Rīgas Tehniska Universitāte

Datorvadības, automātikas un datortehnikas institūts

Datoru tīklu un sistēmas tehnoloģijas katedra

Mikroprocesoru tehnika
Laboratorijas darbs Nr.1

Asist. R. Taranovs
Profesors V.Zagurskis
Students
3.kurss 1.grupa

Uzdevums

1. Iepazīties ar ATmega128 un Charon II;
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;

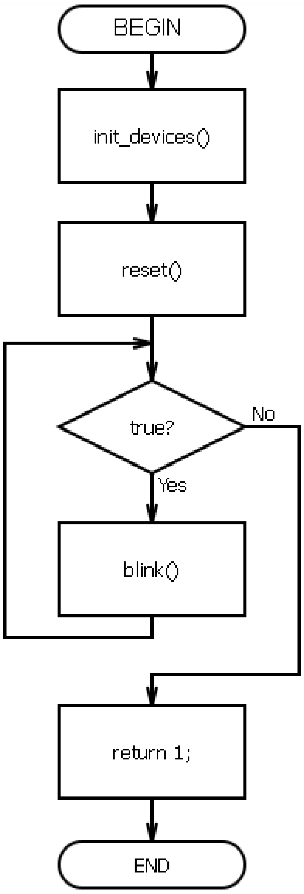
Programmas kods

```
/*****************************************************************/
#define F_CPU 14745600UL  //Mikrokontrollera takts frekvences defin??ana
/********* Standarta C un specialo AVR bibliot?ku iek?au?ana*********/
#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>
/******************* Portu inicializ?cijas funkcija *****************/
void port_init(void){
 DDRA  = 0x00; //visas porta A l?nijas uz IEvadi
 DDRB  = 0x00; //visas porta B l?nijas uz IEvadi
 DDRC  = 0x00; //visas porta C l?nijas uz IEvadi
 DDRD  = 0xFF; //visas porta D l?nijas uz IZvadi
 DDRE  = 0x00; //visas porta E l?nijas uz IEvadi
 DDRF  = 0x00; //visas porta F l?nijas uz IEvadi
 DDRG  = 0x00; //visas porta G l?nijas uz IEvadi

 PORTA  = 0x00; //porta A atsieno?ie rezistori pret +Vcc NEtiek izmantoti
 PORTB  = 0x00; //porta B atsieno?ie rezistori pret +Vcc NEtiek izmantoti
 PORTC  = 0x00; //porta C atsieno?ie rezistori pret +Vcc NEtiek izmantoti
 //PORTD  = 0x00; //porta D izejas l?niju l?me?i uz 0
 PORTE  = 0x00; //porta E atsieno?ie rezistori pret +Vcc NEtiek izmantoti
 PORTF  = 0x00; //porta F atsieno?ie rezistori pret +Vcc NEtiek izmantoti
 PORTG  = 0x00; //porta G atsieno?ie rezistori pret +Vcc NEtiek izmantoti
}
/*****************************************************************/
/******************* Kontrollera inicializ?cija**********************/
void init_devices(void){
cli();                  //Aizliedz visus p?rtraukumus
XDIV = 0x00;            //Takts impulsu dal?t?js NEtiek izmantots
XMCRA = 0x00;           //?r?jo atmi?u NEizmanto
MCUCR = 0x00;           //NEtiek izmantoti nek?di ener?iju taupo?i st?vokli
port_init();            //Izsauc funkciju, kas inicializ? portus
}
/*****************************************************************/
void reset(){
        PORTD=~(0x00);
        _delay_loop_2(65000);
}

void blink(){
        unsigned char i;
        //blink this way ---->>>
        for(i = 1; i < 128; i = i*2){
                PORTD=~i;
```

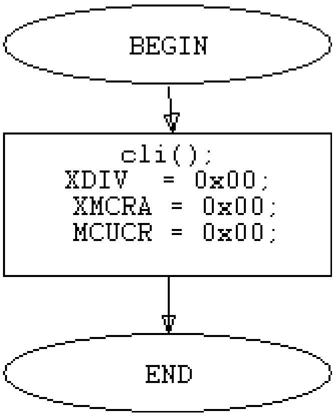
```
		for(int ii=0;ii<10;ii++)_delay_loop_2(65000);
		}

	//another way
	for(i = 128; i > 1; i -= i/2){
		PORTD=~i;
		for(int ii=0;ii<10;ii++)_delay_loop_2(65000);
		}
}
/***************************** Main funkcija **************************/
int main (void){
init_devices(); //Inicializ? mikrokontrolleri
reset();
while(1){
	blink();
}
return 1;
 }
/********************************************************************/
```

Blokshēma

Main() bloks

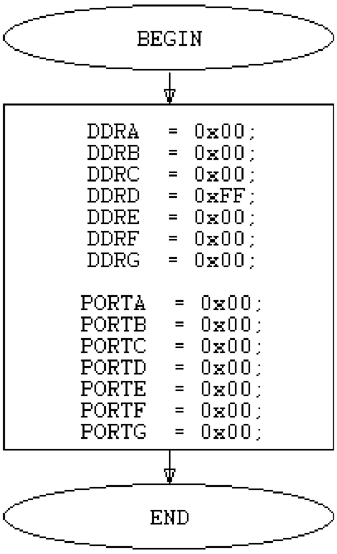


Init_devices()



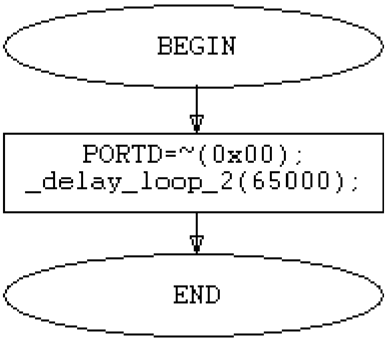
```
     cli();
XDIV   = 0x00;
 XMCRA = 0x00;
 MCUCR = 0x00;
```

Port_init()



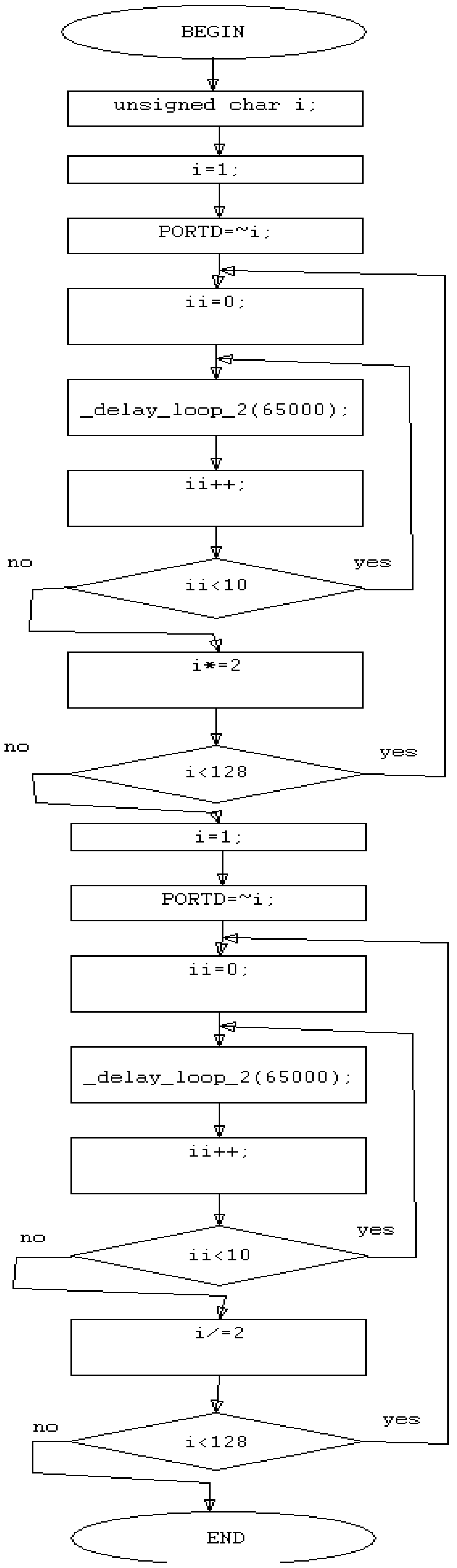
```
DDRA  = 0x00;
DDRB  = 0x00;
DDRC  = 0x00;
DDRD  = 0xFF;
DDRE  = 0x00;
DDRF  = 0x00;
DDRG  = 0x00;

PORTA  = 0x00;
PORTB  = 0x00;
PORTC  = 0x00;
PORTD  = 0x00;
PORTE  = 0x00;
PORTF  = 0x00;
PORTG  = 0x00;
```

Reset()



```
PORTD=~(0x00);
_delay_loop_2(65000);
```

Blink()


BEGIN
unsigned char i;
i=1;
PORTD=~i;
ii=0;
_delay_loop_2(65000);
ii++;
no
yes
ii<10
i*=2
no
yes
i<128
i=1;
PORTD=~i;
ii=0;
_delay_loop_2(65000);
ii++;
no
yes
ii<10
i/=2
no
yes
i<128
END

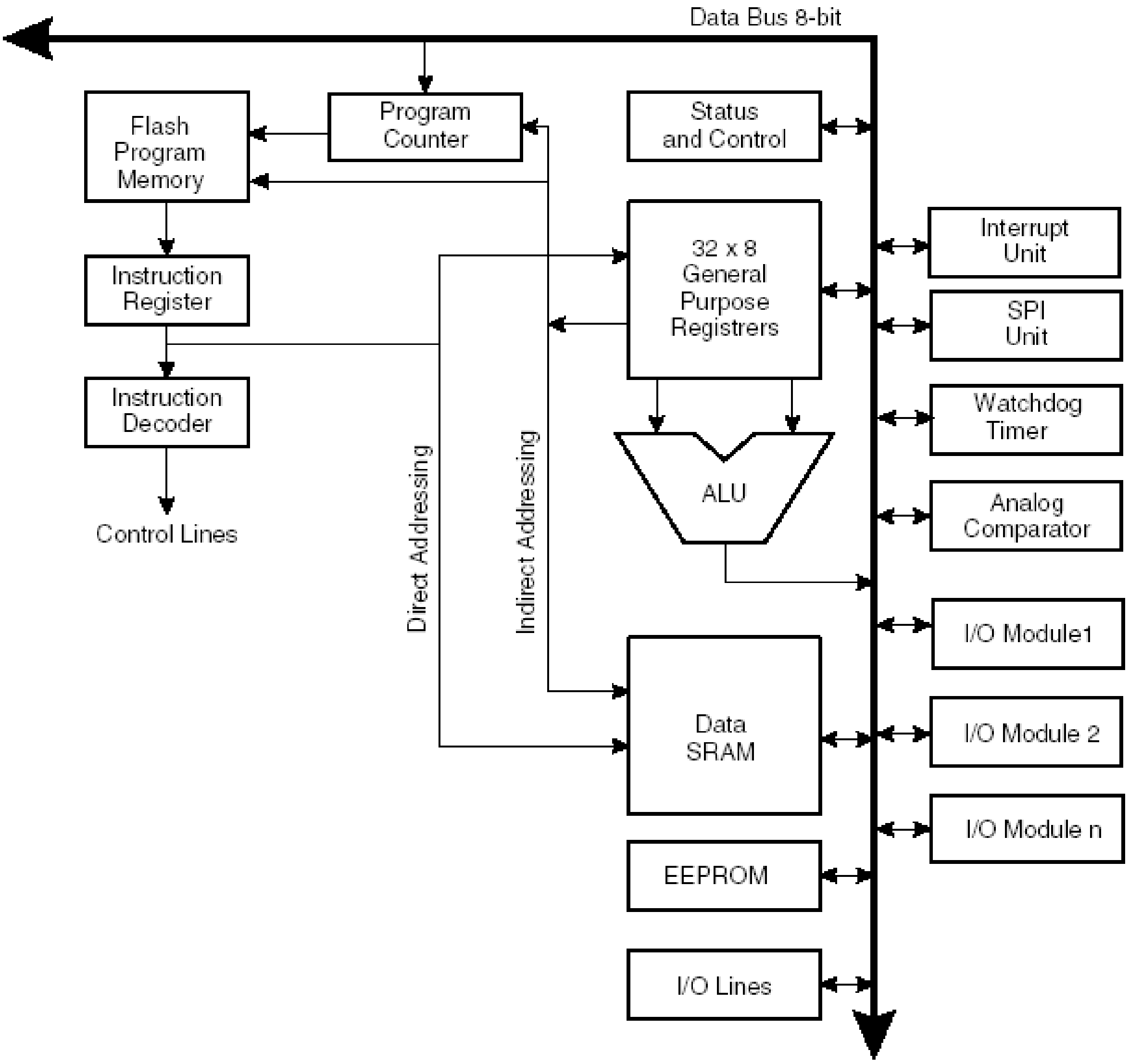
ATmega128 un AT90PWM2/3/2B/3B

ATmega128(L) Features	AT90PWM2/3/2B/3B Features
High Performance, Low Power Atmel® AVR® 8-bit Microcontroller Advanced RISC Architecture – 133 Powerful Instructions - Most Single Clock Cycle Execution – 32 x 8 General Purpose Working Registers+ Peripheral Control Registers – Fully Static Operation – On-chip 2-cycle Multiplier – Up to 16 MIPS Throughput at 16 MHz	High-performance, Low-power AVR® 8-bit Microcontroller Advanced RISC Architecture – 129 Powerful Instructions – Most Single Clock Cycle Execution(nav JMP, CALL,ELPM...) – 32 x 8 General Purpose Working Registers – Fully Static Operation – On-chip 2-cycle Multiplier – Up to 1 MIPS throughput per MHz
High Endurance Non-volatile Memory segments – 128K Bytes of In-System Self-programmable Flash program mem – 4K Bytes EEPROM – 4K Bytes Internal SRAM – Write/Erase cycles: 10,000 Flash/100,000 EEPROM – Data retention: 20 years at 85°C/100 years at 25°C(1) – Optional Boot Code Section with Independent Lock Bits – In-System Programming by On-chip Boot Program – True Read-While-Write Operation – Up to 64K Bytes Optional External Memory Space – Programming Lock for Software Security – SPI Interface for In-System Programming	Data and Non-Volatile Program Memory – 8K Bytes Flash of In-System Programmable Program Memory – 512 Bytes of In-System Programmable EEPROM – 512 Bytes Internal SRAM – Write/Erase cycles: 10,000 Flash/100,000 EEPROM – Optional Boot Code Section with Independent Lock Bits – In-System Programming by On-chip Boot Program – True Read-While-Write Operation – Programming Lock for Flash Program and EEPROM Data Security
JTAG (IEEE std. 1149.1 Compliant) Interface – Boundary-scan Capabilities According to the JTAG Standard – Extensive On-chip Debug Support – Programming of Flash, EEPROM, Fuses and Lock Bits through the JTAG Interface	On Chip Debug Interface (debugWIRE)

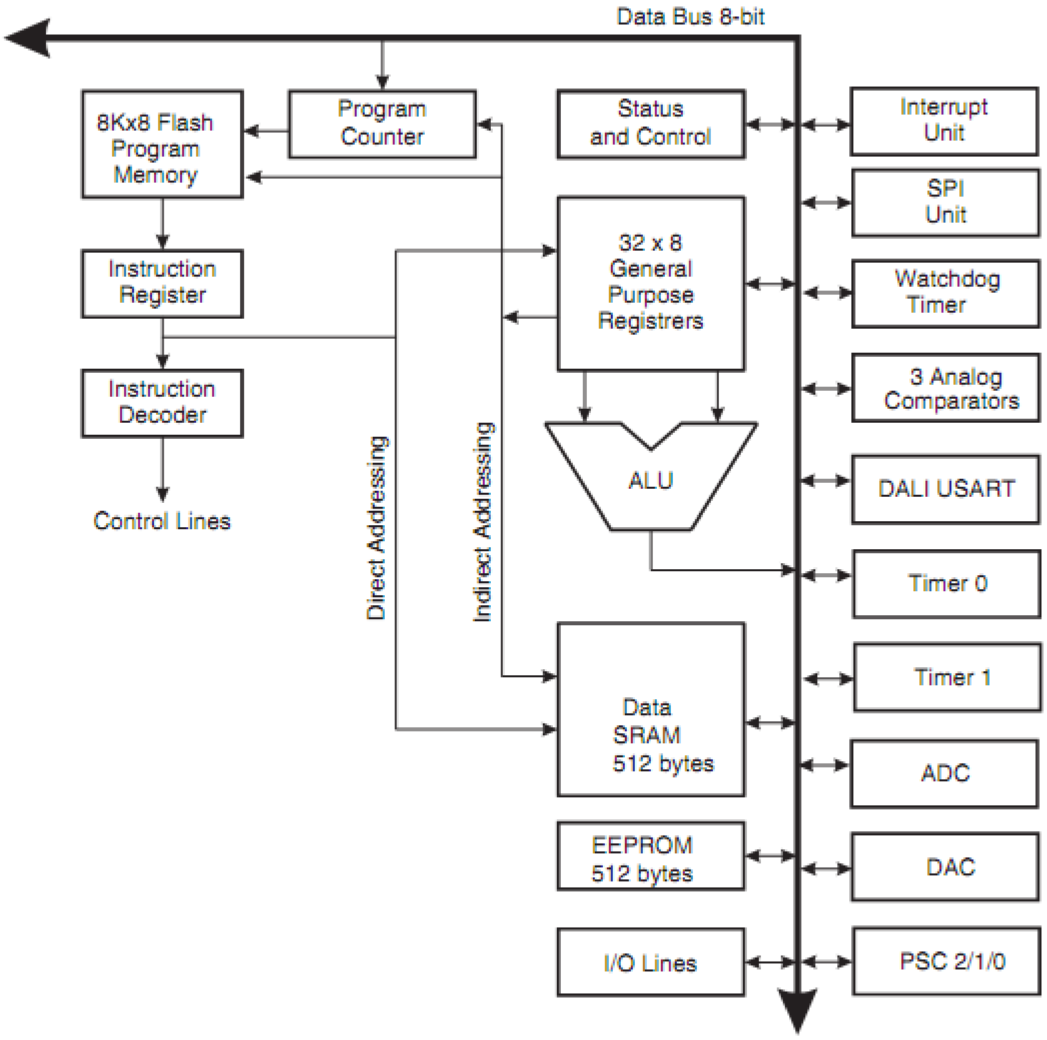
Special Microcontroller Features – Power-on Reset and Programmable Brown-out Detection – Internal Calibrated RC Oscillator – External and Internal Interrupt Sources – Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby – Software Selectable Clock Frequency – ATmega103 Compatibility Mode Selected by a Fuse – Global Pull-up Disable	Special Microcontroller Features – Power On Reset and Programmable Brown Out Detection – Internal Calibrated RC Oscillator (8 MHz) – Low Power Idle, Noise Reduction, and Power Down Modes – Flag Array in Bit-programmable I/O Space (4 bytes) – In-System Programmable via SPI Port – On-chip PLL for fast PWM (32 MHz, 64 MHz) and CPU (16 MHz)
Operating Voltages – 2.7 - 5.5V ATmega128L – 4.5 - 5.5V ATmega128 Speed Grades – 0 - 8 MHz ATmega128L – 0 - 16 MHz ATmega128	Operating Voltage: 2.7V - 5.5V Extended Operating Temperature: 40°C to +105°C
Peripheral Features Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes Two Expanded 16-bit Timer/Counters with Separate Prescaler, Compare Mode and Capture Mode Programmable Watchdog Timer with On-chip Oscillator 8-channel, 10-bit ADC – 8 Single-ended Channels – 7 Differential Channels – 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x Real Time Counter with Separate Oscillator Two 8-bit PWM Channels 6 PWM Channels with Programmable Resolution from 2 to 16 Bits Output Compare Modulator Byte-oriented Two-wire Serial Interface Dual Programmable Serial USARTs Master/Slave SPI Serial Interface On-chip Analog Comparator	Peripheral Features One 8-bit General purpose Timer/Counter with Separate Prescaler and Capture Mode One 16-bit General purpose Timer/Counter with Separate Prescaler, Compare Mode and Capture Mode Programmable Watchdog Timer with Separate On-Chip Oscillator 10-bit ADC – Up To 11 Single Ended Channels and 2 Fully Differential ADC Channel Pairs – Programmable Gain (5x, 10x, 20x, 40x on Differential Channels) – Internal Reference Voltage Two or three 12-bit High Speed PSC (Power Stage Controllers) with 4-bit Resolution Enhancement – Non Overlapping Inverted PWM Output Pins With Flexible Dead-Time – Variable PWM duty Cycle and Frequency – Synchronous Update of all PWM Registers – Auto Stop Function for Event Driven PFC Implementation – Less than 25 Hz Step Width at 150 kHz

	Output Frequency – PSC2 with four Output Pins and Output Matrix Programmable Serial USART(Universal Syncronous Asyncronous Receiver Transmitter) – Standard UART mode – 16/17 bit Biphase Mode for DALI Communications Master/Slave SPI Serial Interface 10-bit DAC(Digital-to-analog Converter) Two or three Analog Comparator with Resistor-Array to Adjust Comparison Vol tage 4 External Interrupts

53 programmējamās līnijas, kuras sastāv no:
– A porta;
– B porta;
– C porta;
– D porta;
– E porta;
– F porta;
– G porta;
	19 programmējamās līnijas, kuras sastāv no:
	– B port(PB7-PB0)
	– C port(PC7-PC0)
	– D port(PD7-PD0)
	– E port(PE2-PE0)



Attēls 1ATmega128 arhitektūras bloku diagramma



Attēls 2 AT90PWM2/3/2B/3B arhitektūras bloku diagramma

Secinājumi

Izpildot pirmo laboratorijas darbu es tiku iepazīstināts ar mikrokontrollera minimālo koda saturu, to struktūru, ka arī ar pašu izstrādes plati. Lai veiksmīgi izpildītu šo darbu vajadzēja atcerēties darbības ar bitiem, jo programmēšanā mēs tos neaiztikām, ka arī skaitļu pieraksta veidus(bin, hex). Ja viss iepriekš minētais būtu atkārtots pirms tam laboratorijas darbs neaizņemtu daudz laika. Papildus es iepazīstinājos ar ATmega128 mikrokontrolleri un Charon II izstrādes plati.