ŠOLSKI CENTER CELJE

Srednja šola za elektrotehniko in kemijo

MIKROKRMILNIŠKI GENERATOR FREKVENCE

(raziskovalna naloga)

Mentor:

prof. Marko Vrečko

Avtor:

Dejan Levec, E-2.E

Celje, marec 2009

KAZALO

1. POVZETEK IN KLJUČNE BESEDE……..... 2
1.1 Povzetek/ Summary.....2
1.2 Ključne besede/ Keywords..... 3
2. UVOD 4
2.1 Predstavitev raziskovalnega problema 4
2.2 Teze/Hipoteze 4
2.3 Opis raziskovalnih metod 4
3. POTEK IZDELAVE 5
3.1 Mikrokrmilnik Atmel Tiny 45 (ATTiny45) 5
3.2 Programiranje mikrokrmilnika 8
3.3 USB (Universal Serial Bus) protokol 8
3.4 Implementacija USB (Universal Serial Bus) protokola 8
3.5 Uporaba USB protokola na operacijskem sistemu Linux in Mac OS 11
3.6 Uporaba USB protokola na operacijskem sistemu Windows 12
3.7 Vezje 13
3.8. Koda za utripanje LED diode 14
4. RAZPRAVA 15
5. ZAKLJUČEK 16
6. VIRI IN LITERATURA 17
7. ZAHVALA 18
8. KAZALO SLIK 19
9. KAZALO TABEL 19
10. PRILOGA 20

1. POVZETEK IN KLJUČNE BESEDE

1.1 Povzetek / Summary

Raziskovalna naloga predstavlja generator frekvence, ki mu nastavljamo frekvenco pravokotnega signala preko računalniškega programa. Vezje je na računalnik povezano preko USB kabla. Sistem krmili mikrokrmilnik Atmel Tiny 45 (ATtiny45). Za programiranje mikrokrmilnika sem uporabil programski jezik C. Generator frekvence oz. funkcijski generator sem uspešno izdelal s pomočjo AVR-USB programske kode in programske knjižnice libusb (ter njene Windows verzije). Vezje porabi malo energije in je tako skladen z USB standardom.

This research work focuses on the frequency generator on which we adjust the frequency of square signal over computer software. Integrated circuit is connected to the computer via the USB cable. The system is controlled by Atmel Tiny 45 (ATtiny45) microcontroller. To programme the microcontroller I applied the programming language C. I successed in making of frequency generator or function generator with the help of AVR-USB programming code and programming library libusb (and its Windows version). The integrated circuit uses very little electrical energy and is in accordance with the USB standard.

1.2 Ključne besede/ Keywords

Atmel – družina mikrokrmilnikov

ATtiny45 – tip mikrokrmilnika

USB – univerzalno serijsko vodilo

Linux – operacijski sistem

Mac – operacijski sistem

Atmel – microcontrollers

ATtiny45 – type of microcontroller

USB – universal serial bus

Linux – operating system

Mac - operating system

2. UVOD

2.1 Predstavitev raziskovalnega problema

Izdelave funkcijskega generatorja sem se lotil, ker sem ga pri delu z elektroniko večkrat potreboval, v trgovini pa je cena zanj previsoka. Za projekt sem izbral Atmelov mikrokrmilnik, saj sem ga že večkrat uspešno uporabil pri raznih projektih.

2.2 Teze/Hipoteze

- Funkcijski generator je enostaven za uporabo.

- Nastavlja se preko računalnika.

2.3 Opis raziskovalnih metod

Projekt sem začel z risanjem skic. Ko je bil načrt sestavljen, sem izbral potreben mikrokrmilnik ter programski jezik, s katerim ga bom sprogramiral. Problemi so se pojavili pri komunikaciji računalnika z mikrokrmilnikom. Potrebno je bilo izdelati računalniški program za nastavljanje frekvence.

3. POTEK IZDELAVE

Prvi korak je bil izbira mikrokrmilnika. Ko sem ga izbral, sem se lotil izdelovanja programa za mikrokrmilnik. Po izdelavi vezja, sem naredil tudi programsko opremo za računalnik.

3.1 Mikrokrmilnik Atmel Tiny 45 (ATtiny45)

ATtiny45 je eden izmed manjših mikrokrmilnikov od Atmela, vendar vseeno ponuja skoraj vse, kar ponujajo večji mikrokrmilniki.

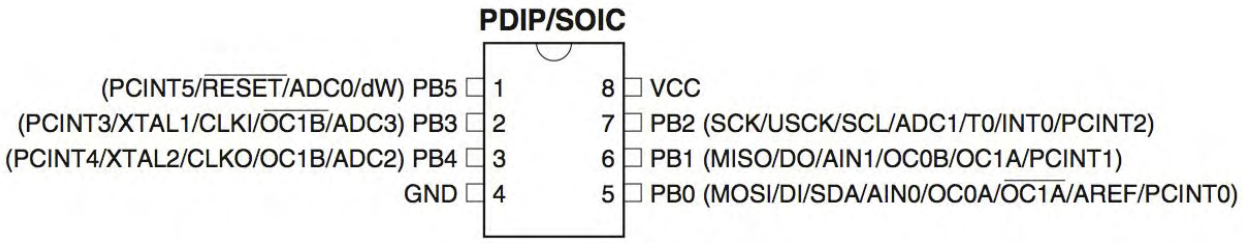
Slika 1: Mikrokrmilnik ATtiny 45

ATtiny45 ima:
- 4 kB spomina
- 256 B EEPROM
- 256 SRAM
- 8 bitni časovnik/števec
- 8 bitni visoko hitrostni časovnik/števec
- 10 bitni ADC (analog to digital converter)
- Watchdog časovnik
- 6 I/O linij
- majhno porabo
- vgrajeni oscilator

Za ta projekt sem izbral ta mikrokrmilnik, ker mi ponuja vse kar potrebujem. Za delovanje vezja sem potreboval samo 1 vhod in 2 izhoda.

Mikrokrmilnik ima vgrajen oscilator, ki ima 1,1% odstopanje, kar je za potrebe tega projekta premalo, zato sem uporabil zunanji quartz kristal s frekvenco 16MHz.

ATtiny45 ima zelo majhno porabo električne energije, kar nam zelo koristi, ker je USB priključek omejen z največ 500mA toka.



Slika 1: Opis kontaktov na mikrokrmilniku

AVR mikrokrmilniki so 8-bitni, njihova arhitektura pa je bila razvita v letu 1996. AVR je eden prvih mikrokrmilnikov, ki imajo vgrajen pomnilnik. Prvi množično uporabljen mikrokrmilnik je bil AT90S1200.

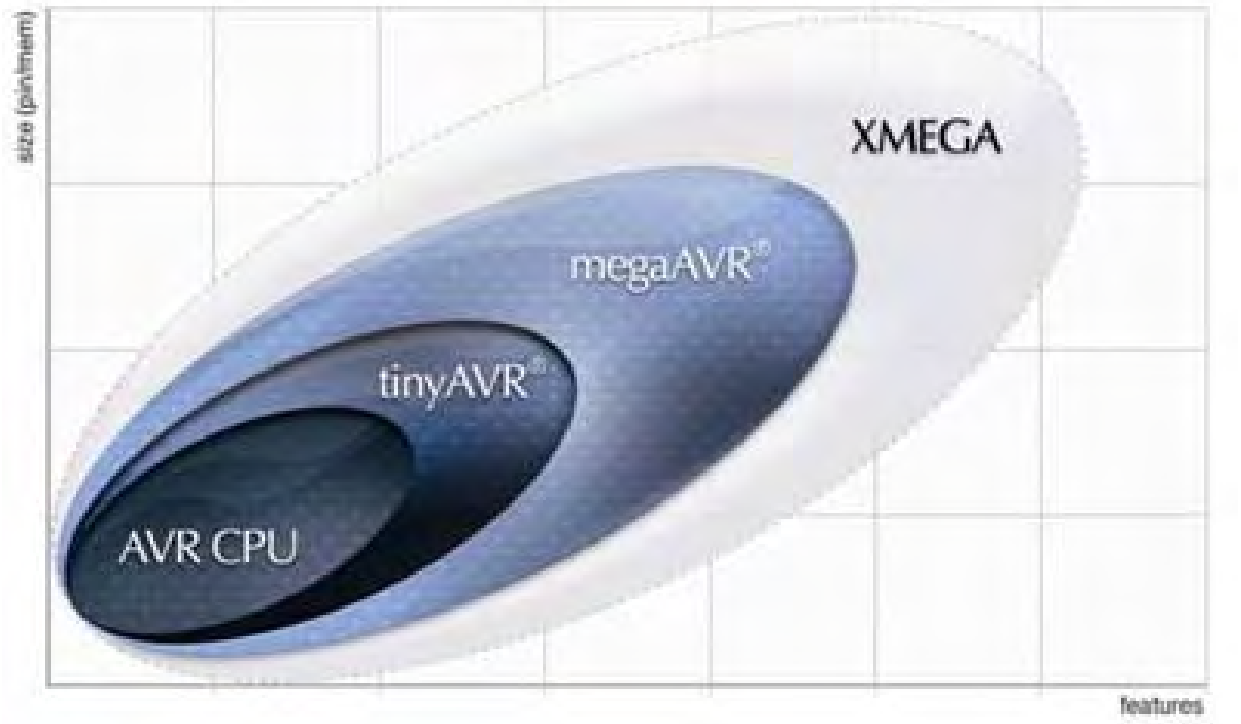
AVR mikrokrmilnike lahko programiramo tudi v programskem jeziku C, s pomočjo katerega si olajšamo pisanje kode. AVR mikrokrmilniki pa so tudi zelo varčni.

Podpirajo:
- delovanje na 1,8 do 5,5 V napetosti
- do 6 načinov spanja
- hitro zbujanje iz spanja
- spreminjanje frekvence iz kode

Dodatno so naredili še zelo varčne mikrokrmilnike, ki jih lahko napajamo z napetostjo 0,7V.

Skupine AVRjev

- tinyAVR - ATtiny serija
  - 1 - 8 kB spomina
- megaAVR - ATmega serija
  - 4 - 256 kB spomina
  - 28 - 100 priključkov
  - najbolj popularna serija
  - podpira več stvari kot ATtiny serija
- XMEGA - ATxmega serija
  - 16 - 384 kB spomina
  - 44 - 64 - 100 priključkov
  - namenjena večjim projektom



Slika 2: Primerjava različnih skupin AVR mikrokontrolerjev

Uporaba

AVR mikrokrmilniki se med drugim uporabljajo tudi v avtomobilski industriji, poleg tega se uporabljajo pri mnogih izdelkih znanih podjetij.

Microsoft je AVR mikrokrmilnik uporabil v kontrolerjih za igralno konzolo Xbox, Lego pa jih je vgradil v NXT opremo.

32-bitni AVRji

V letu 2006 je Atmel izdal 32 bitne mikrokrmilnike, ki ponujajo tudi avdio in video procesiranje. 32-bitni AVRji so primerni z ARM procesorji.

3.2 Programiranje mikrokrmilnika

Vsa koda za mikrokrmilnik je bila prevedena z AVR-GCC verzije 3.4.6, program za upravljanje preko računalnika za operacijski sistem Linux/Mac OS je bil napisan z Xcode
in preveden z GCC 4.0.1.
Dinamična knjižnica DLL za operacijski sistem Windows je bila napisana z Dev-C++ in prevedena s pomočjo MinGW prevajalnika, program za upravljanje pa je bil napisan s pomočjo Visual Studio 2008 Express Edition.

Za programiranje sem uporabil odprtokodni program avrdude, ki ga lahko brezplačno namestimo iz njihove uradne spletne strani.

Za nalaganje kode na mikrokrmilnik sem uporabil programator, ki temelji na ATmega8 in
je preko USB - RS232 FTDI čipa povezan na USB priključek na računalniku.

3.3 USB (Universal Serial Bus) protokol

USB protokol je bil narejen z namenom, da olajša povezovanje naprav na osebni računalnik. Za uporabnika je USB vmesnik preprosto, saj mu o delovanju le tega ni potrebno ničesar vedeti. Napravi ni potrebno določati naslovov vrat in podobnih zadev.
Za vse ostalo poskrbi računalnik in operacijski sistem skupaj z gonilniki. V ozadju pa je protokol precej kompleksen.
Za uspešno komunikacijo je potreben gostitelj (osebni računalnik) in naprava. Gostitelj razpozna napravo, ji dodeli zaporedno številko na vodilu ter poišče zanj, najbolj primerne gonilnike za to napravo.

Vsako napravo operacijski sistem identificira z posebej njej dodeljeno številko proizvajalca (Vendor-ID) in številko izdelka (Product-ID). Kombinacije teh številk je za uporabo potrebno zakupiti.

Na računalniku je komunikacija več plastna:
Aplikacija, ki uporablja napravo, priključeno na USB priključek, komunicira z gonilniki naprave, gonilniki dostopajo do sistemskega vodila, kjer je priključeno tudi USB vodilo.

Vsaka naprava lahko ima svoj gonilnik ali pa uporablja splošni gonilnik, če mu ustreza. Takšne naprave so ponavadi tipkovnice, miške ter USB ključi.

Na vodilo je lahko hkrati povezano do 127 naprav. Podatkovni tok po vodilu je serijski.

3.4 Implementacija USB (Universal Serial Bus) protokola

Vezje se preko USB vrat priključi na računalnik. Preko njega se tudi napaja ter prenaša podatke med programom na računalniku in mikrokrmilnikom.

Ker izbrani mikrokrmilnik ni imel strojne opreme za komunikacijo preko USB protokola, sem moral uporabiti programsko kodo AVR-USB od podjetja Objective Development, za katero ne potrebujemo posebne strojne opreme.

Ker za implementacijo in uporabo USB protokola potrebujemo dovoljenje USB Implementers Forum, Inc., nam pri Objective Development dovolijo brezplačno uporabo njihove Vendor-ID in Product-ID kode za nekomercialne in odprtokodne projekte.

AVR-USB je odprtokodna programska implementacija USB protokola in deluje na vseh AVR mikrokrmilnikih, ki imajo vsaj 2 kB spomina in 128 bajtov delovnega pomnilnika (RAM).

Privzeto ponuja pošiljanje do 254 bajtov, lahko pa implementiramo funkcijo usbFunctionWrite preko katere lahko pošiljamo poljubno velike bloke podatkov.

Funkcija usbFunctionWrite s pomočjo katere lahko na mikrokrmilnik prenesemo bloke podatkov večje od 254 bajtov:

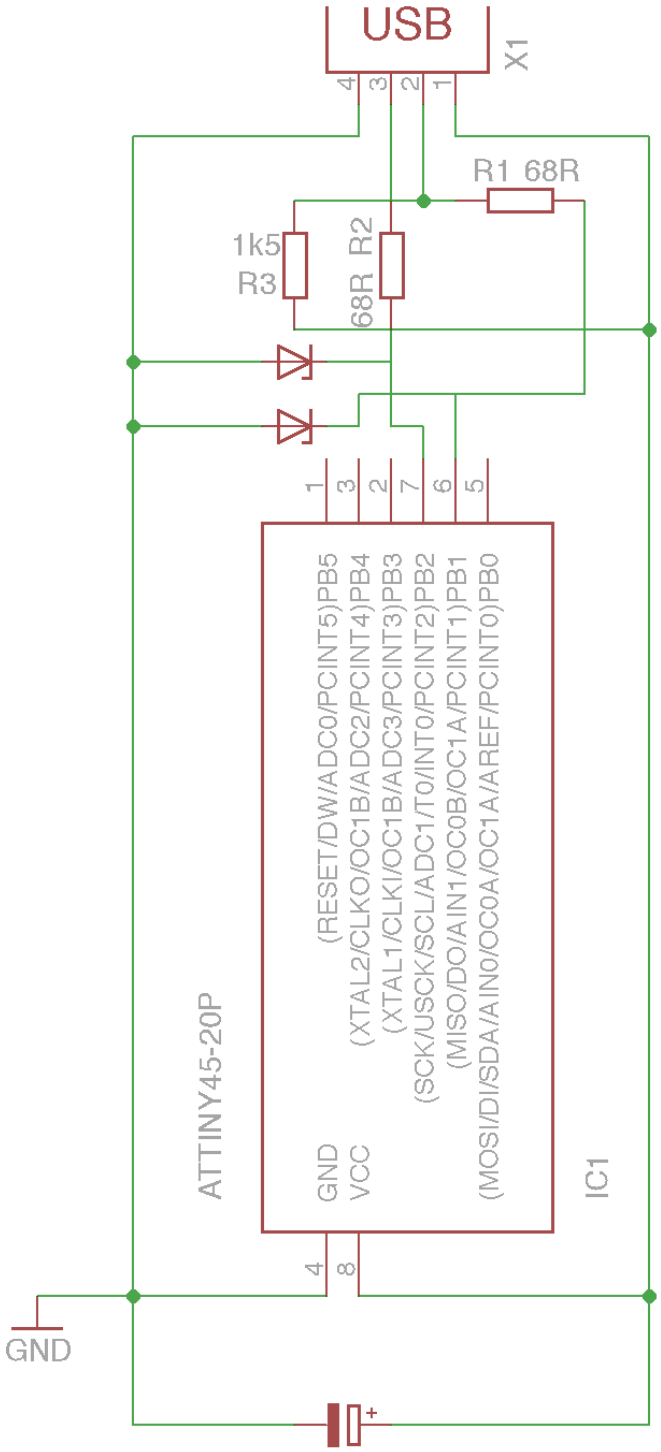
```
uchar usbFunctionWrite(uchar *data, uchar len) {
                                        uchar i;
                                        if(len > bytesRemaining)
                                              len = bytesRemaining;
                                        bytesRemaining -= len;
                                        for(i = 0; i < len; i++)
                                              buffer[currentPosition++] = data[i];

                                        return bytesRemaining == 0;
}
```

Strojni del

Za delovanje AVR-USB potrebujemo 2 upora z vrednostjo od 10 do 100 ohmov, ki sta vezana zaporedno na komunikacijski žici, na D- komunikacijski žici potrebujemo še dodaten 1,5kΩ upor.

Ker nam USB vrata ponujajo na izhodu napetost 5V, USB protokol pa določa komunikacijo na 3,3V, potrebujemo dve dodatni diodi, ki sta vezani med komunikacijskimi žicami in GND priključkom.



Slika 3: Povezava Atmel mikrokontrolerja na USB priključek pri uporabi AVR-USB.

3.5 Uporaba USB protokola na operacijskem sistemu Linux in Mac OS

Za komuniciranje preko USB protokola na operacijskemu sistemu Linux nam večina Linux distribucij ponuja že vgrajeno knjižnico odprtokodno knjižnico libusb za katero ne potrebujemo dodatnih gonilnikov.

Na operacijskem sistemu Mac OS lahko to knjižnico dobimo z namestitvijo Applovega razvojega orodja Xcode, ki je brezplačno priložen na vseh Mac OS DVD medijih poleg tega pa ga je mogoče brezplačno prenesti iz njihove spletne strani.

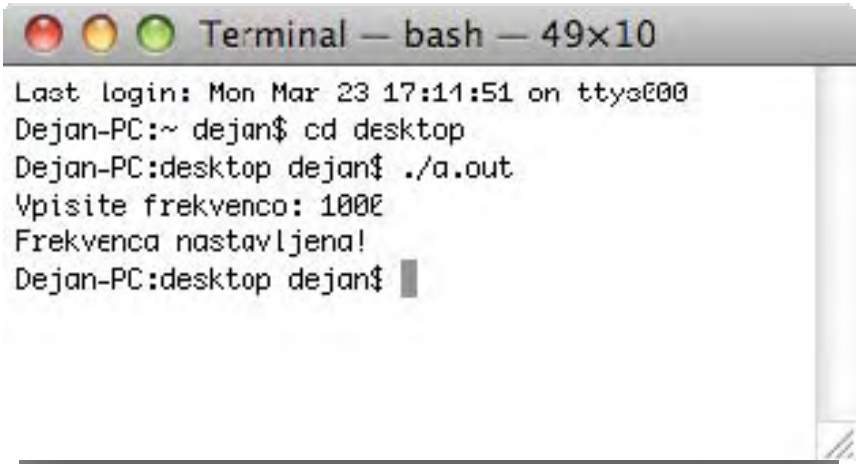
Uporaba te knjižnice je enaka tako na Linuxu kot na Mac OSu in je primerna za uporabo z C/C++ programskima jezikoma.

Povezava z napravo:
usbOpenDevice(&handle, vid, vendor, pid, product, NULL, NULL, NULL)

vid - izračunana vrednost iz številke proizvajalca
vendor - številka proizvajalca
pid - izračunana vrednost iz številke produkta
product - številka produkta

Pošiljanje podatkov na napravo:
usb_control_msg(handle, USB_TYPE_VENDOR | USB_RECIP_DEVICE | USB_ENDPOINT_OUT, 200, 0, 0, mybuff, sizeof(mybuff), 5000);

S pomočjo te kode napravi pošljemo niz mybuff z dolžino 5000 znakov.



Slika 4: Prikaz uporabe programa na operacijskemu sistemu Mac

3.6 Uporaba USB protokola na operacijskem sistemu Windows

Za komuniciranje preko USB protokola na operacijskem sistemu Windows lahko uporabimo odprtokodno knjižico LibUsb-Win32, ki je prirejena libusb knjižnica in deluje tudi na Windowsu.

Na Windows sistemu je obvezna uporaba gonilnikov za vse naprave, ki niso kompatibilne z HID (human interface device). Zato lahko v tem primeru uporabimo gonilnike, ki so jih napisali za LibUsb-Win32 knjižnico.

Za ta frekvenčni generator je bilo potrebno napisati dinamično knjižnico DLL (dynamic link library) za komunikacijo med gonilnikom od naprave ter programom, preko katerega bomo nastavili frekvenco.

Dinamična knjižnica DLL je napisana v programskem jeziku C++, program za nastavitev frekvence pa v C# .NET 3.5.

3.7 Vezje

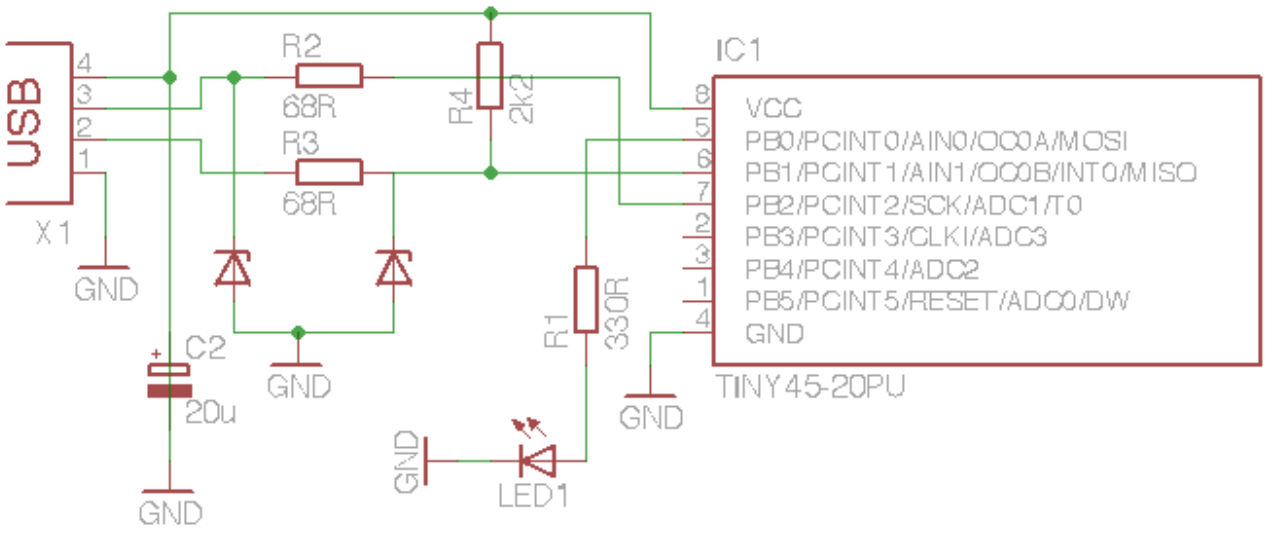
Glavni element vezja je mikrokrmilnik ATtiny45, ki je povezan na USB priključek, preko katerega se celotno vezje napaja ter preko katerega računalnik komunicira z mikrokrmilnikom.

Za komuniciranje sta potrebna dva od 10 do 100Ω upora, 2.2kΩ upor ter dve diodi.

Za glajenje napetosti je potreben še gladilni kondenzator z vrednostjo 20 uF. Če gladilnega kondenzatorja ne bi bilo, bi bila komunikacija motena in bloki podatkov, ki se prenašajo bi se lahko poškodovali.

Dodatno potrebujemo še 330Ω upor za priključitev 2V svetleče diode LED (light-emitting diode).

Za potrebe implementacijske kode AVR-USB potrebuje še zunanji oscilator s frekvenco 16 MHz.



Slika 5: Vezje funkcijskega generatorja

3.8 Koda za utripanje LED diode

```
ISR(TIMER0_OVF_vect) {
                    if(timerX > (float)((float)TIMER_SPEED/(float)1000 *
(float)speed)) {
                        if(timerStatus == 1) {
                            PORTB |= _BV(OUTPUT_BIT);
                            timerStatus = 0;
                        }else{
                            PORTB &= ~_BV(OUTPUT_BIT);
                            timerStatus = 1;
                        }
                        timerX = -1;
                    }
                    timerX++;
}
```

Za uporabo časovnika moramo nastaviti tudi pravilno hitrost.

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/(No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

Tabela 1: Tabela z podatki o nastavitvah hitrosti časovnika

4. RAZPRAVA

Osnova vezja je mikrokrmilnik ATtiny 45. Najprej sem napisal programsko kodo za mikrokrmilnik, nato sem naredil vezje. Kasneje sem napisal še programsko kodo za računalnik.

Ko napravo priključimo na računalnik, operacijskemu sistemu pošlje svoje podatke, da jo lahko le-ta identificira ter naloži potrebne gonilnike, če so le-ti potrebni.
Naprava med tem zažene časovnik in LED dioda začne utripati s privzeto frekvenco,
koda za USB implementacijo pa čaka na prejetje podatkov.

Ko zaženemo potreben program za komunikacijo z mikrokrmilnikom, program najprej preveri prisotnost USB naprave, in če jo najde se začneta sporazumevati. Takrat se vzpostavi povezava in naprava čaka na prejetje podatkov o spremembi hitrosti.

Ko s pomočjo programa nastavimo določene frekvenco, program to sporoči mikrokrmilniku in le-ta spremeni hitrost časovnika.

5. ZAKLJUČEK

V seminarski nalogi sem predstavil funkcijski generator, ki ga lahko uporabljamo pri vezjih, kjer potrebujemo clock (CLK) signal (npr. pri sekvenčnih vezjih).

Težave so se pojavile pri komunikaciji med računalnikom in mikrokrmilnikom, saj mi ni uspelo prenesti več kot 254 bajtov podatkov. S pomočjo funkcije usbFunctionWrite sem uspešno poslal celotno število oz. frekvenco.

6. VIRI IN LITERATURA

- Datasheeti
  Dostopno na spletnem naslovu:
  http://www.datasheetcatalog.com/

- Spletna stran programske implementacije AVR-USB:
  http://www.obdev.at/products/avrusb/index.html

- Spletna stran programske knjižnice libusb:
  http://libusb.wiki.sourceforge.net/

- Spletna stran programske knjižnice LibUsb-Win32:
  http://libusb-win32.sourceforge.net/

7. ZAHVALA

Za pomoč pri izdelovanju oziroma svetovanju se zahvaljujem svojemu mentorju prof. Marku Vrečku.

8. KAZALO SLIK

Slika 1: Opis kontaktov na mikrokrmilniku (Vir: ATtiny45 dokumentacija)

Slika 2: Primerjava različnih skupin AVR mikrokontrolerjev

Slika 3: Povezava Atmel mikrokontrolerja na USB priključek pri uporabi AVR-USB.

Slika 4: Prikaz uporabe programa na operacijskemu sistemu Mac

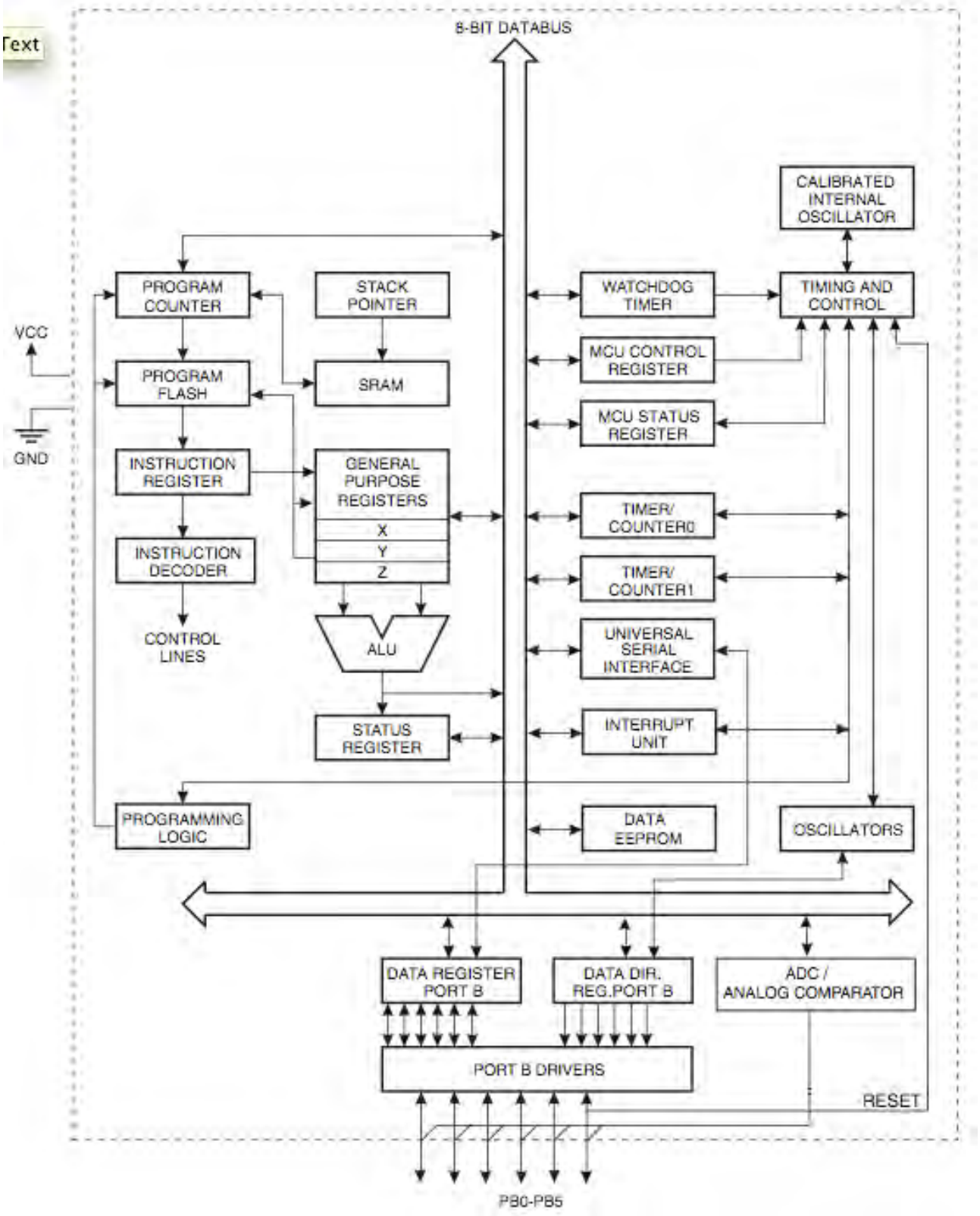
Slika 5: Vezje funkcijskega generatorja

Slika 6: Blokovni diagram mikrokrmilnika ATtiny45

9. KAZALO TABEL

Tabela 1: Tabela z podatki o nastavitvah hitrosti časovnika

10. PRILOGA



Slika 6: Blokovni diagram mikrokrmilnika ATtiny45