**Московский Государственный Технический Университет им.Н.Э.Баумана**

Кафедра ИУ5

**“Системы обработки информации и управления”**

Отчет по лабораторной работе №4

**«Формирование временных интервалов с помощью микроконтроллеров (Таймер ATmega16)»**

**Вариант ХХХ**

Исполнитель: Преподаватель:

Москва 2020

1. Исходные данные для лабораторной работы 4

последовательность импульсов:

Вариант ХХХ: **1 1 3 2 2 3 4 1 5 4 7 50**

1. Алгоритмы работы системы.

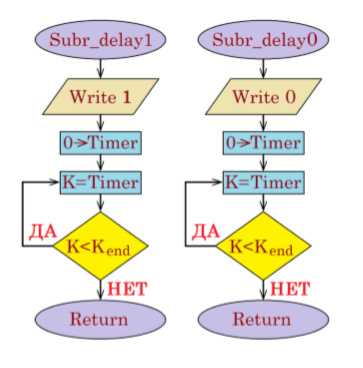
**Алгоритм формирования периодического сигнала, ждущий режим.**



**Алгоритм формирования периодического сигнала, автоколебательный режим.**



**Алгоритм работы функций subr\_delay0 и subr\_delay1**

****

1. Программа для микроконтроллера ATmega16, реализующая разработанный алгоритмы.

**Ждущий режим**

start:

.include "m16def.inc"

.list

.def t=r20

.def temp=r16

.def kold=r17

.def k=r18

.def s=r19

.def k\_\_z=r20

.def k\_\_\_=r21

.def s\_\_\_=r22

;--------------------------------------------

.cseg

.org 0

;--------------------------------------------

ldi temp,0x80

out acsr,temp

;--------------------------------------------

ldi temp,0x00 ; 0 --> temp

out ddrd,temp

ldi temp,0xFF ; 0xff --> temp

out ddrb,temp

out portd,temp

;---------------------------------------------

ldi kold, 0x00 ; 0--->kold

ldi s, 0x00 ; 0--->s\_\_\_

out portb, s ; s\_\_\_(=0)

;---------------------------------------------

ldi temp, 0b101 ; Предделение 1024

out tccr0, temp

ldi temp, 135 ; Коррекция тактовой частоты

out osccal, temp

ldi temp,low(RAMEND) ; инициализация стека

out spl,temp

ldi temp,high(RAMEND)

out sph,temp

;---------------------------------------------

read: ; метка read

in k, pind ; Считали содержимое порта pd (--->k

cp k, kold ; Сравнили k и kold

breq read ; Если k=kold, read

tst kold ; Проверили kold

brne remem ; Если kold!=0, remen

tst k ; Проверили k

breq remem ; Если k=0, remem

jmp lbl1 ; переход к lbl1

;---------------------------------------------

remem: ; метка remem

mov kold, k ; помещаем k в kold

jmp read ; переход к read

;---------------------------------------------

lbl1: ; метка lbl1

cpi k, 0x01 ; проверка условия k = 1

breq met ; если условме выполняется, то met

jmp remem ; переход к remem

met: ; метка met

ldi k\_\_z, 1 ; 1--->k\_\_z

rcall subr\_delay1 ; вызов subr\_delay1

ldi k\_\_z, 1 ; 1--->k\_\_z

rcall subr\_delay0 ; вызов subr\_delay0

ldi k\_\_z, 3 ; и тд

rcall subr\_delay1

ldi k\_\_z, 2

rcall subr\_delay0

ldi k\_\_z, 2

rcall subr\_delay1

ldi k\_\_z, 3

rcall subr\_delay0

ldi k\_\_z, 4

rcall subr\_delay1

ldi k\_\_z, 1

rcall subr\_delay0

ldi k\_\_z, 5

rcall subr\_delay1

ldi k\_\_z, 3

rcall subr\_delay0

ldi k\_\_z, 7

rcall subr\_delay1

ldi k\_\_z, 60

rcall subr\_delay0

jmp remem ; переход к remem

subr\_delay1: ; "1" длится k\_\_z тактов с предделением

ldi s\_\_\_, 1 ; 1 --> s\_\_\_

out portb, s\_\_\_ ; s\_\_\_ --> pb

ldi temp, 0

out tcnt0, temp ; 0 --> tcnt0 Обнуление таймера

ccc1: ; повтор цикла

in k\_\_\_, tcnt0 ; считали таймер

cp k\_\_\_, k\_\_z ; сравнили k\_\_ и k\_\_z

brlo ccc1 ; если k\_\_\_<k\_\_z, ушли в начало

ret ; конец подпрограммы subr\_delay1

subr\_delay0: ; "0" длится k\_\_z тактов с предделением

ldi s\_\_\_, 0 ; 0 --> s\_\_\_

out portb, s\_\_\_ ; s\_\_\_ --> pb

ldi temp, 0

out tcnt0, temp ; 0 --> tcnt0

ccc0: ; повтор цикла

in k\_\_\_, tcnt0 ; считали таймер

cp k\_\_\_, k\_\_z ; сравнили k\_\_ и k\_\_z

brlo ccc0 ; если k\_\_\_<k\_\_z, ушли в начало

ret ; конец подпрограммы subr\_delay

**Автоколебательный режим**

start:

; inc r16

; rjmp start

.include "m16def.inc" ; подключение библиотеки для работы с ATmega16

.list ; включение листинга

.def t=r20

.def temp=r16

.def kold=r17

.def k=r18

.def s=r19

.def k\_\_z=r20

.def k\_\_\_=r21

.def s\_\_\_=r22

;--------------------------------------------

.cseg ; выбор сегмента программного кода

.org 0 ; установка текущего адреса на ноль

;--------------------------------------------

ldi temp,0x80 ; выключение компаратора

out acsr,temp

;--------------------------------------------

ldi temp,0x00 ; 0 --> temp

out ddrd,temp ; Назначаем порт rd на ввод (00000000 --> ddrd)

ldi temp,0xFF ; 0xff --> temp

out ddrb,temp ; Назначаем порт rb на вывод (11111111 --> ddrb)

;---------------------------------------------

ldi temp, 0b101 ; Предделение 1024

out tccr0, temp

ldi temp, 135 ; Коррекция тактовой частоты

out osccal, temp

ldi temp,low(RAMEND) ; инициализация стека

out spl,temp

ldi temp,high(RAMEND)

out sph,temp

ldi temp, 0

met: ; метка met

ldi k\_\_z, 1 ; 1--->k\_\_z

rcall subr\_delay1 ; вызов subr\_delay1

ldi k\_\_z, 1 ; 1--->k\_\_z

**rcall subr\_delay0 ; вызов subr\_delay0**

ldi k\_\_z, 3 ; и тд

rcall subr\_delay1

ldi k\_\_z, 2

**rcall subr\_delay0**

ldi k\_\_z, 2

**rcall subr\_delay1**

ldi k\_\_z, 3

**rcall subr\_delay0**

ldi k\_\_z, 4

**rcall subr\_delay1**

ldi k\_\_z, 1

**rcall subr\_delay0**

ldi k\_\_z, 5

**rcall subr\_delay1**

ldi k\_\_z, 3

**rcall subr\_delay0**

ldi k\_\_z, 7

**rcall subr\_delay1**

ldi k\_\_z, 60

**rcall subr\_delay0**

jmp met ; переход к met

**subr\_delay1: ; "1" длится k\_\_z тактов с предделением**

ldi s\_\_\_, 1 ; 1 --> s\_\_\_

out portb, s\_\_\_ ; s\_\_\_ --> pb

out tcnt0, temp ; 0 --> tcnt0 Обнуление таймера

ccc1: ; повтор цикла

in k\_\_\_, tcnt0 ; считали таймер

cp k\_\_\_, k\_\_z ; сравнили k\_\_ и k\_\_z

brlo ccc1 ; если k\_\_\_<k\_\_z, ушли в начало

ret ; конец подпрограммы subr\_delay1

**subr\_delay0: ; "0" длится k\_\_z тактов с предделением**

ldi s\_\_\_, 0 ; 0 --> s\_\_\_

out portb, s\_\_\_ ; s\_\_\_ --> pb

out tcnt0, temp ; 0 --> tcnt0

ccc0: ; повтор цикла

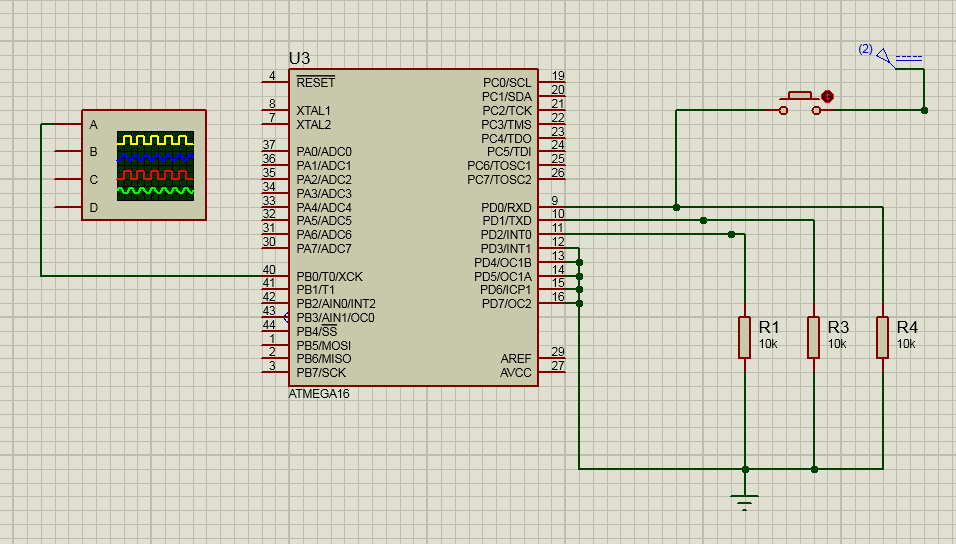
in k\_\_\_, tcnt0 ; считали таймер

cp k\_\_\_, k\_\_z ; сравнили k\_\_ и k\_\_z

brlo ccc0 ; если k\_\_\_<k\_\_z, ушли в начало

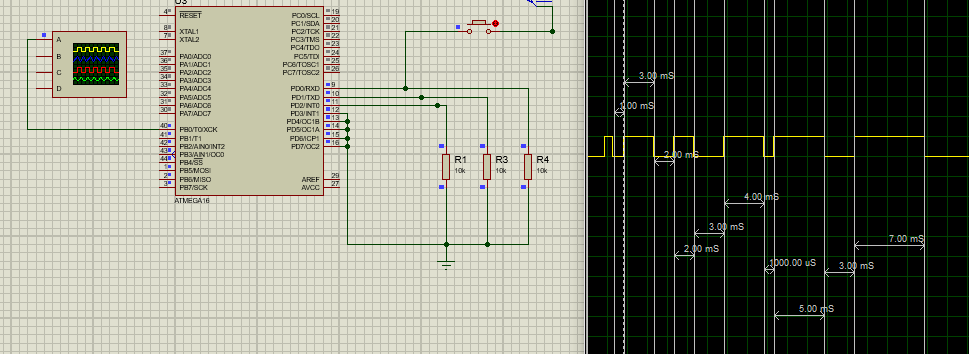
ret ; конец подпрограммы subr\_delay

1. **Схема системы в программе PROTEUS**

****

1. **Функционирование разработанной системы в программе PROTEUS.**

**Ждущий режим**



**Автоколебательный режим**

