ЛАБА 3(Столбец, строка, смайлик)

.INCLUDE "m16def.inc"

.LIST

.CSEG

.ORG 0X0000

.equ DELAY = 0xFF

LDI R16,Low(RAMEND)

OUT SPL,R16

LDI R16,High(RAMEND)

OUT SPH,R16

LDI R16,0b11111111

OUT DDRD,R16

OUT DDRC,R16

OUT DDRB,R16

OUT DDRA,R16

MAIN:

SBI PORTD,0

SBI PORTD,1

SBI PORTD,2

SBI PORTD,3

SBI PORTD,4

SBI PORTD,5

SBI PORTD,6

SBI PORTD,7

RCALL WAIT

CBI PORTD,0

CBI PORTD,1

CBI PORTD,2

CBI PORTD,3

CBI PORTD,4

CBI PORTD,5

CBI PORTD,6

CBI PORTD,7

RCALL WAIT

SBI PORTD,0

RCALL WAIT

SBI PORTC,0

RCALL WAIT

SBI PORTB,0

RCALL WAIT

SBI PORTA,0

RCALL WAIT

CBI PORTA,0

RCALL WAIT

CBI PORTB,0

RCALL WAIT

CBI PORTC,0

RCALL WAIT

CBI PORTD,0

RCALL WAIT

SBI PORTA,4

SBI PORTB,5

SBI PORTB,3

SBI PORTC,5

SBI PORTC,3

SBI PORTD,4

RCALL WAIT

RCALL WAIT

CBI PORTA,4

CBI PORTB,5

CBI PORTB,3

CBI PORTC,5

CBI PORTC,3

CBI PORTD,4

RJMP MAIN

WAIT:

LDI R17,DELAY

NUM1:

LDI R18,0x12

NUM2:

LDI R19,0x45

NUM3:

DEC R19

BRNE NUM3

DEC R18

BRNE NUM2

DEC R17

BRNE NUM1

RET

Лаба 4 (Обработка кнопок PORTD и выведения соответствующих цифр)

.INCLUDE "m16def.inc"

.LIST

.CSEG

.ORG 0x0000

.equ DELAY = 0xFF

LDI R16,LOW(RAMEND)

OUT SPL,R16

LDI R16,HIGH(RAMEND)

OUT SPH,R16

LDI R16,0b11111111

OUT DDRA,R16

OUT DDRB,R16

OUT DDRC,R16

MAIN:

CLR R17

LDI R16,0xFF

OUT DDRD,R17

IN R18,PIND

SBRC R18,0

RCALL ZERO

IN R18,PIND

SBRC R18,1

RCALL ONE

IN R18,PIND

SBRC R18,2

RCALL TWO

IN R18,PIND

SBRC R18,3

RCALL THREE

IN R18,PIND

SBRC R18,4

RCALL FOUR

IN R18,PIND

SBRC R18,5

RCALL FIVE

IN R18,PIND

SBRC R18,6

RCALL SIX

IN R18,PIND

SBRC R18,7

RCALL SEVEN

RJMP MAIN

ZERO:

OUT DDRD,R16

SBI PORTA,2

SBI PORTA,3

SBI PORTA,4

SBI PORTA,5

SBI PORTD,2

SBI PORTD,3

SBI PORTD,4

SBI PORTD,5

SBI PORTB,1

SBI PORTB,6

SBI PORTC,1

SBI PORTC,6

RCALL WAIT

CBI PORTA,2

CBI PORTA,3

CBI PORTA,4

CBI PORTA,5

CBI PORTD,2

CBI PORTD,3

CBI PORTD,4

CBI PORTD,5

CBI PORTB,1

CBI PORTB,6

CBI PORTC,1

CBI PORTC,6

RETI

ONE:

OUT DDRD,R16

SBI PORTA,3

SBI PORTA,6

SBI PORTB,2

SBI PORTB,6

SBI PORTC,1

SBI PORTC,2

SBI PORTC,3

SBI PORTC,4

SBI PORTC,5

SBI PORTC,6

SBI PORTD,6

RCALL WAIT

CBI PORTA,3

CBI PORTA,6

CBI PORTB,2

CBI PORTB,6

CBI PORTC,1

CBI PORTC,2

CBI PORTC,3

CBI PORTC,4

CBI PORTC,5

CBI PORTC,6

CBI PORTD,6

RETI

TWO:

OUT DDRD,R16

SBI PORTA,2

SBI PORTA,5

SBI PORTA,6

SBI PORTB,1

SBI PORTB,4

SBI PORTB,6

SBI PORTC,1

SBI PORTC,3

SBI PORTC,6

SBI PORTD,2

SBI PORTD,6

RCALL WAIT

CBI PORTA,2

CBI PORTA,5

CBI PORTA,6

CBI PORTB,1

CBI PORTB,4

CBI PORTB,6

CBI PORTC,1

CBI PORTC,3

CBI PORTC,6

CBI PORTD,2

CBI PORTD,6

RETI

THREE:

OUT DDRD,R16

SBI PORTA,2

SBI PORTA,5

SBI PORTB,1

SBI PORTB,6

SBI PORTC,1

SBI PORTC,3

SBI PORTC,4

SBI PORTC,6

SBI PORTD,2

SBI PORTD,5

RCALL WAIT

CBI PORTA,2

CBI PORTA,5

CBI PORTB,1

CBI PORTB,6

CBI PORTC,1

CBI PORTC,3

CBI PORTC,4

CBI PORTC,6

CBI PORTD,2

CBI PORTD,5

RETI

FOUR:

OUT DDRD,R16

SBI PORTA,1

SBI PORTA,2

SBI PORTA,3

SBI PORTB,3

SBI PORTC,3

SBI PORTD,1

SBI PORTD,2

SBI PORTD,3

SBI PORTD,4

SBI PORTD,5

SBI PORTD,6

RCALL WAIT

CBI PORTA,1

CBI PORTA,2

CBI PORTA,3

CBI PORTB,3

CBI PORTC,3

CBI PORTD,1

CBI PORTD,2

CBI PORTD,3

CBI PORTD,4

CBI PORTD,5

CBI PORTD,6

RETI

FIVE:

OUT DDRD,R16

SBI PORTA,1

SBI PORTA,2

SBI PORTA,3

SBI PORTA,5

SBI PORTB,1

SBI PORTB,3

SBI PORTB,6

SBI PORTC,1

SBI PORTC,3

SBI PORTC,6

SBI PORTD,1

SBI PORTD,4

SBI PORTD,5

RCALL WAIT

CBI PORTA,1

CBI PORTA,2

CBI PORTA,3

CBI PORTA,5

CBI PORTB,1

CBI PORTB,3

CBI PORTB,6

CBI PORTC,1

CBI PORTC,3

CBI PORTC,6

CBI PORTD,1

CBI PORTD,4

CBI PORTD,5

RETI

SIX:

OUT DDRD,R16

SBI PORTA,2

SBI PORTA,3

SBI PORTA,4

SBI PORTA,5

SBI PORTB,1

SBI PORTB,4

SBI PORTB,6

SBI PORTC,1

SBI PORTC,4

SBI PORTC,6

SBI PORTD,2

SBI PORTD,5

RCALL WAIT

CBI PORTA,2

CBI PORTA,3

CBI PORTA,4

CBI PORTA,5

CBI PORTB,1

CBI PORTB,4

CBI PORTB,6

CBI PORTC,1

CBI PORTC,4

CBI PORTC,6

CBI PORTD,2

CBI PORTD,5

RETI

SEVEN:

OUT DDRD,R16

SBI PORTA,1

SBI PORTB,1

SBI PORTB,4

SBI PORTB,5

SBI PORTB,6

SBI PORTC,1

SBI PORTC,3

SBI PORTD,1

SBI PORTD,2

RCALL WAIT

CBI PORTA,1

CBI PORTB,1

CBI PORTB,4

CBI PORTB,5

CBI PORTB,6

CBI PORTC,1

CBI PORTC,3

CBI PORTD,1

CBI PORTD,2

RETI

WAIT:

LDI R17,DELAY

NUM1:

LDI R18,0x12

NUM2:

LDI R19,0x45

NUM3:

DEC R19

BRNE NUM3

DEC R18

BRNE NUM2

DEC R17

BRNE NUM1

RET

Лаба 5(Обработка ADC и побитовый вывод числа на диоды)

.include "m16def.inc"

.dseg

.cseg

.org 0

RESET:

LDI R16,HIGH(RAMEND)

OUT SPH, R16

LDI R16,LOW(RAMEND)

OUT SPL, R16

LDI R16, 0XFF

OUT DDRD,R16

OUT DDRC,R16

LDI R16,0b01000000

OUT ADMUX, R16

LDI R16,0b11011111

OUT ADCSRA, R16

IN R17,ADCL

IN R18,ADCH

OUT PORTD, R17

OUT PORTC, R18

RJMP RESET

NUM1:

LDI R18,0XFF

OUT PORTD,R18

RCALL WAIT

LDI R18,0X00

OUT PORTD,R18

WAIT:

LDI R19,0XFF

LOOP1:

LDI R20,0X15

LOOP2:

LDI R21,0X10

LOOP3:

DEC R21

BREQ LOOP3

DEC R20

BREQ LOOP2

DEC R19

BREQ LOOP1

RET

Лаба 6(перевод задач на С)

6.1

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRD = 0xFF;

DDRC = 0xFF;

DDRB = 0xFF;

DDRA = 0xFF;

while(1)

{

PORTA = 0xff;

\_delay\_ms(5000);

PORTA = 0x00;

PORTA = 0b00000001;

PORTB = 0b00000001;

PORTC = 0b00000001;

PORTD = 0b00000001;

\_delay\_ms(5000);

PORTA = 0x00;

PORTB = 0x00;

PORTC = 0x00;

PORTD = 0x00;

\_delay\_ms(10000);

PORTA = 0b00010000;

PORTB = 0b00101100;

PORTC = 0b00101100;

PORTD = 0b00010000;

\_delay\_ms(10000);

PORTA = 0x00;

PORTB = 0x00;

PORTC = 0x00;

PORTD = 0x00;

}

}

6.2

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRD = 0x00;

DDRC = 0xFF;

DDRB = 0xFF;

DDRA = 0xFF;

while(1)

{

if(PIND&(1 << PD0))

{

DDRD = 0xFF;

PORTA = 0b00111100;

PORTB = 0b01000010;

PORTC = 0b01000010;

PORTD = 0b00111100;

\_delay\_ms(5000);

PORTA = 0x00;

PORTB = 0x00;

PORTC = 0x00;

PORTD = 0x00;

DDRD = 0x00;

}

if(PIND&(1 << PD1))

{

DDRD = 0xFF;

PORTA = 0b01001000;

PORTB = 0b01000100;

PORTC = 0b01111110;

PORTD = 0b01000000;

\_delay\_ms(5000);

PORTA = 0x00;

PORTB = 0x00;

PORTC = 0x00;

PORTD = 0x00;

DDRD = 0x00;

}

if(PIND&(1 << PD2))

{

DDRD = 0xFF;

PORTA = 0b01000100;

PORTB = 0b01100010;

PORTC = 0b01010010;

PORTD = 0b01001100;

\_delay\_ms(5000);

PORTA = 0x00;

PORTB = 0x00;

PORTC = 0x00;

PORTD = 0x00;

DDRD = 0x00;

}

if(PIND&(1 << PD3))

{

DDRD = 0xFF;

PORTA = 0b00100100;

PORTB = 0b01000010;

PORTC = 0b01011010;

PORTD = 0b00100100;

\_delay\_ms(5000);

PORTA = 0x00;

PORTB = 0x00;

PORTC = 0x00;

PORTD = 0x00;

DDRD = 0x00;

}

if(PIND&(1 << PD4))

{

DDRD = 0xFF;

PORTA = 0b00001110;

PORTB = 0b00001000;

PORTC = 0b00001000;

PORTD = 0b01111110;

\_delay\_ms(5000);

PORTA = 0x00;

PORTB = 0x00;

PORTC = 0x00;

PORTD = 0x00;

DDRD = 0x00;

}

if(PIND&(1 << PD5))

{

DDRD = 0xFF;

PORTA = 0b00101110;

PORTB = 0b01001010;

PORTC = 0b01001010;

PORTD = 0b00110010;

\_delay\_ms(5000);

PORTA = 0x00;

PORTB = 0x00;

PORTC = 0x00;

PORTD = 0x00;

DDRD = 0x00;

}

if(PIND&(1 << PD6))

{

DDRD = 0xFF;

PORTA = 0b00111100;

PORTB = 0b01010010;

PORTC = 0b01010010;

PORTD = 0b00100100;

\_delay\_ms(5000);

PORTA = 0x00;

PORTB = 0x00;

PORTC = 0x00;

PORTD = 0x00;

DDRD = 0x00;

}

if(PIND&(1 << PD7))

{

DDRD = 0xFF;

PORTA = 0b00000010;

PORTB = 0b01100010;

PORTC = 0b00010010;

PORTD = 0b00001110;

\_delay\_ms(5000);

PORTA = 0x00;

PORTB = 0x00;

PORTC = 0x00;

PORTD = 0x00;

DDRD = 0x00;

}

}

}

6.3

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRD = 0xFF;

DDRC = 0xFF;

while(1)

{

ADMUX = 0b01000000;

ADCSRA = 0b11011111;

PORTD = ADCL;

PORTC = ADCH;

\_delay\_ms(1000);

}

}

Лаба 7(Смайлик на прерываниях)

.include "m16def.inc"

.DSEG

.CSEG

.ORG 0

JMP RESET

.ORG $010

JMP TIM1\_OVF

RESET:

LDI R16,HIGH(RAMEND)

OUT SPH,R16

LDI R16,LOW(RAMEND)

OUT SPL,R16

LDI R16,0b11111111

OUT DDRD,R16

OUT DDRC,R16

OUT DDRB,R16

OUT DDRA,R16

LDI R16,0b00000101

OUT TCCR1B,R16

LDI R16,0b00000100

OUT TIMSK,R16

OUT TIFR,R16

LDI R16,0xF0

OUT TCNT1H,R16

OUT TCNT1L,R16

LDI R17,0b00000001

SEI

MAIN:

RJMP MAIN

TIM1\_OVF:

CLI

CPI R17,0b00000001

BREQ LED2

CPI R17,0b00000010

BREQ LED3

CPI R17,0b00000100

BREQ LED4

CPI R17,0b00001000

BREQ LED5

RCALL BUF

VIX:

LDI R16,0xEE

OUT TCNT1H,R16

OUT TCNT1L,R16

LDI R16,0b11111111

SEI

RETI

LED1:

LDI R17,0b00000001

LDI R18,0b01000000

OUT PORTA,R18

LDI R18,0b10110000

OUT PORTB,R18

LDI R18,0b10110000

OUT PORTC,R18

LDI R18,0b01000000

OUT PORTD,R18

RJMP VIX

LED2:

LDI R17,0b00000010

LDI R18,0b00100000

OUT PORTA,R18

LDI R18,0b01011000

OUT PORTB,R18

LDI R18,0b01011000

OUT PORTC,R18

LDI R18,0b00100000

OUT PORTD,R18

RJMP VIX

LED3:

LDI R17,0b00000100

LDI R18,0b00010000

OUT PORTA,R18

LDI R18,0b00101100

OUT PORTB,R18

LDI R18,0b00101100

OUT PORTC,R18

LDI R18,0b00010000

OUT PORTD,R18

RJMP VIX

LED4:

LDI R17,0b00001000

LDI R18,0b00001000

OUT PORTA,R18

LDI R18,0b00010110

OUT PORTB,R18

LDI R18,0b00010110

OUT PORTC,R18

LDI R18,0b00001000

OUT PORTD,R18

RJMP VIX

LED5:

LDI R17,0b00010000

LDI R18,0b00000100

OUT PORTA,R18

LDI R18,0b00001011

OUT PORTB,R18

LDI R18,0b00001011

OUT PORTC,R18

LDI R18,0b00000100

OUT PORTD,R18

RJMP VIX

BUF:

CPI R17,0b00010000

BREQ LED6

CPI R17,0b00100000

BREQ LED7

CPI R17,0b01000000

BREQ LED8

CPI R17,0b10000000

BREQ LED1

RJMP VIX

LED6:

LDI R17,0b00100000

LDI R18,0b00001000

OUT PORTA,R18

LDI R18,0b00010110

OUT PORTB,R18

LDI R18,0b00010110

OUT PORTC,R18

LDI R18,0b00001000

OUT PORTD,R18

RJMP VIX

LED7:

LDI R17,0b01000000

LDI R18,0b00010000

OUT PORTA,R18

LDI R18,0b00101100

OUT PORTB,R18

LDI R18,0b00101100

OUT PORTC,R18

LDI R18,0b00010000

OUT PORTD,R18

RJMP VIX

LED8:

LDI R17,0b10000000

LDI R18,0b00100000

OUT PORTA,R18

LDI R18,0b01011000

OUT PORTB,R18

LDI R18,0b01011000

OUT PORTC,R18

LDI R18,0b00100000

OUT PORTD,R18

RJMP VIX

Лаба 8 (LCD Вывод двух строк)

#define F\_CPU 8000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

#include <stdio.h>

#define e1 PORTD|=0b00001000

#define e0 PORTD&=0b11110111

#define rs1 PORTD|=0b00000100

#define rs0 PORTD&=0b11111011

void port\_ini(void)

{

PORTD=0x00;

DDRD=0xFF;

}

void sendhalfbyte(unsigned char c)

{

\_delay\_ms(15);

c<<=4;

e1;

\_delay\_us(50);

PORTD&=0b00001111;

PORTD|=c;

e0;

\_delay\_us(50);

}

void sendbyte(unsigned char c, unsigned char mode)

{

if (mode==0) rs0;

else rs1;

unsigned char hc=0;

hc=c>>4;

sendhalfbyte(hc); sendhalfbyte(c);

}

void sendchar(unsigned char c)

{

sendbyte(c,1);

}

void LCD\_ini(void)

{

\_delay\_ms(15);

sendhalfbyte(0b00000011);

\_delay\_ms(4);

sendhalfbyte(0b00000011);

\_delay\_us(100);

sendhalfbyte(0b00000011);

\_delay\_ms(1);

\_delay\_ms(1);

sendhalfbyte(0b00000010);

\_delay\_ms(1);

\_delay\_ms(1);

sendbyte(0b00101000, 0);

\_delay\_ms(1);

sendbyte(0b00001100, 0);

\_delay\_ms(1);

sendbyte(0b00000110, 0);

\_delay\_ms(1);

}

void setpos(unsigned char x, unsigned y)

{

char adress;

adress=(0x40\*y+x)|0b10000000;

sendbyte(adress, 0);

}

void sendstring(char str[])

{

int i = 0;

while(str[i]!='\0')

{

sendchar(str[i]);

i++;

}

}

int main(void)

{

port\_ini();

LCD\_ini();

sendstring("KEDRINSKII");

setpos(1,4);

sendstring("DANILA");

\_delay\_ms(3000);

sendbyte(0x01,0);

while(1)

{

asm("NOP");

}

}

АЦП Вывод с помощью столбцов(не мой код, вроде он не работает вообще)

.INCLUDE "m16def.inc"

.LIST

.CSEG

.ORG 0X0000

.equ Delay=100

LDI R16, 0x03

OUT SPH, R16

LDI R16, 0xFF

OUT SPL, R16

OUT DDRB, R16

OUT DDRC, R16

OUT DDRD, R16

LDI R16, 0b11000011

OUT ADMUX, R16

LDI R16, 0b11100110

OUT ADCSRA, R16

start:

IN R16, ADCSRA

LDI R17, 0b00010000

AND R17, R16

CPI R17, 0b00010000

BREQ CHECK

RJMP start

CHECK:

IN R26, ADCL

IN R27, ADCH

CPI R27, 3

BREQ BLINK3

CPI R27, 2

BREQ TwoOrThree

CPI R27, 1

BREQ OneOrTwo

CPI R26, 0b11110110

//BRLT BLINK0

RJMP BLINK1

TwoOrThree:

CPI R26, 0b11100001

//BRLT BLINK2

RJMP BLINK3

OneOrTwo:

CPI R26, 0b11101100

//BRLT BLINK1

RJMP BLINK2

BLINK1:

LDI R29, 0xFF

LDI R30, 0x00

OUT PORTD, R29

OUT PORTC, R30

OUT PORTB, R30

LDI R16, 0b11100110

OUT ADCSRA, R16

RJMP start

BLINK2:

LDI R29, 0xFF

LDI R30, 0x00

OUT PORTD, R29

OUT PORTC, R29

OUT PORTB, R30

LDI R16, 0b11100110

OUT ADCSRA, R16

RJMP start

BLINK3:

LDI R29, 0xFF

OUT PORTD, R29

OUT PORTC, R29

OUT PORTB, R29

LDI R16, 0b11100110

OUT ADCSRA, R16

RJMP start

BLINK0:

LDI R29, 0x00

OUT PORTD, R29

OUT PORTC, R29

OUT PORTB, R29

LDI R16, 0b11100110

OUT ADCSRA, R16

RJMP start

Обработка кнопок 4х4(вывод на диоды, ЗАДАЧА С ЗАЧЕТА)

.INCLUDE "m16def.inc"

.LIST

.CSEG

.ORG 0x0000

.equ DELAY = 0xFF

LDI R16,LOW(RAMEND)

OUT SPL,R16

LDI R16,HIGH(RAMEND)

OUT SPH,R16

LDI R16,0b11111111

OUT DDRA,R16

OUT DDRB,R16

OUT DDRD,R16

MAIN:

LDI R17,0b11110000

OUT DDRC,R17

LDI R16,0b10000000

OUT PORTC,R16

IN R18,PINC

SBRC R18,3

RCALL ONE

IN R18,PINC

SBRC R18,2

RCALL TWO

IN R18,PINC

SBRC R18,1

RCALL THREE

IN R18,PINC

SBRC R18,0

RCALL A

LDI R16,0b01000000

OUT PORTC,R16

IN R18,PINC

SBRC R18,3

RCALL FOUR

IN R18,PINC

SBRC R18,2

RCALL FIVE

IN R18,PINC

SBRC R18,1

RCALL SIX

IN R18,PINC

SBRC R18,0

RCALL B

LDI R16,0b00100000

OUT PORTC,R16

IN R18,PINC

SBRC R18,3

RCALL SEVEN

IN R18,PINC

SBRC R18,2

RCALL EIGHT

IN R18,PINC

SBRC R18,1

RCALL NINE

IN R18,PINC

SBRC R18,0

RCALL C

LDI R16,0b00010000

OUT PORTC,R16

IN R18,PINC

SBRC R18,3

RCALL STAR

IN R18,PINC

SBRC R18,2

RCALL ZERO

IN R18,PINC

SBRC R18,1

RCALL SHARP

IN R18,PINC

SBRC R18,0

RCALL D

RJMP MAIN

ZERO:

OUT DDRC,R16

LDI R19,0b00111100

OUT PORTA,R19

OUT PORTD,R19

LDI R19,0b01000010

OUT PORTC,R19

OUT PORTB,R19

RCALL WAIT

CLR R19

OUT PORTA,R19

OUT PORTB,R19

OUT PORTC,R19

OUT PORTD,R19

LDI R17,0b11110000

OUT DDRC,R17

RETI

ONE:

OUT DDRC,R16

LDI R19,0b01001000

OUT PORTA,R19

LDI R19,0b01000100

OUT PORTB,R19

LDI R19,0b01111110

OUT PORTC,R19

LDI R19,0b01000010

OUT PORTD,R19

RCALL WAIT

CLR R19

OUT PORTA,R19

OUT PORTB,R19

OUT PORTC,R19

OUT PORTD,R19

LDI R17,0b11110000

OUT DDRC,R17

RETI

TWO:

OUT DDRC,R16

SBI PORTA,2

SBI PORTA,5

SBI PORTA,6

SBI PORTB,1

SBI PORTB,4

SBI PORTB,6

SBI PORTC,1

SBI PORTC,3

SBI PORTC,6

SBI PORTD,2

SBI PORTD,6

RCALL WAIT

CLR R19

OUT PORTA,R19

OUT PORTB,R19

OUT PORTC,R19

OUT PORTD,R19

LDI R17,0b11110000

OUT DDRC,R17

RETI

THREE:

OUT DDRC,R16

SBI PORTA,2

SBI PORTA,5

SBI PORTB,1

SBI PORTB,6

SBI PORTC,1

SBI PORTC,3

SBI PORTC,4

SBI PORTC,6

SBI PORTD,2

SBI PORTD,5

RCALL WAIT

CLR R19

OUT PORTA,R19

OUT PORTB,R19

OUT PORTC,R19

OUT PORTD,R19

LDI R17,0b11110000

OUT DDRC,R17

RETI

FOUR:

OUT DDRC,R16

SBI PORTA,1

SBI PORTA,2

SBI PORTA,3

SBI PORTB,3

SBI PORTC,3

SBI PORTD,1

SBI PORTD,2

SBI PORTD,3

SBI PORTD,4

SBI PORTD,5

SBI PORTD,6

RCALL WAIT

CLR R19

OUT PORTA,R19

OUT PORTB,R19

OUT PORTC,R19

OUT PORTD,R19

LDI R17,0b11110000

OUT DDRC,R17

RETI

FIVE:

OUT DDRC,R16

SBI PORTA,1

SBI PORTA,2

SBI PORTA,3

SBI PORTA,5

SBI PORTB,1

SBI PORTB,3

SBI PORTB,6

SBI PORTC,1

SBI PORTC,3

SBI PORTC,6

SBI PORTD,1

SBI PORTD,4

SBI PORTD,5

RCALL WAIT

CLR R19

OUT PORTA,R19

OUT PORTB,R19

OUT PORTC,R19

OUT PORTD,R19

LDI R17,0b11110000

OUT DDRC,R17

RETI

SIX:

OUT DDRC,R16

SBI PORTA,2

SBI PORTA,3

SBI PORTA,4

SBI PORTA,5

SBI PORTB,1

SBI PORTB,4

SBI PORTB,6

SBI PORTC,1

SBI PORTC,4

SBI PORTC,6

SBI PORTD,2

SBI PORTD,5

RCALL WAIT

CLR R19

OUT PORTA,R19

OUT PORTB,R19

OUT PORTC,R19

OUT PORTD,R19

LDI R17,0b11110000

OUT DDRC,R17

RETI

SEVEN:

OUT DDRC,R16

SBI PORTA,1

SBI PORTB,1

SBI PORTB,4

SBI PORTB,5

SBI PORTB,6

SBI PORTC,1

SBI PORTC,3

SBI PORTD,1

SBI PORTD,2

RCALL WAIT

CLR R19

OUT PORTA,R19

OUT PORTB,R19

OUT PORTC,R19

OUT PORTD,R19

LDI R17,0b11110000

OUT DDRC,R17

RETI

EIGHT:

OUT DDRC,R16

LDI R19,0b00100100

OUT PORTA,R19

LDI R19,0b01011010

OUT PORTB,R19

LDI R19,0b01011010

OUT PORTC,R19

LDI R19,0b00100100

OUT PORTD,R19

RCALL WAIT

CLR R19

OUT PORTA,R19

OUT PORTB,R19

OUT PORTC,R19

OUT PORTD,R19

LDI R17,0b11110000

OUT DDRC,R17

RETI

NINE:

OUT DDRC,R16

LDI R19,0b0010100

OUT PORTA,R19

LDI R19,0b0101010

OUT PORTB,R19

LDI R19,0b0101010

OUT PORTC,R19

LDI R19,0b0011100

OUT PORTD,R19

RCALL WAIT

CLR R19

OUT PORTA,R19

OUT PORTB,R19

OUT PORTC,R19

OUT PORTD,R19

LDI R17,0b11110000

OUT DDRC,R17

RETI

A:

OUT DDRC,R16

LDI R19,0b0010100

OUT PORTA,R19

LDI R19,0b0101010

OUT PORTB,R19

LDI R19,0b0101010

OUT PORTC,R19

LDI R19,0b0011100

OUT PORTD,R19

RCALL WAIT

CLR R19

OUT PORTA,R19

OUT PORTB,R19

OUT PORTC,R19

OUT PORTD,R19

LDI R17,0b11110000

OUT DDRC,R17

RETI

B:

OUT DDRC,R16

LDI R19,0b0010100

OUT PORTA,R19

LDI R19,0b0101010

OUT PORTB,R19

LDI R19,0b0101010

OUT PORTC,R19

LDI R19,0b0011100

OUT PORTD,R19

RCALL WAIT

CLR R19

OUT PORTA,R19

OUT PORTB,R19

OUT PORTC,R19

OUT PORTD,R19

LDI R17,0b11110000

OUT DDRC,R17

RETI

C:

OUT DDRC,R16

LDI R19,0b0010100

OUT PORTA,R19

LDI R19,0b0101010

OUT PORTB,R19

LDI R19,0b0101010

OUT PORTC,R19

LDI R19,0b0011100

OUT PORTD,R19

RCALL WAIT

CLR R19

OUT PORTA,R19

OUT PORTB,R19

OUT PORTC,R19

OUT PORTD,R19

LDI R17,0b11110000

OUT DDRC,R17

RETI

D:

OUT DDRC,R16

LDI R19,0b0010100

OUT PORTA,R19

LDI R19,0b0101010

OUT PORTB,R19

LDI R19,0b0101010

OUT PORTC,R19

LDI R19,0b0011100

OUT PORTD,R19

RCALL WAIT

CLR R19

OUT PORTA,R19

OUT PORTB,R19

OUT PORTC,R19

OUT PORTD,R19

LDI R17,0b11110000

OUT DDRC,R17

RETI

STAR:

OUT DDRC,R16

LDI R19,0b0010100

OUT PORTA,R19

LDI R19,0b0101010

OUT PORTB,R19

LDI R19,0b0101010

OUT PORTC,R19

LDI R19,0b0011100

OUT PORTD,R19

RCALL WAIT

CLR R19

OUT PORTA,R19

OUT PORTB,R19

OUT PORTC,R19

OUT PORTD,R19

LDI R17,0b11110000

OUT DDRC,R17

RETI

SHARP:

OUT DDRC,R16

LDI R19,0b0010100

OUT PORTA,R19

LDI R19,0b0101010

OUT PORTB,R19

LDI R19,0b0101010

OUT PORTC,R19

LDI R19,0b0011100

OUT PORTD,R19

RCALL WAIT

CLR R19

OUT PORTA,R19

OUT PORTB,R19

OUT PORTC,R19

OUT PORTD,R19

LDI R17,0b11110000

OUT DDRC,R17

RETI

WAIT:

LDI R20, DELAY

LOOP1:

LDI R21,0X45

LOOP2:

LDI R22,0x12

LOOP3:

DEC R22

BRNE LOOP3

DEC R21

BRNE LOOP2

DEC R20

BRNE LOOP1

RET

Мигаем светодиодом(Задача с зачета, диоды моргают один за одним в одном столбце)

#include <avr/io.h>

#include <util/delay.h>

int main(void) {

DDRA = 0xFF;

PORTA = 0x01;

while (1) {

PORTA = PORTA<<1;

if(PORTA == 0x00)

PORTA = 0x01;

\_delay\_ms(100);

}

}

На зачете еще были задачи Смайлик на прерываниях, вывод ADC на LCD, Вывод смалика на С, Считывание кнопки и вывод спирали на диоды, змейка на прирываниях таймера

Удачи на зачете!