zgornjih lokacij notranjega RAM-a (IRAM), moramo uporabiti posredno naslavljanje.

Posredno naslavljanje je izredno učinkovito in večinoma zagotavlja veliko prožnost. Je tudi edini način za dostopanje do zgornjih 128 zlogov notranjega RAM-a (IRAM). Posredno naslavljanje izgleda takole:

MOV A,@R0

Ta inštrukcija povzroči to, da se najprej preveri vrednost v registru R0. CPE nato naloži vrednost, ki se nahaja v notranjem RAM-u na lokaciji, kamor kaže R0, v akumulator. Primer: R0 vsebuje vrednost 40h in na naslovu 40h notranjega RAM-a se nahaja vrednost 67h. Ko se zgornja inštrukcija izvede, CPE preveri vrednost R0. Ker R0 vsebuje vrednost 40h, CPE pobere vrednost iz notranjega RAM-a na naslovu 40h in jo shrani v akumulator. Kot rezultat te inštrukcije imamo sedaj v akumulatorju vrednost 67h. Posredno naslavljanje se vedno nanaša na notranji RAM, nikoli pa na registre SFR.

Zunanje neposredno naslavljanje omogoča dostop do zunanjega pomnilnika z dvema ukazoma:

MOVX A@DPTR

MOVX @DPTR,A

Kot lahko vidimo, oba ukaza uporabljata DPTR. V teh inštrukcijah je najprej potrebno naložiti DPTR z želenim naslovom zunanjega pomnilnika, ki ga želimo brati ali pisati. Ko

DPTR vsebuje pravilen naslov zunanjega pomnilnika, bo prvi ukaz kopiral vsebino na naslovu v akumulator. Drugi ukaz pa naredi ravno nasprotno, vrednost iz akumulatorja kopira na naslov v zunanjem pomnilniku.

Zunanje posredno naslavljanje se lahko uporablja za naslavljanje zunanjega pomnilnika. Podajmo primer takega naslavljanja:

```
MOVX @R0,A
```

Vrednost registra R0 se najprej prebere in potem se vrednost v akumulatorju kopira v zunanji pomnilnik na naslov, ki ga vsebuje R0.

6.3 Pomnilniki mikrokrmilnika

Pomnilniki so organizirani v več skupin: podatkovni pomnilnik, notranji podatkovni pomnilnik, SFR, zunanji podatkovni pomnilnik, programski pomnilnik in EEPROM.

Podatkovni pomnilnik vsebuje 128 zlogov podatkovnega prostora (00h do 7Fh), dosegljivega z neposrednim ali posrednim naslavljanjem ter uporabo inštrukcij, ki so različne od MOVX in MOVC. Cel sklad ali del sklada je lahko v tem pomnilniškem prostoru.

Notranji podatkovni pomnilnik ima 256 zlogov notranjega podatkovnega prostora (00h do FFh), dosegljivega z neposrednim ali posrednim naslavljanjem ter uporabo inštrukcij, ki so različne od MOVX in MOVC. Cel sklad ali del sklada je lahko v tem pomnilniškem prostoru. Ta prostor vključuje podatkovni prostor podatkovnega pomnilnika in prostor 128 zlogov, ki je takoj nad njim.

SFR so lahko izbrani registri CPE, registri za nadzor periferije ter statusni registri, dosegljivi samo z neposrednim naslavljanjem.

Zunanji podatkovni pomnilnik ali zunanji RAM podvoji 64 kB pomnilniški prostor klasičnega 80C51. Naslavljam ga z inštrukcijo MOVX in uporabo SPTR, R0 ali R1. Ves ali del tega pomnilniškega prostora je lahko v čipu. P89LPC932 ima 512 zlogov vgrajenega pomnilniškega prostora tega tipa.

Programski pomnilnik zavzema 64 kB programskega pomnilniškega prostora, dosegljivega kot del izvajalnega programa s pomočjo inštrukcije MOVC. P89LPC932 ima vgrajenih 8 kB kodnega spomina.

EEPROM je dodatno vgrajen v mikrokrmilnik P89LPC932 in obsega 512 zlogov pomnilniškega prostora, dosegljivega preko registrov SFR.

6.4 Vrata

P89LPC932 ima štiri I/O vrata [3]: vrata 0, vrata 1, vrata 2, vrata 3. Vrata 0, 1 in 2 so 8-bitna vrata, medtem ko so vrata 3 2-bitna. Točno število razpoložljivih vrat je odvisno od urinih in izbranih reset opcij. Vsa vrata, z izjemo treh priključkov, lahko nastavimo programsko na enega izmed štirih možnih načinov bit po bitu. Možne nastavitve so: navidez dvosmerna vrata, izhodna vrata z dvižnim uporom, vrata z odprtim ponorom in vhodna vrata. Trije priključki vrat, ki jih ni možno nastaviti na vse načine, so:

- P1.5 (negiran RST) je lahko le vhodni in ga ne moremo nastavljati.
- P1.2 (SCL/T0) in P1.3 (SDA/negiranINT0) lahko nastavimo samo kot vhodna vrata ali kot izhodna ponorna vrata.

Navidez dvosmerna vrata lahko uporabljamo kot vhod ali izhod, kjer ni potrebe po dodatnem ponastavljanju vrat. To je mogoče zaradi tega, ker je z izhodnim visokim signalom izhodna logika vrat šibko krmiljena. To dovoljuje zunanji napravi, da izhodni priključek povleče na nizek nivo. Ko postavimo na priključek nizek nivo, ga krmilimo močneje in tako je sposoben prevajati zelo velike tokove. Način je podoben kot pri ponornem izhodu, s to razliko, da so pri navidez dvosmernem izhodnem načinu uporabljeni trije tranzistorji, ki služijo drugim potrebam. P89LPC932 deluje pri napajalni napetosti 3 V, vendar so vsi priključki tolerantni tudi na 5 V. Če priključimo 5 V na priključek z izbranim navidez dvosmernim načinom, se bo pojavil tok v smeri priključka proti V_{DD} , kar povzroči dodatno trošenje moči. Zaradi tega v tem načinu ni priporočljivo priključevati napetosti 5 V na priključke. Navidez dvosmerna vrata vsebujejo še Schmittovo prožilno vezje z vezjem za dušenje kratkotrajnih izgub signala.

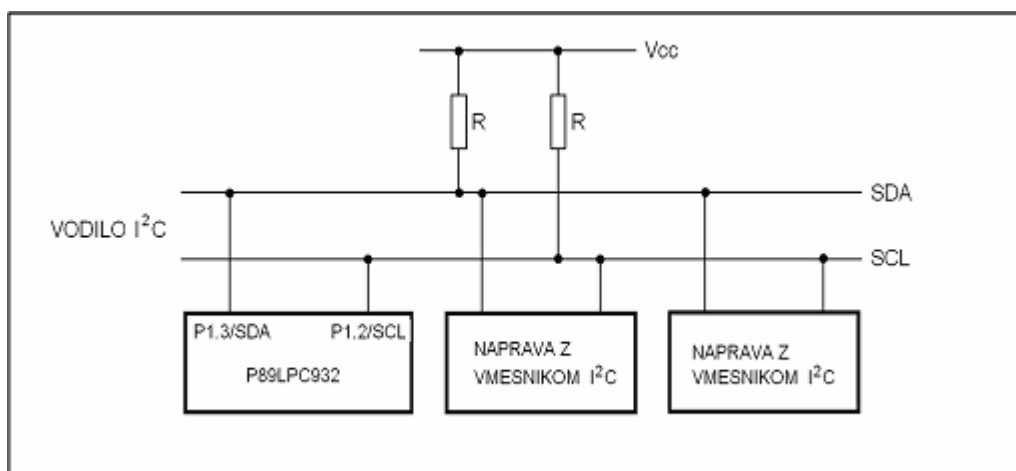
Izhodna ponorna vrata ne vsebujejo notranjih dvižnih uporov in krmilijo samo tranzistor vratnega krmilnika, ko vratni pomnilnik vsebuje logično 0. Za nastavitve teh vrat v vlogo logičnega izhoda potrebujemo zunanji dvižni upor, priključen na V_{DD} . Izhodna ponorna vrata vsebujejo še Schmittovo prožilno vezje z vezjem za dušenje kratkotrajnih izgub signala.

Vhodna vrata nimajo izhodnega krmilnika, vsebujejo pa Schmittov prožen vhod z vezjem za dušenje kratkotrajnih izgub signala.

Izhodna vrata z dvižnim uporom imajo enako izhodno strukturo kot izhodna ponorna vrata in kot navidez dvosmerna vrata, vendar poleg tega zagotavljajo še močen dvižni upor, ko vratni pomnilnik vsebuje logično 1. Ta način lahko uporabljamo, ko želimo več izvirnega toka iz izhoda. Izhodna vrata z dvižnim uporom vsebujejo še Schmittovo prožilno vezje z vezjem za dušenje kratkotrajnih izgub signala.

6.5 Komunikacijski vmesnik I²C

Serijsko vodilo I²C so razvili v podjetju Philips semiconductors že leta 1980. Uporablja se za povezovanje integriranih vezij s hitrostjo 100 kbitov/s v standardnem načinu S do 400 kbitov/s v hitrem načinu F in v novejšem zelo hitrem načinu HS s hitrostmi prenosa do 3,4 Mbitov/s. Napajalne napetosti za način F in S so 5 V, za HS pa 3 V. Vezja, ki delujejo v načinu HS, lahko delujejo tudi v načinu S in F. Aktivni dvosmerni liniji sta podatkovna linija SDA in urina linija SCL. Liniji moramo povezati na pozitivno napajalno napetost preko dvižnih uporov. Vsaka elektronska komponenta, kot so na primer mikrokrmilnik, LCD prikazovalnik, gonilnik, pomnilnik in drugi, ki so povezani na vodilo, imajo svoj naslov. Nadzor nad vodilom ima gospodar vodila, to je ponavadi mikrokrmilnik. Na vodilu se dopušča tudi več gospodarjev. Slika 6.2 prikazuje priključevanje naprav na vodilo I²C.



Slika 6.2: Priključevanje naprav na vodilo I²C

6.6 EEPROM

P89LPC932 ima vgrajenega 512 kB podatkovnega EEPROM-a, ki ga uporabljamo za shranjevanje nastavitvenih parametrov (v našem primeru razlika temperatur). Podatkovni EEPROM je dostopen preko SFR-jev, ki jih beremo, vpisujemo in brišemo zlogovno (vrstično polnjenje in polnjenje sektorja). Uporabnik lahko bere, piše in polni pomnilnik preko treh registrov SFR in ene prekinitve:

- Naslovni register DEEADR, se uporablja za naslovne bite od 7 do 0 (bit 8 se nahaja v registru DEECON)
- Kontrolni register DEECON se uporablja za naslavljanje bita 8, za nastavitve delovnega načina in za bit statusne zastavice.
- Podatkovni register DEEDAT, se uporablja za pisanje podatkov v ali branje podatkov iz podatkovnega EEPROM-a.

Register DEECON vsebuje naslednje bite: EADR, ECTL, HVERR in EEIF. Tabela 6.1 prikazuje vsebino registra DEECON.

Tabela 6.1: Register DEECON

7	6	5	4	3	2	1	0
EEIF	HVERR	ECTL.1	ECTL.0	-	-	-	EADR

DEECON: naslov: F1h
ni bitnega naslavljanja
izvor ponastavitve: katerakoli ponastavitev
po ponastavitvi: 0x00xxx0B

EEIF: Prekinitvena zastavica podatkovnega EEPROM-a. Postavi se po zaključitvi branja ali pisanja, resetiramo jo programsko.

HVERR: Napaka visoke napetosti. Kaže na napetostno napako med programiranjem ali brisanjem.

ECTL.1, ECTL.0: Izbira načina delovanja

00 način branja/pisanja zloga

10 polnjenje vrstice (64 zlogov)

11 polnjenje bloka (512 zlogov)

EADR: MSB naslova (bit 8) podatkovnega EEPROM-a. EADR.7-0 so v DEEADR registru.

Zlogovni način: V tem načinu pišemo in beremo podatke po en zlog naenkrat. Podatek je v registru DEEDAT, naslov pa v registru DEEADR. Vsako pisanje porabi približno 4 ms. Vsako branje zahteva tri strojne cikle po vpisu naslova v register DEEADR.

Polnjenje vrstice: V tem načinu je naslovljena vrstica (64 zlogov, naslovi DEEADR.5-0 se ne upoštevajo) polnjena z vzorcem podatka v DEEDAT. Za brisanje cele vrstice na 00h ali programiranje cele vrstice na FFh vpišemo 00h ali FFh pred polnjenjem vrstice. Vsako polnjenje vrstice traja približno 4 ms.

Polnjenje bloka: V tem načinu je vseh 512 zlogov napolnjenih z vzorcem v registeru DEEDAT. Za brisanje bloka na 00h ali programiranje na FFh vpišemo 00h ali FFh v register DEEDAT pred polnjenjem bloka. Pred uporabo tega ukaza moramo postaviti bit EADR na 1. Vsako polnjenje bloka traja približno 4 ms.

V vseh naštetih načinih strojna oprema po končani operaciji postavi bit EEIF. Prekinitve lahko omogočimo preko bita v registru IEN. Če sta EEIF in EA postavljena, se generira zahteva za prekinitve. Bit EEIF moramo brisati programsko.

Pokažimo še, kako na osnovi povedanega napišemo kodo za vpis in branje s programskim jezikom BASCOM (klicaj pred ukazom pomeni ukaz v zbirnem jeziku) :

Vpis:

```
!clr ie.7
```

```
Deecon = 1
```

```
Placevalue &H00 , Deedat
```

```
Placevalue &HF1 , Deeadr
```

```
Do
```

```
Eeif = Deecon
```

```
Loop Until Eeif > 127
```

```
Deecon = 1
```

```
Placevalue Dif , Deedat
```

```
Placevalue &HF1 , Deeadr
```

```
Do
```

Eeif = Deecon

Loop Until Eeif > 127

!setb ie.7

Return

Branje:

!clr ie.7

Deecon = 1

Placevalue &HF1 , Deeadr

Do

Eeif = Deecon

Loop Until Eeif > 127

Dif = Deedat

!setb ie.7

Return

7. KOMUNIKACIJA RS 232C

Mikrokrmilnik ima vgrajeno enoto za serijsko komunikacijo s protokolom UART. Protokol UART deluje v različnih načinih.

Način 0: Serijske podatke sprejema mikrokrmilnik preko priključka RxD ter oddaja preko priključka TxD. Osem bitov oddajamo ali sprejemamo tako, da se najprej odda ali sprejme LSB. Hitrost prenosa je fiksirana na 1/16 hitrosti ure CPE-ja.

Način 1: Preko RxD in TxD pošiljamo ali sprejemamo deset bitov: najprej start bit (logična 0), osem podatkovnih bitov (najprej LSB) in stop bit (logična 1). Ko so podatki sprejeti, se postavi bit RB8 v registru SFR SCON. Prenosna hitrost je spremenljiva ter določena s prekoračitvijo časovnika 1 ali generatorjem prenosne hitrosti.

Način 2: Preko RxD in TxD pošiljamo ali sprejemamo enajst bitov: najprej start bit (logična 0), osem podatkovnih bitov (najprej LSB), nastavljen deveti bit in stop bit (logična 1). Ko pošiljamo podatke, postavimo deveti bit na logično 1 ali logično 0 s pomočjo TB8 v registru SCON. Ko sprejemamo podatke, gre vrednost devetega bita v RB8 registra SCON, stop bit se ne shrani. Prenosna hitrost je nastavljiva z bitom v registru PCON na 1/16 ali 1/32 hitrosti ure CPE-ja.

Način 3: Preko RxD in TxD pošiljamo ali sprejemamo enajst bitov: najprej start bit (logična 0), osem podatkovnih bitov (najprej LSB), nastavljen deveti bit in stop bit (logična 1). Način 3 je popolnoma enak načinu 2, razlikuje se le pri prenosni hitrosti. Prenosna hitrost v načinu 3 je spremenljiva, določimo jo s prekoračitvijo časovnika 1 ali z generatorjem prenosne hitrosti. Protokol UART uporablja naslednje registre SFR: PCON, SCON, SBUF, SADDR, SADEN, SSTAT, BRGR1, BRGR0, BRGCON. Podrobneje opišemo samo Nastavitveni register SCON, ki je prikazan v tabeli 7.1.

Tabela 7.1: Vsebina registra SCON

7	6	5	4	3	2	1	0
SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI

SCON: naslov: 98h
bitno naslavljanje
izvor ponastavitve: katerakoli ponastavitev
po ponastavitvi: 00000000B

SM0/FE: Uporaba tega bita je določena z vrednostjo bita SMOD0 v registru PCON. Če je SMOD0 = 0, je ta bit uporabljen kot SM0 in skupaj z bitom SM1 določa način UART. Če je SMOD0 = 1, se ta bit uporablja kot zastavica za napako okvirja, bit FE. Bit FE postavi sprejemnik, kadar odkrije neveljaven stop bit. Ko je bit FE postavljen, ga lahko brišemo samo programsko.

SM1: Skupaj z bitom SM0 določa način UART

SM0	SM1	način UART	prenosna hitrost
0	0	način 0	ura CPE-ja/16
0	1	način 1	spremenljiva
1	0	način 2	ura CPE-ja/16 ali ura CPE-ja/32
1	1	način 3	spremenljiva

SM2: Omogoča večprocesorsko komunikacijo v načinu 2 in načinu 3. Če je v načinu 2 ali načinu 3 postavljen na 1, potem se bit RI ne bo postavil na 1 po sprejetem devetem bitu RB8=0. V načinu 0 naj bo SM2 = 0, v načinu 1 mora biti SM2 = 0.

REN: Omogoči serijsko sprejemanje. Postavimo in brišemo ga programsko.

TB8: Deveti podatkovni bit, ki bo poslan v načinu 2 in načinu 3. Postavimo in brišemo ga programsko.

RB8: Deveti podatkovni bit, ki je sprejet v načinu 2 in načinu 3. V načinu 1 je bit RB8 stop bit. Za način 0 je RB8 nedefiniran.

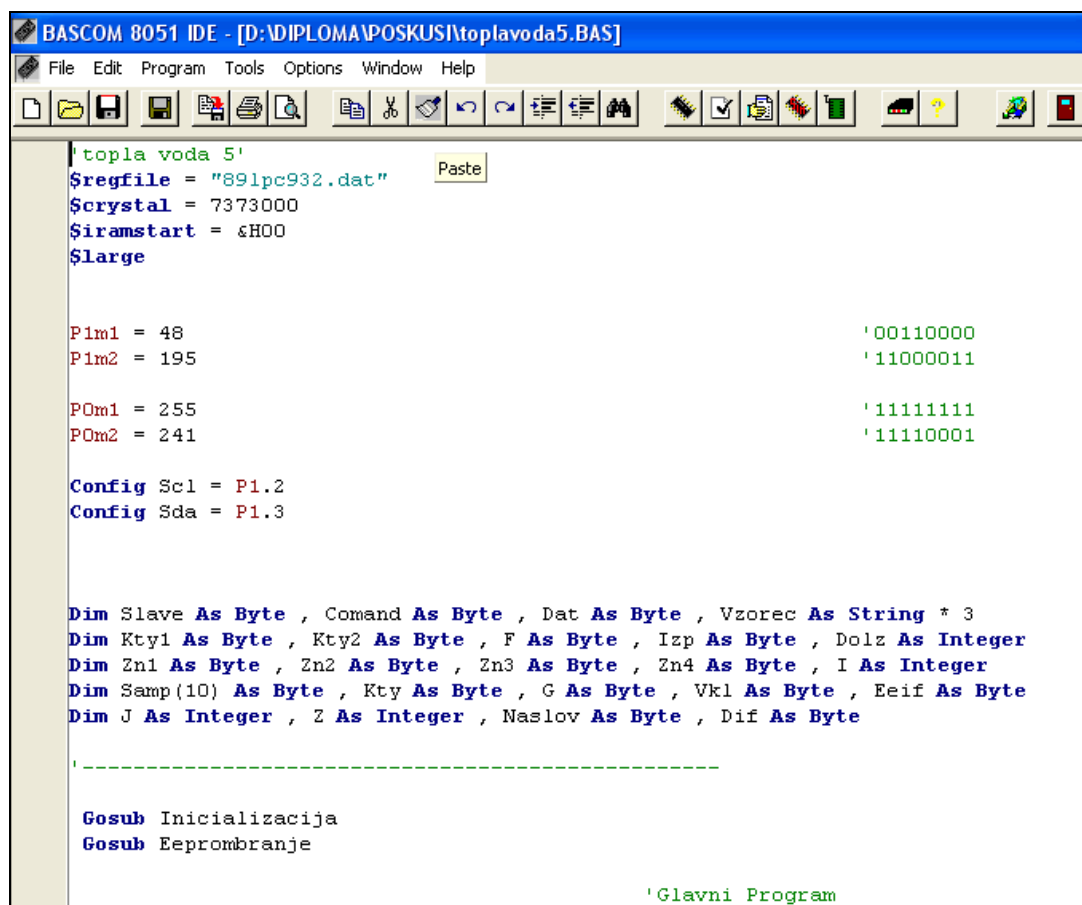
TI: Postavi jo strojna oprema ob koncu osmega bita v načinu 0 ali ob stop bitu v drugih načinih. Brišemo jo programsko.

RI : Postavi jo strojna oprema ob koncu osmega bita v načinu 0, na sredini stop bita v načinu 1. V načinu 2 in načinu 3 jo strojna oprema postavi v odvisnosti od vrednosti bita SMOD0. Če je SMOD0=0, se postavi na sredini devetega podatkovnega bita, če je SMOD0=1, se postavi na sredini stop bita. Zastavico brišemo samo programsko.

8. PROGRAMIRANJE MIKROKRMILNIKA

8.1 Razvojno okolje

Za programiranje mikrokrmilnika smo uporabili programski jezik Bascom, ki predstavlja tudi združeno razvojno okolje [1,2]. Združeno razvojno okolje Bascom-8051 je prikazano na sliki 8.1.



```
BASCOM 8051 IDE - [D:\DIPLOMA\POSKUSI\toplavoda5.BAS]
File Edit Program Tools Options Window Help
[Icons]
toplavoda5.BAS
$regfile = "891pc932.dat"
$crystal = 7373000
$ramstart = &H00
$large

P1m1 = 48
P1m2 = 195

P0m1 = 255
P0m2 = 241

Config Scl = P1.2
Config Sda = P1.3

Dim Slave As Byte , Comand As Byte , Dat As Byte , Vzorec As String * 3
Dim Kty1 As Byte , Kty2 As Byte , F As Byte , Izp As Byte , Dolz As Integer
Dim Zn1 As Byte , Zn2 As Byte , Zn3 As Byte , Zn4 As Byte , I As Integer
Dim Samp(10) As Byte , Kty As Byte , G As Byte , Vkl As Byte , Eeif As Byte
Dim J As Integer , Z As Integer , Naslov As Byte , Dif As Byte

-----

Gosub Inicializacija
Gosub Eeprombranje

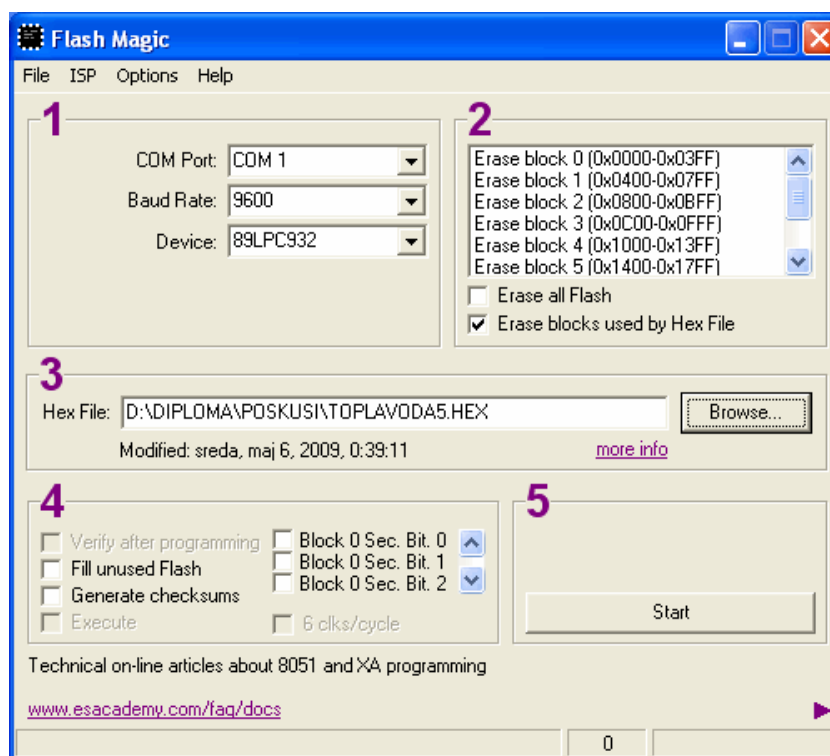
'Glavni Program
```

Slika 8.1: Združeno razvojno okolje Bascom-8051

Obstajajo tri različice tega programa:

- BascomLT (za družino mikrokrmilnikov 8051),
- Bascom 8051 (za družino mikrokrmilnikov 8051),
- Bascom AVR (za družino mikrokrmilnikov AVR).

Vse tri različice so precej podobne. Osnova vsem je klasični Basic, ki je nadgrajen s podporo za delo z mikrokrmilniki. Vsebuje ukaze in funkcije, s katerimi lahko nastavljam posamezne sklope mikrokrmilnika. Celotna arhitektura mikrokrmilnika je poimenovana s simbolnimi nazivi, ki poenostavijo programiranje in razumevanje načina delovanja teh relativno zahtevnih sklopov. Razvojno okolje Bascom vsebuje tudi simulator in programsko podporo za okrog deset različnih programatorjev, s pomočjo katerih prenesemo kodo iz računalnika v mikrokrmilnik. V našem primeru tega dela nismo uporabili, Bascom nima podpore za naš programator, vendar smo kodo enostavno prenesli v mikrokrmilnik s pomočjo programa "Flash Magic". Slika 8.2 prikazuje okno programa "Flash Magic". Program je označen z velikimi številkami, ki pomenijo zaporedje korakov nastavitvev. Prvi korak je izbira izhodnih vrat našega računalnika, prenosne hitrosti in tipa



Slika 8.2: Okno programa "Flash Magic".

mikrokrmilnika. V drugem koraku povemo, ali želimo pred programiranjem izbrisati celoten programski pomnilnik ali samo del programskega pomnilnika. Pod zaporedno številko tri vpišemo pot do izhodne datoteke našega programa, ki je zapisana v intelovem HEX formatu. Če želimo še preveriti pravilnost vnosa programa, imamo to možnost pri četrtem koraku. V zadnjem koraku pošljemo program v mikrokrmilnik.

8.2 Nastavitve mikrokrmilnika v razvojnem okolju

Inicializacija mikrokrmilnika se v združenem programskem okolju Bascom izvede na začetku programa in je obvezna, če želimo karkoli delati z mikrokrmilnikom. Za delovanje našega programa smo uporabili sledeče nastavitve:

Nastavitev registrske datoteke

Z nastavitvijo \$regfile = "file" prevajalniku povemo, kateri zapis naj uporablja, pri čemer je "file" ime, pod katerim je shranjen registrski zapis. Registrski zapis za nekaj zelo razširjenih mikrokrmilnikov najdemo kar v datoteki na disku, kjer je Bascom naložen v računalnik. V našem primeru imamo mikrokrmilnik novejšega tipa. To predstavlja problem, ki smo ga morali nujno rešiti, saj v nasprotnem primeru Bascom ne bi obvladal številnih funkcionalnosti mikrokrmilnika. Problema smo se lotili tako, da smo pregledali obstoječe registrske zapise ter navodila za pisanje registrskega zapisa v Bascomu. Na osnovi teh pregledov ugotovimo, da je tak zapis razdeljen v tri skupine (BIT, BYTE, MISC). Te tri skupine nam povedo, ali se registri SFR naslavljajo po posameznih bitih (BIT) ali po zlogih (BYTE), v tretji skupini se nahajajo razne nastavitve (MISC). Skupina razno (MISC) je sestavljena iz nekaj točk:

up kratko ime mikrokrmilnika,

IRAM razpoložljiva količina notranjega RAM-a (128 ali 256 zlogov),

org heksadecimalni naslov, kjer se lahko začne programska koda, to je 3 zloge po zadnji vpisani prekinitvi, ker bo zadnja prekinitve izvedla LJMP na ISR. LJMP potrebuje tri zloge.

I_xxx, kjer je **xxx** ime dodatne prekinitve. Ime ne sme biti daljše od šestih znakov. V primeru spodaj vidimo, da ima zadnji vpis prekinitve **I_TIMERI** heksadecimalni naslov 73, torej nastavimo **org =76** ($73 + 3 = 76$). Sami nastavljamo samo dodatne prekinitve, prekinitve, kot so INT0, INT1, TIMER0, TIMER1 in SERIAL prevajalnik že obvlada.

clockdiv faktor deljenja oscilatorja, če ga ne nastavimo, je izbrana vrednost 12. Nekateri mikrokrmilniki imajo faktor deljenja oscilatorja 6, 4 itd.

Primer registrskega zapisa za naš mikrokrmilnik:

[BIT]

ACC = E0 , 00

B = F0 , 00

P0 = 80 , FF

P1 = 90 , FF

P2 = A0 , FF

[BYTE]

SP = 81 , 07

DEECON = F1 , 00

DEEDAT = F2 , 00

DEEADR = F3 , 00

[MISC]

I_BROWN = 2B

I_I2C = 33

I_KBI = 3B

I_CMF2 = 43

I_WDT = 53

I_CMF1 = 63

I_TIMERI = 73

org = 76

up = P89LPC932

IRAM = 256

clockdiv = 2

Seveda je tu izpisanih le nekaj naslovov registrov SFR, ker je celoten izpis predolg. Vrednost za vejico je vrednost registra po resetu, naprimer pri kazalcu sklada SP je za vejico zapisana heksadecimalna vrednost 07.

Nastavitev oscilatorja

Frekvenco notranjega oscilatorja nastavimo z izjavo \$crystal = var, pri čemer je var zelena frekvenca kristala.

Nastavitev hitrosti prenosa

Hitrost prenosa preko RS232 vmesnika nastavimo z izjavo \$boud = var, pri čemer je var zelena hitrost prenosa. Pri tej nastavitvi je zelo pomembno najti pravo razmerje med frekvenco oscilatorja in želeno hitrostjo prenosa. Hitrost prenosa se dobi z deljenjem sistemske ure, tako lahko dobimo najpreciznejše vrednosti, ko uporabljamo kristal s frekvenco 11.0592 MHz. Uporabljamo lahko tudi druge kristale, vendar dobljena hitrost prenosa ne bo nikoli natančno 2400 ali 4800 baudov, višje hitrosti pa so skoraj nemogoče.

```

TOPLAVODA5 - Notepad
File Edit Format View Help
Compiler : BASCOM 8051 LIBRARY V 2.11
Processor : 87LPC932
Report : TOPLAVODA5
Date : 05-05-2009
Time : 15:57:21|

Baud Timer : 1
Baudrate : 2399
Frequency : 11059000
Clock div. : 2
ROM start : &H0
RAM start : &H0
LCD mode : 4-bit
StackStart : &H9A
Used ROM : &H1B67 7015 (dec) > doesn't fit into selected codesize : 2048

-----
Variable Type Address(hex) Address(dec)
-----
ERR Bit 20.4

SLAVE Byte 0032 50
COMAND Byte 0033 51
DAT Byte 0034 52
KTY1 Byte 0039 57
KTY2 Byte 003A 58
F Byte 003B 59
IZP Byte 003C 60
ZN1 Byte 003F 63
ZN2 Byte 0040 64
ZN3 Byte 0041 65
ZN4 Byte 0042 66
SAMP Byte (10) 0045 69
KTY Byte 004F 79
G Byte 0050 80
VKL Byte 0051 81
EEIF Byte 0052 82
NASLOV Byte 0057 87
DIF Byte 0058 88

DOLZ Integer 003D 61

```

Slika 8.3 : Poročilo

Če prenos ne deluje, najprej pogledamo v generirano datoteko s končnico .rpt, to je poročilo. Primer poročila prikazuje slika 8.3. V tej datoteki najdemo točno hitrost, ki jo je generiral prevajalnik. Če hitrost odstopa za več kot 2,5 %, bomo najverjetneje imeli napačno prenesene vrednosti, komunikacija ne bo delovala.

Nastavitev dolgih skokov

Pove prevajalniku, da se mora uporabiti izjava LCALL. Za klic podprograma se interno uporablja izjava ACALL. Instrukcija ACALL potrebuje dva zloga, medtem ko potrebuje LCALL tri zloge. ACAL lahko naslavlja podprograme z maksimalnim odmikom 2048 vrstic. Z izjavo \$large se lahko odmik poveča na naslovni prostor 64 kB.

Nastavitev vhodno-izhodnih vrat

Za nastavljanje vhodno-izhodnih vrat je potrebno vpisati podatke v dva registra in sicer Pxm1 ter Pxm2. Spremenljivka x predstavlja številko vrat, le-teh ima mikrokrmilni ponavadi več. Kot primer za vrata P0 povejmo, da vpisujemo nastavitvene podatke v

register P0M1 in P0M2. Mi smo uporabili dvoje vrat, in sicer vrata 0 ter vrata 1 z nastavitvijo:

P0M1 = 0 '00000000' P1M1 = 50 '00110010'

P0M2 = 255 '11111111' P1M2 = 193 '11000001'

Sedaj še razložimo, kako pridemo do teh vrednosti, to pa je najlažje kar s tabelo 8.1, ki ponazarja nastavitve vrat 1. V tabeli ni predstavljen način izhodnih ponornih vrat, tega nastavimo tako, da na primer namesto 0 vpišemo 1 v stolpec P1M1 pri vratih P1.7. S tem bi dobili novo vrednost P1M1=178 (50+128=178) in vrata P1.7 ne bi bila več nastavljena na izhodna vrata z dvižnim uporom, ampak na izhodna ponorna vrata.

Tabela 8.1: Nastavljanje vhodno-izhodnih vrat

vrednost utežnega bita	P1M1	P1M2	oznaka vrat	nastavitev izhodnih vrat
1	0	1	P1.0	izhodna vrata z dvižnim uporom
2	1	0	P1.1	vhodna vrata
4	0	0	P1.2	na videz dvosmerna vrata
8	0	0	P1.3	na videz dvosmerna vrata
16	1	0	P1.4	vhodna vrata
32	1	0	P1.5	vhodna vrata
64	0	1	P1.6	izhodna vrata z dvižnim uporom
128	0	1	P1.7	izhodna vrata z dvižnim uporom
SKUPAJ (max256):	50	193	/	/

Nastavitev serijskega protokola I²C

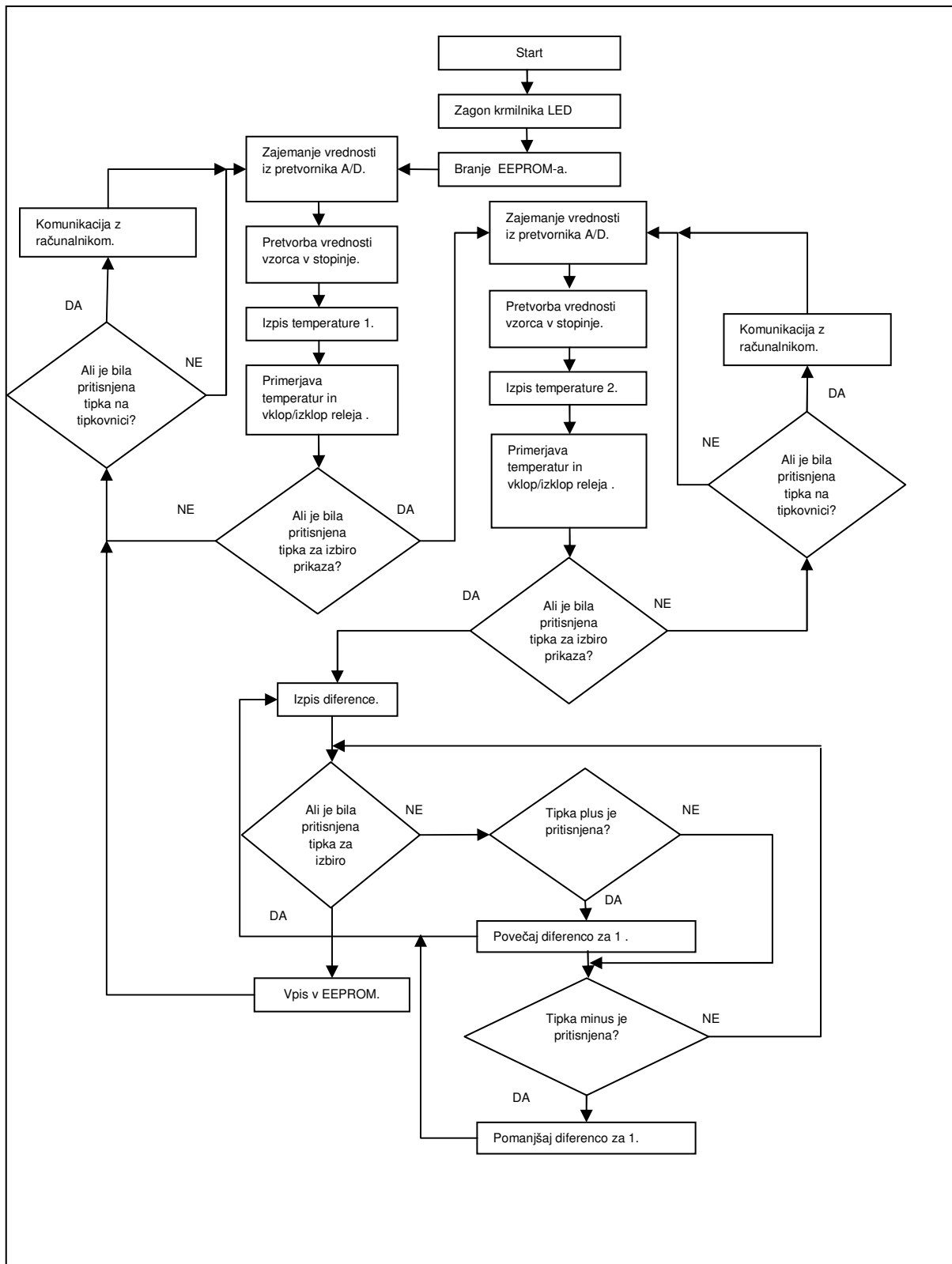
Vrata, ki se bodo uporabljala za komunikacijo I²C s suženjskimi čipi določita nastavitvi:

Config Scl = P1.2

Config Sda = P1.3

Pri tej nastavitvi moramo biti pazljivi na to, da smo že predhodno pravilno določili način delovanja izhodnim vratom. Oboja vrata, tako za linijo SCL kot za linijo SDA, nastavimo na način navidez dvosmernih vrat.

8.3 Programska programa

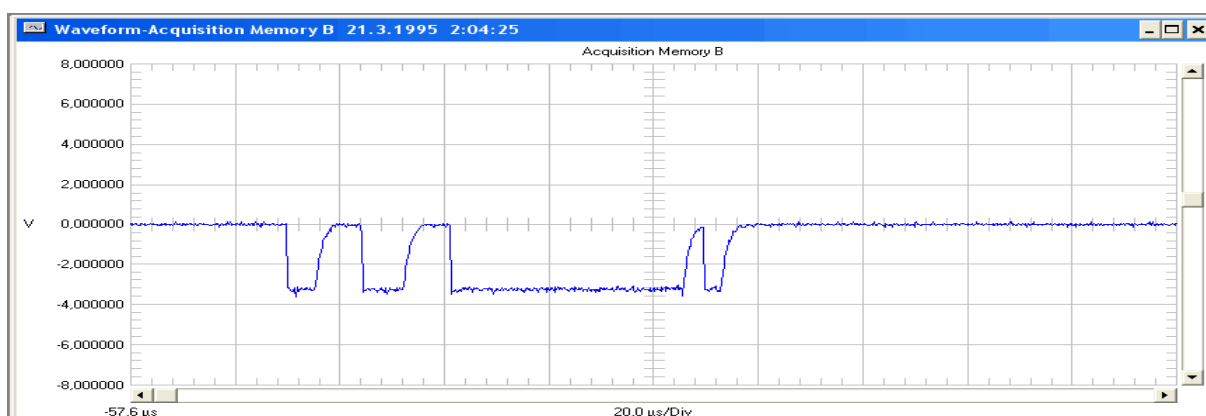


Slika 8.4: Odločitveni diagram

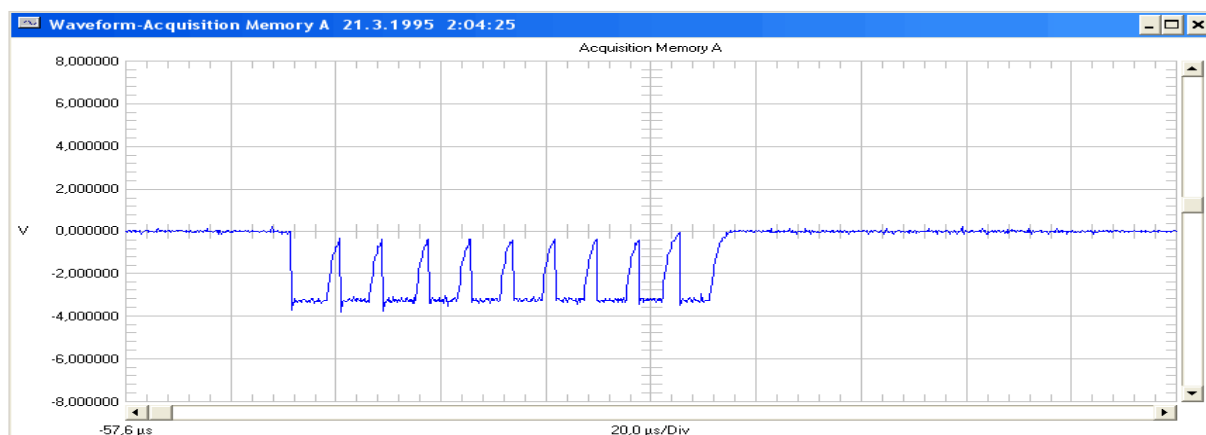
Kodo programa sestavlja glavni program in več podprogramov. Z glavnim programom narisanim v odločitvenem diagramu, se koda izvaja glede na pritisk tipk. Poglejmo kako je glavni program sestavljen. Od zagonu mikrokrmilnika glavni program najprej pokliče podprogram za zagon krmilnika LED, ki pošlje na vodilo I²C ukazni zlog in nastavitvene zloge krmilniku diod LED. Naslednji podprogram, ki se izvede je branje EEPROM-a. S tem pod programom preberemo zadnje shranjeno vrednost, ki predstavlja diferenco med senzorjema. Sedaj s skokom v podprogram za zajemanje vrednosti preberemo kodo, ki ustreza določeni napetosti na pretvorniku A/D. Vrednost, ki smo jo prebrali naslednji podprogram pretvori v stopinje (°C). Pretvorjeno vrednost naslednji podprogram izpiše na prikazovalniku LED. Če je razlika med temperaturoma obe senzorjev večja od želene, vezje vklopi rele in s tem vodno črpalko. Od tu naprej deluje glavni program v zanki in se odziva na pritiske tipke za izbiro prikaza, tipk za nastavljanje temperaturne difference in tipk tipkovnice, če priključimo vezje na računalnik. Če pritisnemo tipko za izbiro prikaza, glavni program ponovno vzame oba vzorca temperature, ju pretvori v stopinje in izpiše vrednost na drugem senzorju. Naslednji pritisk tipke prikaže nastavljeno diferenco, ki se uporablja pri izračunu za vklop releja. To je parameter, ki ga lahko spreminjamo s pritiskom na plus in minus tipko. S pritiskom na tipko za izbiro prikaza se novo nastavljena vrednost shrani v EEPROM, program nato skoči nazaj na prikaz prve vrednosti na prvem senzorju. Glavni program samodejno skoči iz podprograma za vnašanje difference po določenem času, tudi če smo pozabili pritisniti tipko za izbiro prikaza. Če pritisnemo katerikoli tipko na tipkovnici računalnika, program izpiše v oknu hyperterminala obe temperaturi in nas vpraša, če želimo nastaviti diferenco. Komunikacija se zaključi, ko vnesemo ali ne diferenco za vklop releja.

9. REZULTATI

Naše vezje je zgrajeno iz mikrokrmilnika in dveh integriranih vezij, ki sta priljučeni na mikrokrmilnik z vodilom I²C. Želimo preveriti, če realno dogajanje na vodilu I²C usteza opisom protokola, določili bomo tudi frekvenco ure, ki nam določa hitrost prenosa bitov. Za merjenje uporabimo osciloskop Scopemeter 123 proizvajalca Fluke. Na vodilo I²C pošljemo 8-bitni vzorec vrednosti A0 heksadecimalno. Signal merimo hkrati na liniji SDA in SCL, tako da priključimo vhod A osciloscopa na upor linije SDA in vhod B na upor linije SCL. Oba upora imata enako vrednost 3,7k Ω . Osciloskop lahko priljučimo na osebni računalnik, zato lahko signal posnamemo. Dobili smo rezultat, ki ga prikazujeta sliki 9.1 in 9.2.



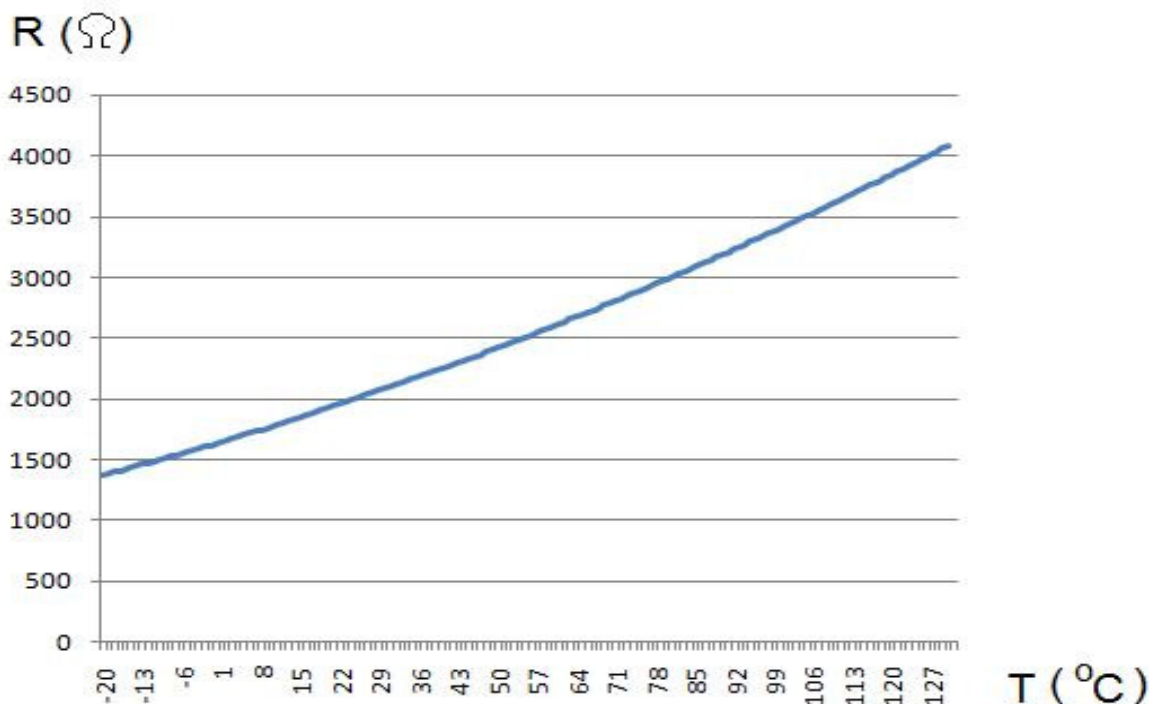
Slika 9.1: Meritev na vodilu SDA.



Slika 9.2: Meritev na vodilu SCL.

Gospodar začne prenos s pogojem START, tako da povleče linijo SDA z visokega na nizek nivo, medtem ko je linija SCL na visokem nivoju, kar se vidi iz primerjave slike 9.1 in slike 9.2. Na sliki vidimo tudi potrditveni bit, ta je vsak deveti bit ure, ki potrdi sprejem podatkovnega zloga. Gospodar konča prenos s pogojem STOP, tako da povleče linijo SDA z nizkega na visok nivo, medtem ko je linija SCL na visokem nivoju. Amplituda signala znaša 3,3 V, frekvenca prenosa je 125 kilobitov na sekundo.

Ocenili bomo še napako, vzorčenja zaradi nelinearne karakteristike senzorja na osnovi izračunanih vrednosti. Na sliki 9.4 je graf spreminjanja razlike napetosti na mostičku v odvisnosti od temperature na senzorju. Opazimo, da je karakteristika na sliki 9.4 bolj linearna kot karakteristika spreminjanja upornosti senzora od temperature, slika 9.3. Razlog je v spreminjanju toka, v mostičku, ki se z upornostjo senzorja tudi spreminja. Na sliki 9.4 vidimo, da je napetost 0V pri temperaturi 65 °C. To smo določili z izbiro uporov v merilnem mostičku. Največja napaka je na skrajnih točkah karakteristike na sliki 9.4, pri temperaturi 130°C in -20 °C. V tretjem poglavju je izračun za 25 °C. S pomočjo isti enačb smo prišli do vrednosti na zgornjem robu 129,1 °C in -18,9 °C na spodnjem robu. Če grobo ocenimo napako znaša 1%. Taka natančnost je za našo napravo zadovoljiva.



ERROR: stackunderflow
OFFENDING COMMAND: ~

STACK: