

RĪGAS TEHNISKĀ UNIVERSITĀTE
DATORZINĀTNES UN INFORMĀCIJAS TEHNOLOĢIJAS
FAKULTĀTE

DATORVADĪBAS, AUTOMĀTIKAS UN DATORTEHNIKAS INSTITŪTS

Datoru tīklu un sistēmas tehnoloģijas katedra

Mikroprocesoru tehnika

1. laboratorijas darbs

Izpildīja:

Grupa:

Apl. numurs:

III RDB F02

101RDB121

2012./2013. māc. gads

Saturs

Uzdevums.....	5
Programmas pirmteksts.....	6
Programmas pirmteksta algoritma blokhēma.....	10
ATmega128 un AT90CAN32/64/128 mikrokontrolleru salīdzinājums.....	13
Secinājumi.....	14

Uzdevums

1. Iepazīties ar ATmega128 un CharonII;
2. Nokompilēt piedāvātu izejas kodu un ierakstīt *.hex* programfailu mikrokontrollera atmiņā. Pārbaudīt testa programmas korektu darbību;
3. Pārrakstīt programmu tā, lai visas gaismas diodes ieslēgtos pēc kārtas, izveidot „skrejošas gaismas” efektu;

Programmas pirmteksts

```

/*****
#define F_CPU 14745600UL          //Mikrokontrollera takts frekvences definēšana

/***** Standarta C un speciało AVR bibliotēku iekļaušana*****/
#include <avr/io.h>
#include <avr/iom128.h>
#include <avr/interrupt.h>
#include <math.h>
#include <stdlib.h>
#include <stdint.h>
#include <stdio.h>
#include <util/delay.h>
/*****

#define BIT0 0x01 //Nodefinēta 0-taa bita maska
#define BIT1 0x02 //Nodefinēta 1-taa bita maska
#define BIT2 0x04 //Nodefinēta 2-taa bita maska
#define BIT3 0x08 //Nodefinēta 3-taa bita maska
#define BIT4 0x10 //Nodefinēta 4-taa bita maska
#define BIT5 0x20 //Nodefinēta 5-taa bita maska
#define BIT6 0x40 //Nodefinēta 6-taa bita maska
#define BIT7 0x80 //Nodefinēta 7-taa bita maska

/***** Portu inicializācijas funkcija *****/
void port_init(void)
{
    DDRA = 0x00; //visas porta A līnijas uz IEvadi
    DDRB = 0x00; //visas porta B līnijas uz IEvadi
    DDRC = 0x00; //visas porta C līnijas uz IEvadi
    DDRD = 0xFF; //visas porta D līnijas uz IZvadi
    DDRE = 0x00; //visas porta E līnijas uz IEvadi
    DDRF = 0x00; //visas porta F līnijas uz IEvadi
    DDRG = 0x00; //visas porta G līnijas uz IEvadi

    PORTA = 0x00; //porta A atsienošie rezistori pret +Vcc NEtiek izmantoti
    PORTB = 0x00; //porta B atsienošie rezistori pret +Vcc NEtiek izmantoti
    PORTC = 0x00; //porta C atsienošie rezistori pret +Vcc NEtiek izmantoti
    PORTD = 0x00; //porta D izejas līniju līmeni uz 0
    PORTE = 0x00; //porta E atsienošie rezistori pret +Vcc NEtiek izmantoti
    PORTF = 0x00; //porta F atsienošie rezistori pret +Vcc NEtiek izmantoti
    PORTG = 0x00; //porta G atsienošie rezistori pret +Vcc NEtiek izmantoti
}
/*****

/***** Kontrollera inicializācija*****/
void init_devices(void)
{
    cli();                //Aizliedz visus pārtraukumus
    XDIV = 0x00;          //Takts impulsu dalītājs NEtiek izmantots
    XMCRA = 0x00;         //Ārējo atminu NEizmanto

```

```

MCUCR = 0x00;          //NETiek izmantoti nekādi enerģiju taupoši stāvokli
port_init();           //Izsauc funkciju, kas inicializē portus
}
/*****/

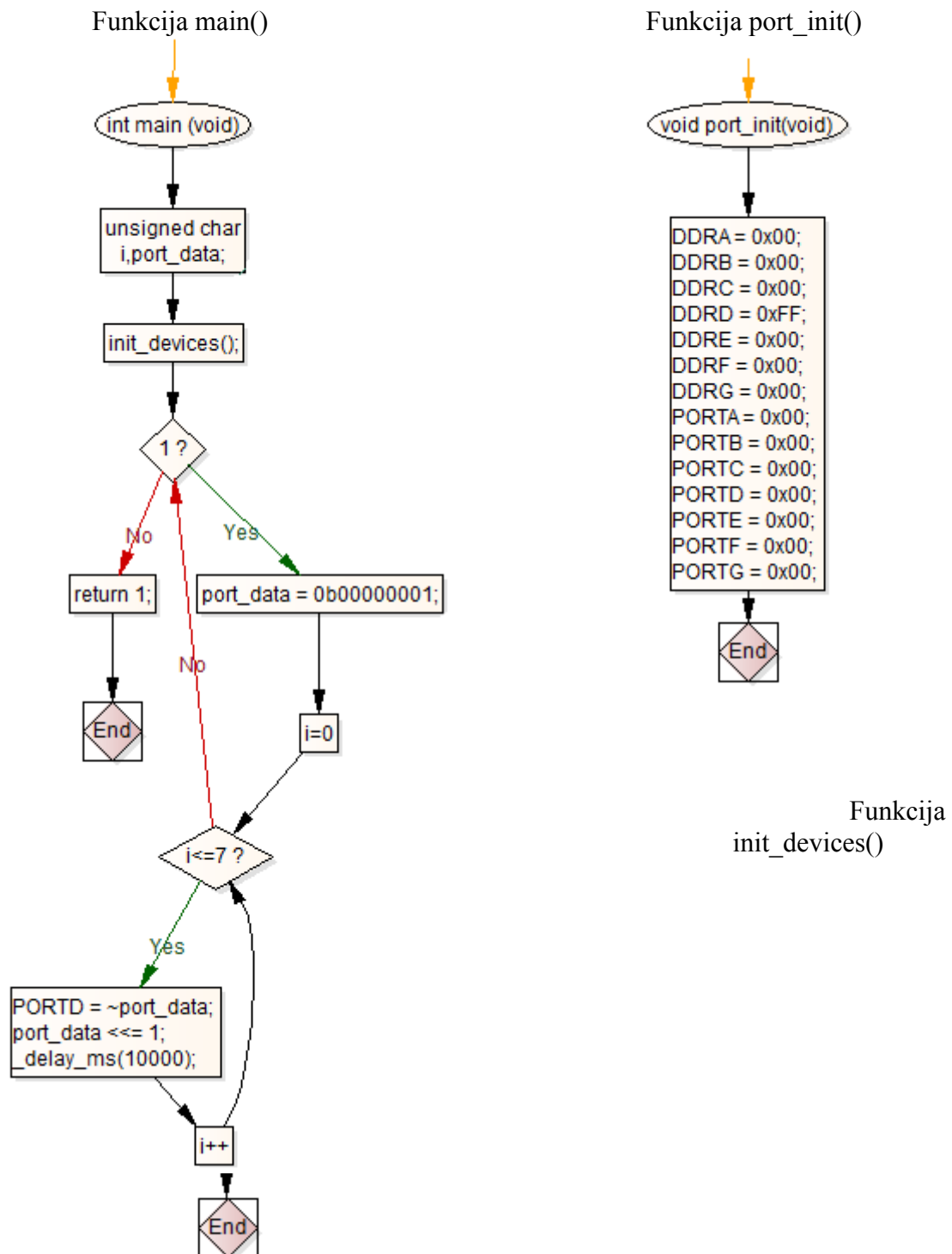
/*****/ Main funkcija *****/
int main (void)
{
    unsigned char i,port_data;    //Mainīgo definēšana

    init_devices();               //Mikrokontrollera inicializēšana

    while(1)                      //Mūžīgais cikls, lai programma nekad nebeigtos
    {
        port_data = 0b00000001;   //Bināras vērtības piešķiršana
        for(i=0;i<=7;i++)         //Cikls, kas izpildās 8 reizes, lai ieslēgtu visas 8 diodes pēc kārtas
        {
            PORTD = ~port_data;    //Izvadīt uz Porta D līnijām apgrieztu port_data mainīga vērtību
            port_data <<= 1;       //Pārbīdīt bināro vērtību uz vienu bitu pa kreisi
            _delay_ms(10000);      //Aizture pirms nākamās gaismas diodes iedegšanas
        }
    }
    return 1;                    //main funkcija atgriež vērtību 1
}
/*****/

```

Programmas pirmteksta algoritma blokshēma



ATmega128 un AT90CAN32/64/128 mikrokontrolleru salīdzinājums

ATmega128 specifikācijas

Augstas veiktspējas, mazjaudas Atmel AVR 8-bitu mikrokontrolleris

• Uzlabota RISC arhitektūra

- 133 jaudīgas instrukcijas – lielākā daļa izpildās vienā takts frekvences ciklā
- 32 x 8 vispārējās lietošanas reģistri + perifērijas kontroles reģistri
- Pilnībā statiskā darbība
- Līdz 16MIPS caurlaidspēja pie 16MHz
- Iebūvēts 2-ciklu reizinātājs

• Augstas izturības energoneatkarīgi atmiņas segmenti

- 128Kbaitu sistēmā iebūvētas programmējamās Flash atmiņas
- 4Kbaitu EEPROM
- 4Kbaitu iekšējā SRAM
- Rakstīšanas / dzēšanas cikls: 10000 Flash/100, 000 EEPROM
- Datu saglabāšana: 20 gadi 85°C/100 gadi pie 25°C
- Izvēles ielādes koda sadaļa ar neatkarīgiem bloķēšanas bitiem
- Sistēmas programmēšana ar čipā iebūvēto ielādes programmu
- Patiesa lasa-kamēr-raksta operācija
- Līdz 64Kbaitiem izvēles ārējās atmiņas vieta
- Programmēšana bloķēšana programmu drošības
- SPI interfeiss priekš sistēmas programmēšanas

• QTouch® bibliotēkas atbalsts

- Kapacitatīvās pieskares pogas, slīdņi un diski
- QTouch un QMatrix iegūšana
- Līdz 64 izjūtas kanāli

• JTAG (IEEE STD 1149,1 RoHS.)

Interfeiss

- Robežu skenēšanas iespējas saskaņā ar JTAG standartu
- Plašs čipa iebūvēts atklūdošanas atbalsts

AT90CAN32/64/128 specifikācijas

• Augstas veiktspējas, mazjaudas AVR 8-bitu mikrokontrolleris

• Uzlabota RISC arhitektūra

- 133 jaudīgas instrukcijas – lielākā daļa izpildās vienā takts frekvences ciklā
- 32 x 8 vispārējās lietošanas reģistri + perifērijas kontroles reģistri
- Pilnībā statiskā darbība
- Līdz 16MIPS caurlaidspēja pie 16MHz
- Iebūvēts 2-ciklu reizinātājs

• energoneatk programmu un datu atmiņa

- 32K/64K/128K baitu iebūvētās pārprogrammējamās Flash atmiņas
- Izturība: 10,000 rakstīt / dzēst ciklu
- Izvēles Boot kodeksa sadaļa ar Neatkarīgo Lock Bits

• uzstādāmais ielādes izmērs: 1KBaiti, 2KBaiti, 4KBaiti vai 8KBaiti

- Sistēmas programmēšana ar iebūvēto ielādes programmu (CAN, UART, ...)
- Patiesa lasa-kamēr-raksta operācija
- 1K/2K/4KBaiti EEPROM (Izturība: 100000 rakstīt / dzēst cikliem)
- 2K/4K/4KBaiti iekšējā SRAM
- Līdz 64Kbaitiem izvēles ārējās atmiņas vieta

- Programmēšana bloķēšana programmu drošības

• JTAG (IEEE STD 1149,1 RoHS.)

Interfeiss

- Robežu skenēšanas iespējas saskaņā ar JTAG standartu
- Flash, EEPROM, drošinātāju un bloķēšanas bitu programmēšana, caur JTAG interfeisu
- Plašs čipa iebūvēts atklūdošanas atbalsts

• CAN kontrolieris 2.0A & 2.0b - ISO

16.845 sertificēts

- 15 pilna paziņojumu objekti ar atsevišķām identifikatoriem un maskām
- Nosūtīšanas, saņemšanas, automātiskās atbildes un kadru bufera režīmi

- Flash, EEPROM, drošinātāju un bloķēšanas bitu programmēšana, caur JTAG interfeisu
- Perifērijas funkcijas
- Divi 8-bitu taimeris / skaitītājs ar atsevišķu mērogotāju un salīdzināšanas režīmu
- Divi paplašināti 16-bit taimeri / skaitītāji ar atsevišķu mērogotāju, salīdzināšanas un satveršanas režīmiem
- Reālā laika skaitītājs ar atsevišķu oscilatoru
- Divi 8-bit PWM kanāli
- 6 PWM kanāli ar programmējamu izšķirtspēju 2-16 biti
- Rezultāta salīdzināšanas modulators
- 8-kanālu, 10-bitu ADC
 - 8 viena gala kanāli
 - 7 diferenciāli kanāli
 - 2 diferenciāli kanāli ar programmējamu pieaugumu 1X, 10x vai 200x
- Uz baitiem orientēts divvadu sērijas interfeiss
- Divi programmējami sērijas USART
- Master / slave SPI sērijas interfeiss
- Programmējams Watchdog taimeris ar čipā iebūvēto oscilatoru
- čipā iebūvēts analogais komparators
- Īpašas mikrokontrollera funkcijas
- Ieslēgšanas Reset un programmējamā Brown-out atrašana
- Iekšējais kalibrēts RC oscilators
- Ārējie un iekšējie pārtraukumu avoti
- Seši miega režīmi: dīkstāve, ADC trokšņa samazināšana, enerģijas saglabāšanas, izslēgšanās, uzgaidīšanas un pagarināts gaidīšanas
- Programmatūras uzstādāma takts frekvence
- ATmega103 saderības režīms uzstādams ar drošinātāju
- Globālā Pull-up atslēgšana
- I / O un paketes
- 53 programmējamās I / O līnijas
- 64-svina TQFP un 64-pad QFN / MLF
- Pamatdarbības spriegumi
- 2.7 - 5.5V ATmega128L
- 4.5 - 5.5V ATMEGA128
- Ātruma pakāpes
- 0 - 8MHz ATmega128L
- 0 - 16MHz ATMEGA128
- 1Mbits / s maksimālais pārsūtīšanas ātrums pie 8 MHz
- Laika štancēšana, TTC un klausīšanās režīmi (Spying vai Autobaud)
- Perifērijas funkcijas
- Programmējams Watchdog taimeris ar čipā iebūvēto oscilatoru
- 8 bitu sinhronais taimeris/skaitītājs-0
- 10-bitu mērogotājs
- Ārējais notikumu skaitītājs
- Izejas pielīdzināšanai vai 8 bitu PWM izeja
- 8 bitu asinhronais taimeris / skaitītājs-2
- 10-bitu mērogotājs
- Ārējais notikumu skaitītājs
- Izejas salīdzināt vai 8-Bitu PWM izeja
- 32Khz oscilators ar RTC operāciju
- Duālais 16-bitu sinhronais taimeris / skaitītājs -1 & 3
- 10-bitu mērogotājs
- Izejas satveršana ar trokšņu novēršana
- Ārējais notikumu skaitītājs
- 3-Izejas Salīdzināt vai 16 bitu PWM izeja
- Izejas salīdzināšanas modulācija
- 8-kanālu, 10-bitu SAR ADC
- 8 viena gala kanāli
- 7 diferenciāli kanāli
- 2 diferenciāli kanāli ar programmējamu pieaugumu 1X, 10x vai 200x
- Iebūvēts analogais komparators
- Uz baitiem orientēts divvadu sērijas interfeiss
- Divi programmējami sērijas USART
- Master / Slave SPI sērijas interfeiss
- Programmēšana Flash (Aparatūras ISP)
- Īpašas mikrokontrollera funkcijas
- Ieslēgšanas Reset un programmējamā Brown-out atrašana
- Iekšējais kalibrēts RC oscilators
- 8 ārējie pārtraukumu avoti
- 5 miega režīmi: dīkstāve, ADC trokšņa samazināšana, enerģijas saglabāšanas, izslēgšanās un uzgaidīšanas
- Programmatūras uzstādāma takts frekvence
- Globālā Pull-up atslēgšana
- I / O un paketes
- 53 Programmējamās I / O līnijas
- 64-svina TQFP un 64 novadījumu QFN
- Pamatdarbības Spriegums: 2.7 - 5.5V
- Maksimālais frekvence: 8 MHz pie 2.7V, 16 MHz pie 4.5V

Apskatot mikrokontrolleru specifikācijas, var secināt, ka abi mikrokontrolleri ir ļoti līdzīgi gan uzbūves, gan funkciju ziņā, īpaši ja salīdzināt ATmega128 ar AT90CAN128. Abi ir 8-bitu mikrokontrolleri, strādā ar līdzīgu takts frekvenci un spriegumu, ka arī ir aprīkoti ar vienāda izmēra Flash, EEPROM un SRAM atmiņu. Arhitektūra (RISC), instrukcijas un reģistru uzbūve arī sakrīt.

- Tomēr ATmega128 ar AT90CAN128 mikrokontrolleriem ir arī zināmas atšķirības: ATmega128 ir ārējie un iekšējie pārtraukumu avoti, bet AT90CAN128 ir minēti tikai ārējie;
- ATmega128 mikrokontrollerim ir 6 miega režīmi, bet AT90CAN128 – 5 (nav pagarinātās jeb ilgas gaidīšanas režīma);
- ATmega128 ir QTouch bibliotēkas atbalsts;
- AT90CAN128 ir iebūvēts CAN kontrolieris 2.0A & 2.0b;
- AT90CAN128 aprakstā nav minēts SPI interfeiss sistēmas programmēšanai;
- AT90CAN128 (pēc apraksta) nav reālā laika skaitītāja;
- ATmega128 mikrokontrollerim ir kapacitatīvās pieskares (skārienjūtīgas) pogas, slīdņi un diskus

Ir arī zināmas atšķirības izejās, kanālos un režīmu veidos, tāpēc izvēloties mikrokontrolleri kāda darba izpildīšanai ir rūpīgi jāizlasa apraksts un jāizdara

Secinājumi

Pirmā laboratorijas darba gaitā es iepazinos ar AVR Studio 4 vidi, kā arī ar mikrokontrollera ATmega128 un CharonII izstrādes plates darbību, izpildot un papildinot kodu C programmēšanas valodā.

Lai izpildītu šo darbu bija nepieciešams atcerēties darbības ar bitiem, jo lai gaismas diodes ieslēgtos pēc kārtas, programmas kodā bija jālieto binārā nobīde. Iepriekšējosursos iegūtās zināšanas par dažādiem skaitļu pieraksta veidiem (binārais, heksadecimālais) palīdzēja izprast programmas kodu. Laboratorijas darbs nebija grūts un neaizņēma daudz laika. Darbs tiktpaveikts vēl ātrāk, ja pirms tā veikšanas es atkārtotu darbības ar bitiem un zinātu, kā pareizi realizēt aizturi (delay).

Uzskatu, ka laboratorijas darbs ir veiksmīgi izpildīts, jo papildinātais kods strādā bez kļūdām un pēc tā realizēšanas, gaismas diodes uz laboratorijas darbam piedāvātās shēmas ieslēdzās pēc kārtas, ka tika pieprasīts uzdevumā.