



Министерство науки и высшего образования Российской  
Федерации Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

---

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

**О Т Ч Е Т**

**по лабораторной работе № 4**

**Название:** Таймеры микроконтроллера ATx 8515

**Дисциплина:** Микропроцессорные системы

Студент

ИУ6-66Б

(Группа)

\_\_\_\_\_  
(Подпись, дата)

Р.М. Аксенов

(И.О. Фамилия)

В.Н. Гокарев

(И.О. Фамилия)

Т.А. Тищенко

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

А.К. Халайджи

(И.О. Фамилия)

Москва, 2020

## Цель работы:

- изучение структур и режимов работы таймеров и их программирование;
- анализ схем выключения таймеров для проведения исследований;
- программирование задач с таймером.

## Задание 1

Проверить на плате STK500 работу исходной программы 4.1. Изменить программу, исключив влияние на работу таймера возможность “дребезга” кнопки.

Программа с исключением влияния на работу таймера возможности «дребезга» кнопки.

«дребезга» кнопки.

```
;*****  
***
```

```
;Программа 4.1 для МК ATx8515:
```

```
;демонстрация работы таймера T0 в режиме счётчика  
событий; ;событие – нажатие кнопки SW0.
```

```
;Соединения: порт PB0-SW0, шлейфом порт PD-LED ;Светодиоды  
включаются после четвертого нажатия кнопки
```

```
SW0 ;*****  
**
```

```
***                                определений ATmega8515
```

```
.include "m8515def.inc" ;файл
```

```
.def temp = r16 ;временный регистр
```

```
;***Таблица векторов прерываний
```

```
.org $000
```

```
    rjmp INIT ;обработка сброса
```

```
.org $007 ;обработка переполнения таймера T0
```

```
    rjmp T0_OVF
```

```
;***Инициализация МК
```

```
INIT:
```

```
    ldi temp,low(RAMEND) ;установка
```

```
    out SPL,temp ; указателя стека
```

```
    ldi temp,high(RAMEND) ; на последнюю
```

```
    out SPH,temp ;ячейку ОЗУ
```

```
    ldi temp,0x01 ;инициализация вывода порта PB0
```

```
    out DDRB,temp ; на выход, а PB7 на вход
```

```
    ldi temp,0x80 ;включение 'подтягивающего' резистора
```

```
    out PORTB,temp ; входа PB7
```

```
    ser temp ;инициализация выводов порта PD
```

```
    out DDRD,temp ; на вывод
```

```
    out PORTD,temp ;выключение светодиодов
```

```
    ldi temp,(1<<SE) ;разрешение перехода
```

```
    out MCUCR,temp ; в режим Idle
```

```
;***Настройка таймера T0 на режим счётчика событий
```

```
    ldi temp,0x02 ;разрешение прерывания по
```

```

    out TIMSK,temp ; переполнению таймера T0
    ldi temp,0x07 ;переключение таймера
    out TCCR0,temp ; по положительному перепаду напряжения
    sei ;глобальное разрешение прерываний ldi temp,0xFC ;
    $FC=-4 для
    out TCNT0,temp ; отсчёта 4-х нажатий
LOOP:
    sbic PINB, 7
    rjmp CHK_DOWN
    cbi PORTB, 0
    sbi PORTB, 0
    cbi PORTB, 0
WAIT_BTN_UP:
    sbis PINB, 7
    rjmp WAIT_BTN_UP
    rcall DELAY_HALF_SECOND
CHK_DOWN:
    rjmp LOOP
;***Обработка прерывания при переполнении таймера T0
T0_OVF:
    clr temp
    out PORTD,temp ;включение светодиодов
    rcall DELAY ;задержка
    ser temp
    out PORTD,temp ;выключение
    светодиодов ldi
    temp,0xFC ;перезагрузка out
    TCNT0,temp ; TCNT0
    reti
;*** Задержка ***
DELAY:
    ldi r19,6
    ldi r20,255
    ldi r21,255
dd:
    dec r21
    brne dd
    dec r20
    brne dd
    dec r19
    brne dd
    ret

DELAY_HALF_SECOND:
    ldi r19, 3
    ldi r20, 255
    ldi r21, 255
dd2:
    dec r21
    brne dd2
    dec r20
    brne dd2

```

```

dec r19
brne dd2
ret

```

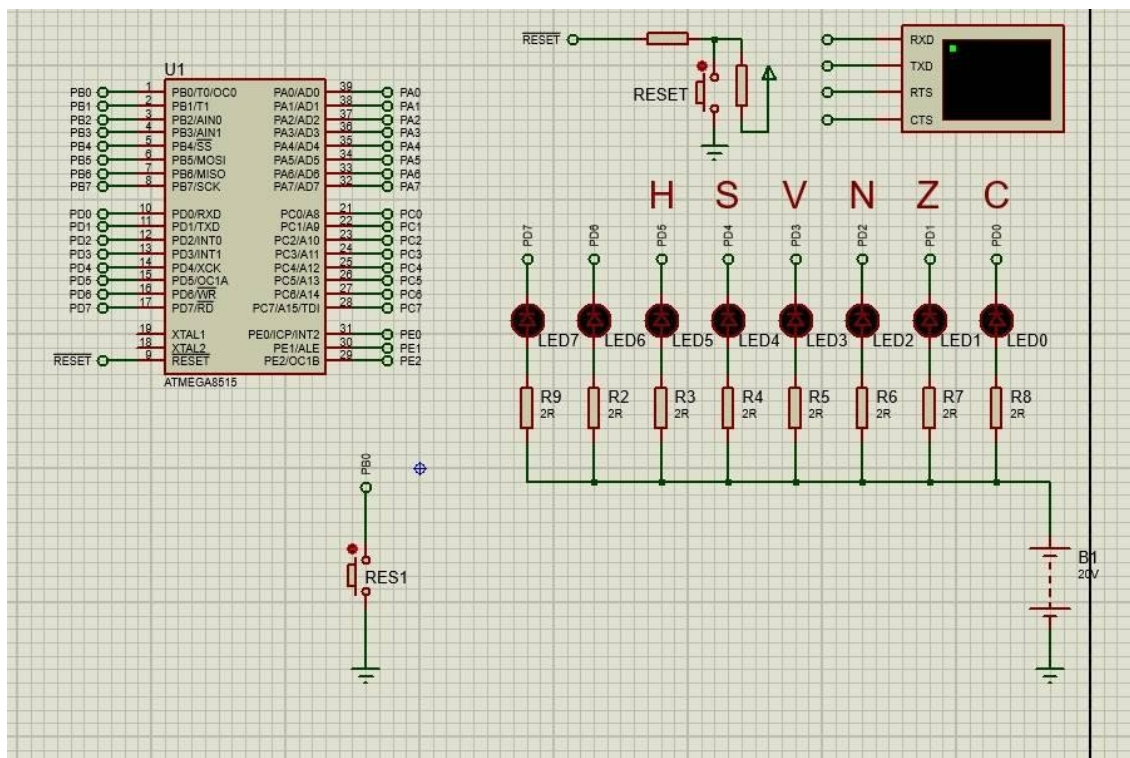


Рисунок 1. Схема таймера/счетчика T0 в Proteus 8 Pro

В данном задании был использован режим счетчика у таймера T0 микроконтроллера. В таком режиме он может вести подсчет событий. Здесь событием такого типа стало нажатие кнопки, подключенной к выводу микроконтроллера, закрепленному за данным счетчиком как «вывод внешнего сигнала». Для подавления дребезга использован программный метод. После фиксации нажатия кнопки программа ожидает некоторое время чтобы дать возможность пользователю отпустить кнопку и пропустить дребезг, за этим следующий.

## Задание 2. Режим таймера.

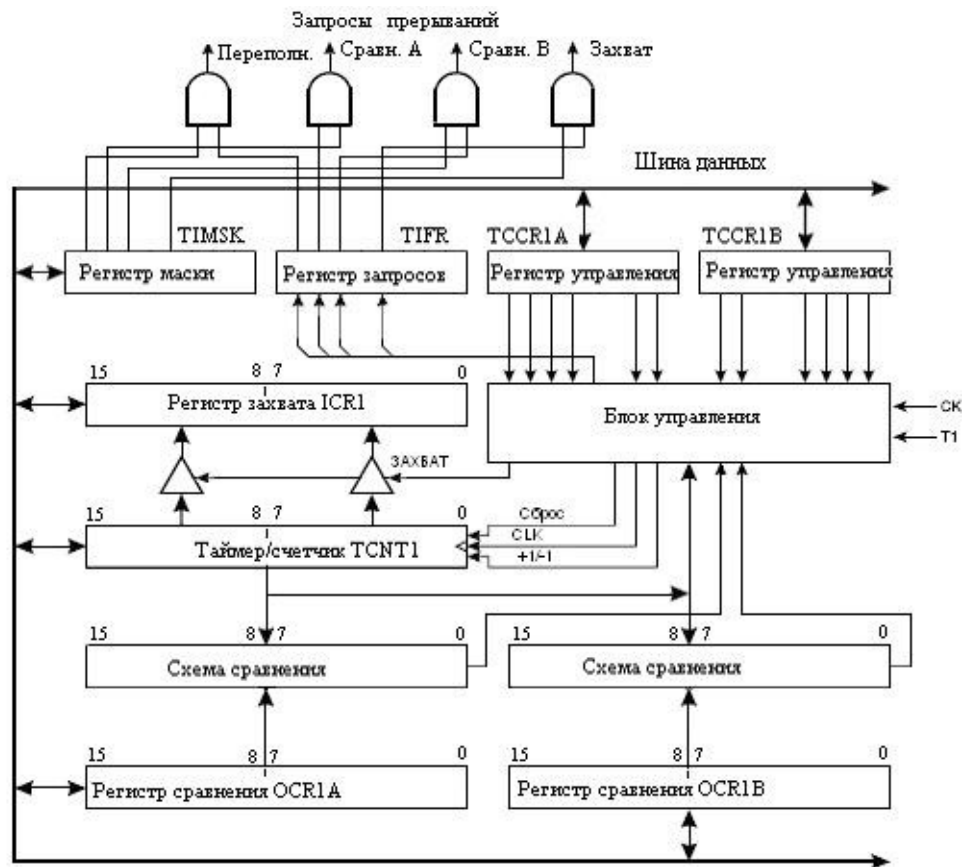


Рисунок 2. Структурная схема таймера/счетчика T1

$$T = (65536 - TCNT1) * K / F_{ck}$$

$F_{ck} = 3.69\text{MHz}$  - Частота счетных импульсов

$K_{SW0} = 1024$  - Коэффициент делителя

$K_{SW1} = 256$  - Коэффициент делителя

Начальное значение счётчика =  $0x8000 = 32\,768$

$$T_{SW0} = (65536 - 32\,768) * 1024 / 3.69 / 10^{**6} = 9.09\text{с}$$

$$T_{SW1} = (65536 - 32\,768) * 256 / 3.69 / 10^{**6} = 2.27\text{с}$$

Исходная схема задание 4.2:

\* Кнопка 0 11с - на практике, 9.09с - теоретическое;

\* Кнопка 1 2.8с - на практике, 2.27с -

теоретическое;  $T_{SW0} / T_{SW1} = 4$

Расчёт теоретического времени таймера в 4 раза меньше исходного:

$$T\_SW0 = 2.27с = (65536 - 32768) * 256 / 3.69 / 10^{**6} = 2.27с$$

$$T\_SW1 = 0.57с = (65536 - 32768) * 64 / 3.69 / 10^{**6} = 0.57с$$

$$T\_SW0 / T\_SW1 = 4$$

Измененная схема задания 4.2:

- Кнопка 0 2.86с - на практике, 2.27с - теоретическое;
- Кнопка 1 0.5с - на практике, 0.57с - теоретическое;

Текст изменённой программы:

```
;*****
*****
;Программа 4.2 для МК АТх8515: демонстрация работы таймера T1 ;в
режиме таймера. При частоте работы микроконтроллера FCK =3,69
МГц
;после нажатия SW0 на вход счётчика поступают сигналы с частотой
FCK/1024,
;после нажатия SW1 - FCK /256.
;Соединения: PD0,PD1-SW0,SW1; PB-LED
;*****
*****
;файл определений AT90S8515
;.include "8515def.inc"
;.include "m8515def.inc" ;файл определений ATmega8515
.def temp = r16 ;временный регистр
.equ SW0 = 0 ;0-ой вывод порта PD
.equ SW1 = 1 ;1-ий вывод порта PD

;***Таблица векторов прерываний
.org $000
rjmp INIT ;обработка сброса
.org $006
rjmp T1_OVF ;обработка переполнения таймера T0

;***Инициализация МК
INIT:
ldi temp,low(RAMEND);установка
out SPL,temp ; указателя стека
ldi temp,high(RAMEND) ;на последнюю
out SPH,temp ;ячейку ОЗУ
clr temp ;инициализация выводов порта PD
out DDRD,temp ; на ввод
ldi temp,0x03 ;включение 'подтягивающих' резисторов
out PORTD,temp ; в разрядах 0,1 порта PD
ser temp ;инициализация выводов порта PB
out DDRB,temp ; на вывод
```

```

        out PORTB,temp ;выключение
светодиодов ;***Настройка таймера T1 на
режим таймера

        ldi temp,0x80 ;разрешение прерывания по
        out TIMSK,temp ; переполнению таймера T1
        clr temp ;таймер T1
        out TCCR1B,temp ; остановлен
        ldi temp,0x80 ; загрузка TCNT1
        out TCNT1H,temp
        ldi temp,0x00
        out TCNT1L,temp
        sei ;глобальное разрешение прерываний

;***Ожидание нажатия кнопок

test_sw0:                ;проверка нажатия
        sbic PIND,SW0
        rjmp test_sw1 ;кнопки SW0
;***Обработка нажатия кнопки SW0
        ldi temp,0x04 ;для настройки предделителя (K=256)
        rcall LED_ON ;включение светодиодов
test_sw1:                ;проверка нажатия
        sbic PIND,SW1
        rjmp test_sw0 ;кнопки SW1
;***Обработка нажатия кнопки SW1
        ldi temp,0x03 ; для настройки предделителя (K=64)
        rcall LED_ON ;включение светодиодов
        rjmp test_sw0

;***Включение светодиодов

LED_ON:                ;запуск таймера с предделителем
        out TCCR1B,temp
        clr temp ;включение
        out PORTB,temp ; светодиодов
        ret

;***Обработка прерывания при переполнении таймера
T1 T1_OVF:
        ser temp
        out PORTB,temp ;выключение светодиодов
        clr temp ;останов ; таймера T1
        out TCCR1B,temp
        ldi temp,0x80 ; перезагрузка TCNT1
        out TCNT1H,temp
        ldi temp,0x00
        out TCNT1L,temp
        reti

```

В этом задании задействован таймер T1. Задание обыгрывает использование различных значений предделителя. Предделитель нужен таймеру для отсчета значимого для человека отрезка времени. Если подсчитывать чистую тактовую частоту процессора без использования предделителя, то счетчик даже с регистром в два байта (TCNT1) очень быстро (его максимальное значение ~65 тысяч, при частоте работы процессора от миллиона тактов в секунду).

Поэтому используется предделитель, он делит воспринимаемую счетчиком частоту на 8, 64, 256 или 1024, в зависимости от режима. «Делит» - означает, что, к примеру, на каждые 256 тактов работы процессора счетчик увеличивает своё значение на один

### Задание 3. Функция сравнения

По заданию:  $t_i = 6$  с,  $t_z = 2$  с. Рассчитаем значения регистров по следующим формулам:

- $OCR1A = t_i \cdot F_{ck} / K = 6 \cdot 3,69 \cdot 10^6 / 1024 \approx 21621 = 0x5475$ ;
- $t_z \cdot F_{ck} / K = 2 \cdot 3,69 \cdot 10^6 / 1024 \approx 7207 = 0x1C27$ ;
- Т.к.  $t_z = (OCR1A - OCR1B) \cdot K / F_{ck}$ , то  $OCR1B = 5475 - 1C27 = 384D$ .

Текст переработанной программы:

```
;*****
*****
;.include "8515def.inc" ;файл определений AT90S8515
.include "m8515def.inc" ;файл определений ATmega8515
.def temp = r16 ;временный регистр
.equ START = 0 ;0-ой вывод порта PD
.org $000
rjmp INIT ;обработка сброса
.org $001
rjmp STOP_PRESSED ;обработка внешнего прерывания INT0 -
; нажатие

STOP ;***Инициализац
ия МК
```



```

INIT:      ldi temp,low(RAMEND) ;установка
           out SPL,temp ; указателя стека
           ldi temp,high(RAMEND) ; на последнюю
           out SPH,temp ; ячейку ОЗУ
           ldi temp,0x20 ;инициализация вывода PD5
           out DDRD,temp ; как выхода
           ldi temp,0x05 ;включение 'подтягивающих' резисторов
           out PORTD,temp ; в PD0, PD2
порта      ldi temp,0x04 ;/// для ATmega8515 инициализация вывода
           out DDRE,temp ;/// PE2 (OC1B) на вывод

           ldi temp,(1<<INT0) ;разрешение прерывания INT0
           out GICR,temp ; в регистре GICR (или GIMSK)
           clr temp ;обработка прерывания INT0
           out MCUCR,temp ; по низкому уровню
;***Настройка функции сравнения таймера T1
           cli ;запрещение прерываний
           ldi temp,0x50 ;при сравнении состояния выводов OC1A и
           out TCCR1A,temp ; OC1B изменяются на противоположные
           clr temp ;останов
           out TCCR1B,temp ; таймера

           ldi temp,0x38 ;запись числа в

           out OCR1BH,temp ; регистр сравнения,
           ldi temp,0x4D ; первым записывается
           out OCR1BL,temp ; старший байт
           ldi temp,0x54 ;запись числа в
           out OCR1AH,temp ; регистр сравнения,
           ldi temp,0x75 ; первым записывается
           out OCR1AL,temp ; старший байт
           clr temp ;обнуление
           out TCNT1H,temp ; счётного
           out TCNT1L,temp ; регистра
           sei ;разрешение прерываний
WAITSTART: sbic PIND,START ;ожидание нажатия
           rjmp WAITSTART ; кнопки START
           ldi temp,0x0D ;запуск таймера с предделителем K=1024,
           out TCCR1B,temp ; при совпадении с OCR1A - сброс
           LOOP: nop ;во время цикла происходит
           rjmp LOOP ; увеличение содержимого счётного регистра
;***Обработка прерывания от кнопки STOP
STOP_PRESSED: temp ;останов
           clr
           out TCCR1B,temp ; таймера
WAITSTART_2: ;ожидание
           sbic PIND,START ; нажатия
           rjmp WAITSTART_2 ; кнопки START
           ldi temp,0x0D ;запуск
           out TCCR1B,temp ; таймера с предделителем K=1024
           reti

```

Диаграмма выходных сигналов:

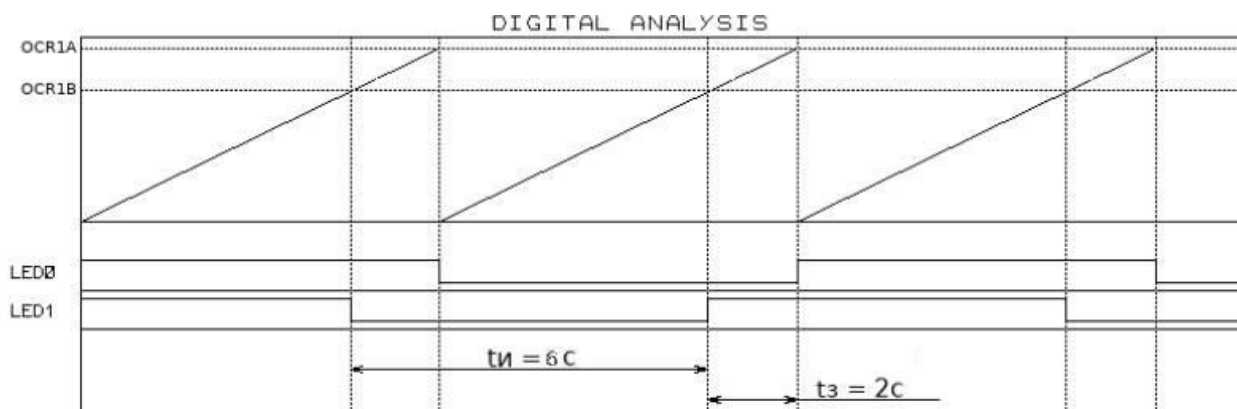


Рисунок 4. Диаграмма выходных сигналов

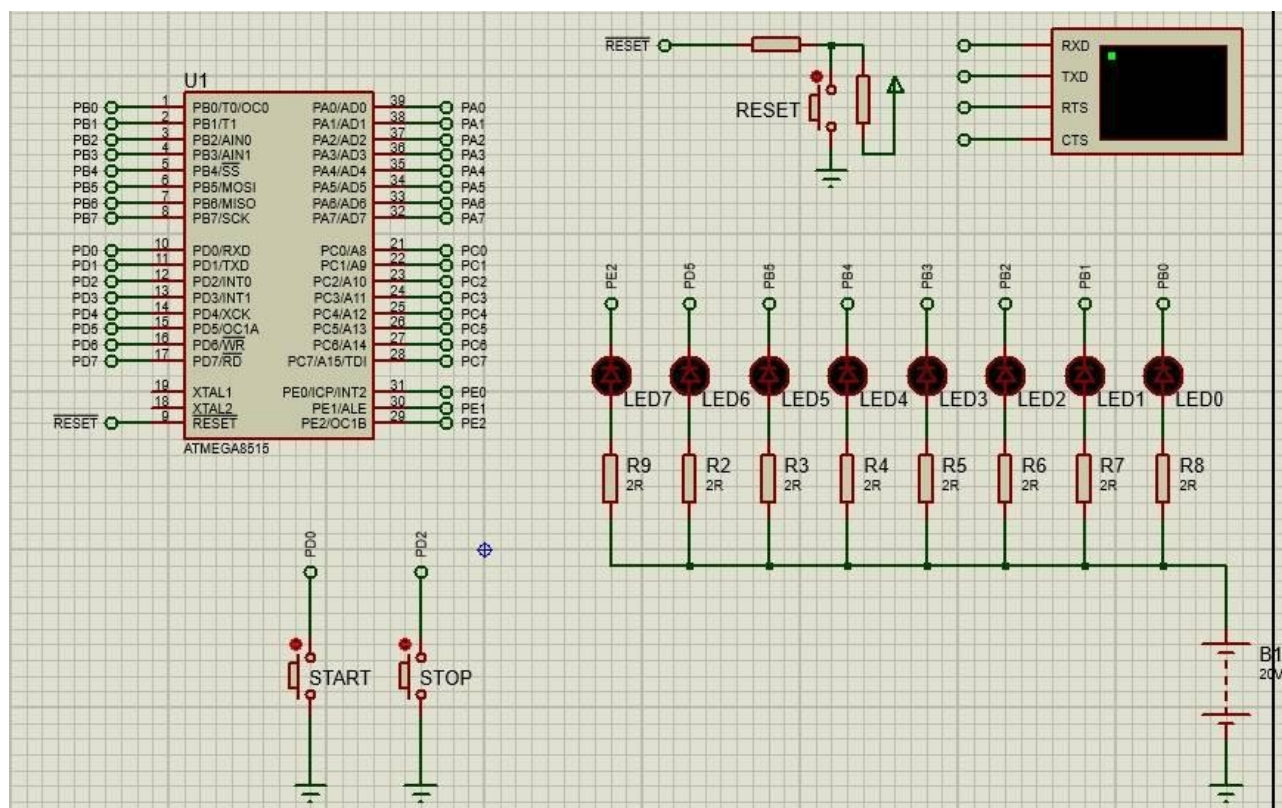


Рисунок 5. Реализация функции сравнения в Proteus 8

В этом задании используется функция сравнения таймера. Она используется для сравнения текущего значения таймера с регистром сравнения. Позволяет вызывать прерывание не только по переполнению счетчика, но и по достижении им определенного значения, заданного в регистре сравнения. В рассматриваемом микроконтроллере есть два регистра сравнения. Работают они независимо друг от друга, каждый такт происходит сравнение значения счетчика (TCNT1) с каждым из регистров сравнения.

## Задание 4. Режим ШИМ

Согласно заданию, необходимо изменить программу для формирования двух выходных ШИМ-сигналов с коэффициентом заполнения  $\frac{3}{4}$  для 10-разрядного режима ШИМ.

ШИМ 10-разрядный, поэтому модуль счёта TOP=1023. K = 1024.

$$T = 2 * TOP * K / F_{ck} = 2 * 1023 * 1024 / (3,69 * 10^6) \approx 0,57c$$

$$t_{и} = 3/4 T \approx 0,43 c$$

$$t_{з} = 1/4 T \approx 0,14 c$$

Тогда для F1:

$$OCR1A = OCR1B = 1/4 * TOP \approx 256 = 0x100$$

Для F2:

$$OCR1A = 3/4 * TOP \approx 767 = 0x2FF$$

Текст изменённой программы:

```
*****
;
*****      "8515def.inc" ;файл определений AT90S8515
; .include
; .include "m8515def.inc" ;файл определений ATmega8515
; .def temp = r16 ;временный регистр
; ***Выходы порта PD
; .equ SW0 = 0
; .equ SW1 = 1
; .org $000 ;обработка сброса
rjmp INIT
; ***Инициализация МК      ;инициализация PD5
INIT:      ldi temp,0x20
out DDRD,temp ; на вывод;включение 'подтягивающих'
          ldi temp,0x03
          out PORTD,temp ; резисторов порта PD
          ldi temp,0x04 ;/// для ATmega8515 инициализация
          out DDRE,temp ;/// PE2 (OC1B) на вывод
          cli ;запрещение прерываний
;настройка таймера: 10-разрядный режим ШИМ, на выводе
;OC1A неинвертированный сигнал, OC1B - инвертированный сигнал
          ldi temp,0xB3
          out TCCR1A,temp
          clr temp ; обнуление
          out TCNT1H,temp ; счётного
          out TCNT1L,temp ; регистра
          ldi temp,0x05 ;таймер
          out TCCR1B,temp ; запущен с делителем 1024
```

```

F1: sbic PIND,SW0 ;проверка нажатия SW0
    rjmp F2
;***Установка порога F1 ;запись числа в
    ldi temp,0x01
    out OCR1AH,temp ; регистры сравнения,
    out OCR1BH,temp ; первым записывается
    ldi temp,0x00 ; старший байт
    out OCR1AL,temp
F2:    out OCR1BL,temp
    sbic PIND,SW1 ; проверка нажатия SW1
    rjmp F1
;*** Установка порога F2 ;запись числа в
    ldi temp,0x02
    out OCR1AH,temp ; регистры сравнения,
    out OCR1BH,temp ; первым записывается
    ldi temp,0xff ; старший байт
    out OCR1AL,temp
    out OCR1BL,temp
    rjmp F1

```

Диаграмма формирования ШИМ-сигнала для режима F1:

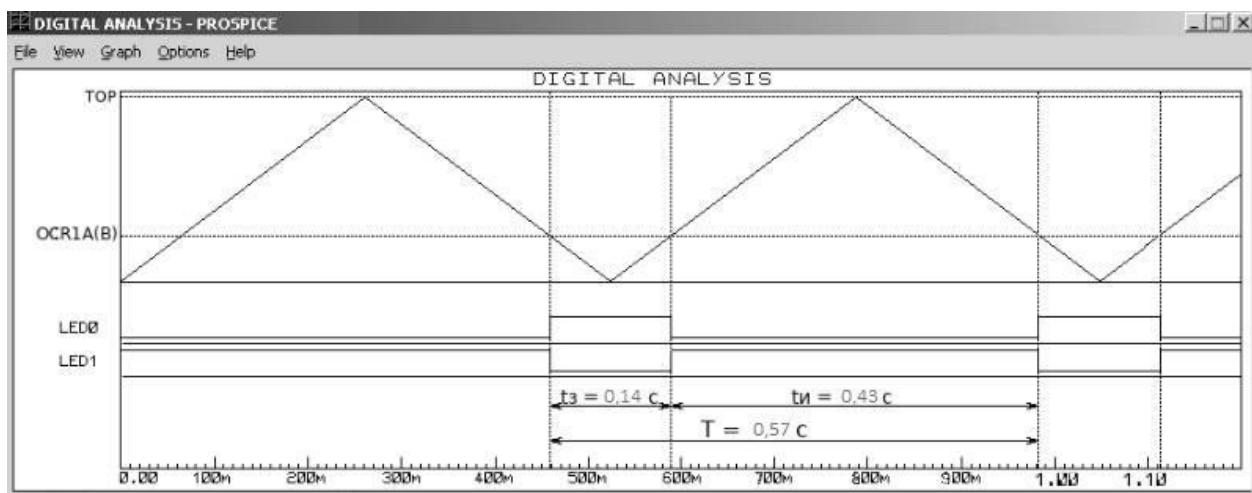


Рисунок 6. Диаграмма формирования ШИМ-сигнала для режима F1

Диаграмма формирования ШИМ-сигнала для режима F2:

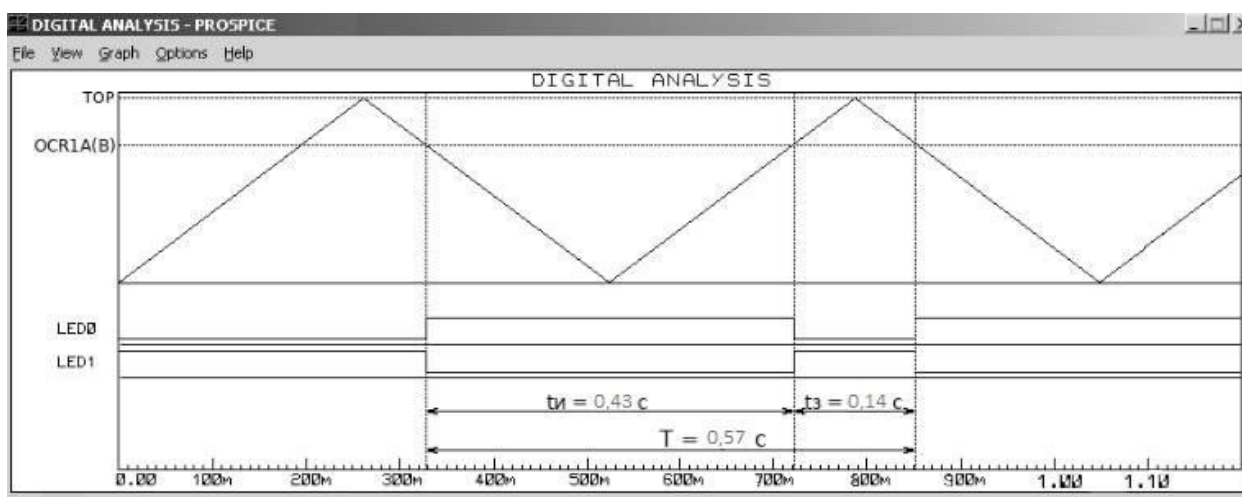


Рисунок 7. Диаграмма формирования ШИМ-сигнала для режима F2

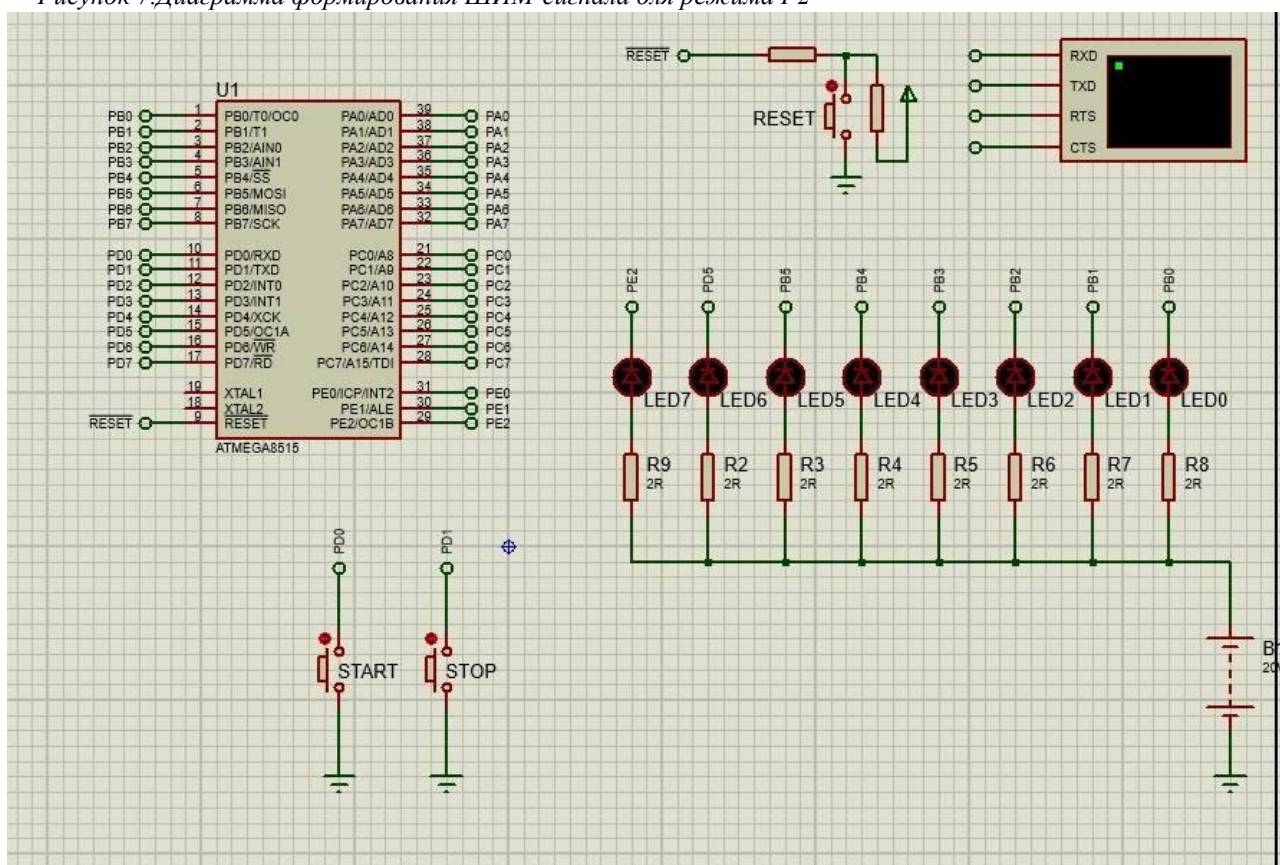


Рисунок 8. Режим ШИМ в Proteus 8 Pro

Режим ШИМ позволяет микроконтроллеру генерировать сигналы с задаваемой частотой и скважностью. Подробное описание работы приведено в методических указаниях. Данная функция может открыть дополнительные возможности при использовании периферийных устройств, в особенности в отсутствие у микроконтроллера аналоговых выходов.

В режиме ШИМ недоступно большинство функций счетчика, это необходимо учитывать при работе.

## Задание 5. Функция захвата

Текст программы:

```
;*****
*****
;.include "8515def.inc" ;файл определений AT90S8515
.include "m8515def.inc" ;файл определений ATmega8515
.def temp = r16 ;временный регистр
.def H_byte = r17 ;для хранения старшего байта
.def L_byte = r18 ;для хранения младшего байта
.equ START = 0 ;0-ой вывод порта PD
.equ SHOW_L = 1 ;1-ый вывод порта
PD .equ SHOW_H = 2 ;2-ой вывод
порта PD ;***Векторы
прерываний .org $000
rjmp INIT ;обработка сброса
.org $003
rjmp CAPT_PRESSED ;обработка внешнего прерывания INT0 -
; нажатие
CAPT ;***Инициализац
ия МК
INIT: ldi temp,low(RAMEND) ;установка out
      SPL,temp ; указателя стека
      ldi temp,high(RAMEND) ; на
      последнюю out SPH,temp ; ячейку ОЗУ
      clr temp ;инициализация выводов
      out DDRD,temp ; порта PD на ввод
      ldi temp,0x07 ;включение 'подтягивающих'
      out PORTD,temp ; резисторов порта PD
      clr temp ;/// для ATmega8515 инициализация
      out DDRE,temp ;/// PE0 (ICP) на ввод
      ldi temp,0x01 ;/// включение 'подтягивающего'
      out PORTE,temp ;/// резистора порта PE0
      ser temp ;инициализация выводов
      out DDRB,temp ; порта PB на вывод
      out PORTB,temp ;выключение светодиодов
      cli ;запрещение прерываний
      clr temp ;отключение от таймера
      out TCCR1A,temp ; выводов
      портов clr temp ;таймер
      out TCCR1B,temp ; остановлен
      ldi temp,0xFF ;запись числа в
      out OCR1AH,temp ; регистр сравнения,
      ldi temp,0xFF ; первым записывается
      out OCR1AL,temp ; старший байт
      clr temp ;обнуление
      out TCNT1H,temp ; содержимого
      out TCNT1L,temp ; счётного регистра
      ldi L_byte,0x00 ; обнуление
      ldi H_byte,0x00 ; регистров вывода
```

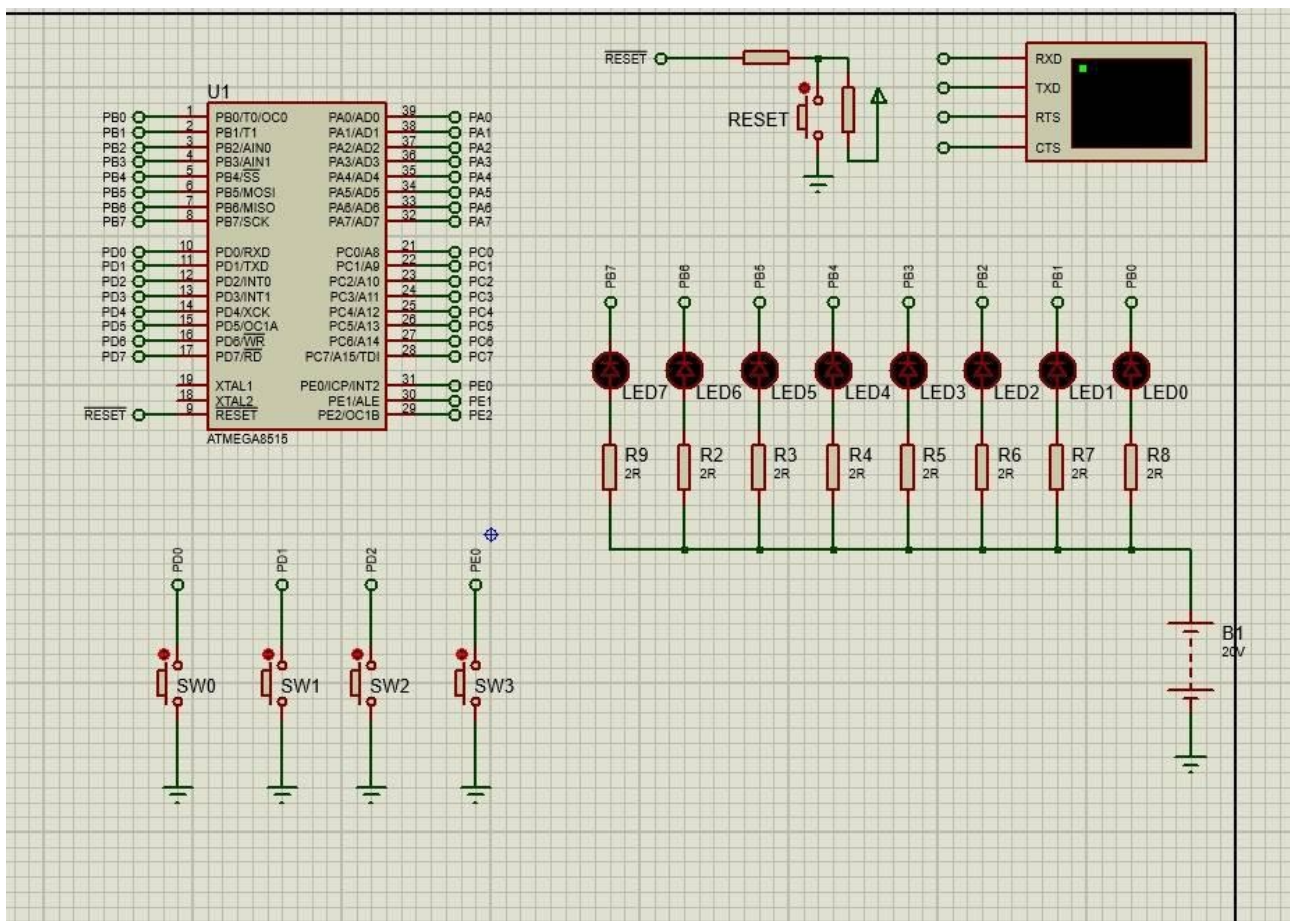


```

sei ;глобальное разрешение прерываний
WAITSTART: sbic PIND,START ;ожидание нажатия
            rjmp WAITSTART ; кнопки START
            ldi temp,0x08 ;разрешение прерывания
            out TIMSK,temp ; по событию "захват" таймера
            ldi temp,0xCD ;запуск таймера с предделителем 1024,
            out TCCR1B,temp ; при совпадении с OCR1A - сброс
WAIT_L: sbic PIND,SHOW_L ;ожидание нажатия
         rjmp WAIT_H ; кнопки SHOW_L - показать младший
         байт out PORTB,L_byte ;вывод на СД
WAIT_H: sbic PIND,SHOW_H ;ожидание нажатия
         rjmp WAIT_L ; кнопки SHOW_H - показать старший
         байт out PORTB,H_byte ;вывод на СД rjmp WAIT_L

;***Обработка прерывания от кнопки CAPT
CAPT_PRESSED:
            in L_byte,ICR1L ;считывание младшего байта
            in H_byte,ICR1H ;считывание старшего байта
            com L_byte ;инвертирование для вывода
            com H_byte ;инвертирование для
            вывода reti

```



Проведём эксперименты и сравним расчётное время по таймеру с измеренным с помощью секундомера:

№	Время на секундомере, с	Показания регистров захвата таймера	Расчётное время, с
1	5,59	0101000001100101 (20581)	$20581 \cdot 1024 / (3.69 \cdot 10^6) \approx 5,71$
2	8,29	0111010101000101 (30021)	$30021 \cdot 1024 / (3.69 \cdot 10^6) \approx 8,33$
3	2,31	0010000100111101 (8509)	$8509 \cdot 1024 / (3.69 \cdot 10^6) \approx 2,36$

## Ответы на контрольные вопросы

1. В каком порядке выполняется запись и чтение байтов в 16 - разрядные регистры таймера T1?

Так как необходимо одновременно изменить (внести данные или считать их) два байта информации, используется дополнительный, скрытый и недоступный пользователю однобайтовый регистр TEMP. При записи данных необходимо сначала записать байт в старший регистр, он будет автоматически помещен в регистр TEMP. После этого происходит запись в младший регистр. Процессор автоматически использует данные из TEMP и осуществит запись одновременно в два регистра — младший и старший. При считывании порядок обратный. Сначала нужно считать младший регистр, при этом данные из старшего автоматически попадут в TEMP, потом считать старший, данные при этом будут получены не непосредственно из старшего регистра, а из TEMP.

2. Как подключены светодиоды к выводам порта микроконтроллера? Сигнал какого уровня (логического 0 или логической 1) включает светодиод?

Светодиоды на рассматриваемой в работе модели платы подключены таким образом, чтобы их включение (появление светового сигнала на них) происходило при поступлении сигнала логического 0. При поступлении сигнала логической 1 светодиод выключен (не горит).

3. Какова длительность сигналов, формируемых программой 4.3? Какими параметрами настройки таймера определяются период и величина относительного сдвига по времени выходных сигналов OC1B и OC1A?

Период и величина сдвига зависят от значений в регистрах сравнения OCR1A для одного сигнала и OCR1B для другого, а так же от частоты работы процессора и значения делителя (три бита регистра TCCR1B).

Длительность сигнала:  $OCR1A \cdot K / F_{ck} = 4096 \cdot 1024 / 3690000 = \sim 1.14 \text{ с}$

4. Какова длительность сигналов, формируемых программой 4.4? Какие значения необходимо занести в регистры сравнения для формирования ШИМ - сигналов скважностью 4 при том же периоде ШИМ – сигналов?

$$F_{tck1} = F_{ck} / 1024 = 3690000 / 1024 = 3604$$

F1:



$$\text{OCR1A} = 0x00FF = 255$$

$$2n/\text{Ftck1} = 510/3604 = 0.14c$$

F2:

$$\text{OCR1B} = 0x02FF = 767$$

$$2n/\text{Ftck1} = 1534/3604 = 0.43c$$

«Скважность 4» означает, что соотношение периода ко времени активного сигнала должно быть 4. Для 10-битного ШИМ-сигнала период равен  $2046/\text{Ftck1} = 0.57c$ . Длительность сигнала должна быть  $0.57/4 = 0.14$ . В регистры сравнения необходимо внести значение  $0x00FF$ , это уже сделано в предложенной в задании программе для F1.

5. Как изменится скважность ШИМ - сигналов, формируемых при работе программы 4.4, если изменить управляющее слово в регистре TCCR1A на  $0xB1$  ( $0xB2$ )?

При  $0xB2$  битность ШИМ-сигнала изменится с 10 на 9. При этом скважность F1 уменьшится вдвое (при старом значении 255 в регистре сравнения в два раза уменьшится период), а скважность F2 станет равной скважности F1 ( $=2$ )? потому что младшие 9 бит в числе  $0x02FF$  (регистр сравнения для F2) равные аналогичным в  $0x00FF$  (регистр сравнения F1).

При  $0xB2$  битность ШИМ-сигнала станет равной 8, при этом скважность обоих сигналов станет равной 1. Рассуждения аналогичны предыдущему пункту.

6. Назовите максимальное время захвата, которое можно зарегистрировать при работе программы 4.5; при рабочей частоте микроконтроллера 1 МГц. В каких приложениях можно применить эту программу?

$$\text{Максимальное значение, которое может находиться в счетчике} — 0xFFFF = 65535$$

$$65535 * 1024 / (1 * 10^6) \approx 67,1c$$

7. В каких разрядах регистра захвата регистрируются секундные интервалы времени при частоте работы микроконтроллера 3,69 МГц с коэффициентом делителя 1024?

Подберем значение счетчика, эквивалентное 1 секунде захвата при заданных параметрах.

$$1 * 3,69 * 10^6 / 1024 = 3603 = 0x0E2E = 00001110\ 00101110$$

Последним задействованным разрядом является 11, но его переключение (при шаге от  $0x07FF$  к  $0x0800$  произойдет раньше, чем будет достигнуто найденное значение, эквивалентное захвату 1 секунды. Следовательно, 1 секунду можно зафиксировать в 12 разряде.

## Вывод

В ходе данной работы были изучены структуры и режимы работы таймеров и их программирование, проанализированы схемы включения таймеров для проведения исследований, проведено программирование задач с таймером. Были проверены режимы счётчика, таймера, широко-импульсной модуляции, сравнения и захвата. Режим счётчика позволяет считать события. Режим таймера даёт возможность формировать временные интервалы. ШИМ режим позволяет генерировать импульсы с заданной скважностью. Функция сравнения даёт возможность сравнивать содержимое счётчика с регистром

сравнения, а функция захвата обеспечивает сохранение состояния таймера или счётчика в регистре захвата.