



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»

НАПРАВЛЕНИЕ ПОДГОТОВКИ «09.03.01 Информатика и вычислительная техника»

ОТЧЕТ по лабораторной работе № 2

Название: «Обработка внешних прерываний в микроконтроллерах AVR»

Дисциплина: «Микропроцессорные системы»

Вариант № 19

Студент ИУ6-62Б
(Группа)

(Подпись, дата)

А.Е.Медведев
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Б.И. Бычков
(И. О. Фамилия)

2022 г.

1 Цель работы:

- изучение системы прерываний микроконтроллеров AVR,
- освоение системы команд микроконтроллеров AVR,
- ознакомление с работой стека при вызове подпрограмм и обработчиков прерываний,
- программирование внешних прерываний.

2 Задания:

2.1 Задание 1

Запустив AVR Studio, проверить работу программы в шаговом режиме. С целью ускорения отладки сократить время задержек до минимума. Проконтролировать работу стека при вызове подпрограмм delay1, delay2.

Листинг 2.1 – Код реализации прерывания на плате ATmega8515

```
.include "m8515def.inc"          ;???? ????????????? ??? ATmega8515
.def temp = r16                  ;???????????? ???????
.equ led = 0                     ;0-? ??? ?????? PB
.equ sw0 = 0                     ;0-? ??? ?????? PA
.equ sw1 = 1                     ;1-? ??? ?????? PA

.org $000
    rjmp INIT                    ;???????????? ???????

INIT:    ldi temp,$5F             ;????????????
        out SPL,temp             ; ????????????? ??????
        ldi temp,$02             ; ?? ?????????????
        out SPH,temp             ; ??????? ???
        ser temp                 ;???????????????????? ???????
        out DDRB,temp            ; ?????? PB ?? ??????
        out PORTB,temp           ;???????????? LED
        clr temp                 ;????????????????????
        out DDRA,temp            ; ?????? PA ?? ?????

        ldi temp,0b00000011      ;???????????? ?????????????????
```

```

        out PORTA,temp        ; ?????????? ????? PA
test_sw0:
        sbic PINA,sw0        ;????????? ??????????
        rjmp test_sw1        ; ?????? sw0
        cbi PORTB, led
        rcall delay1
        sbi PORTB,led
wait_0:
        sbis PINA,sw0
        rjmp wait_0
test_sw1:
        sbic PINA,sw1        ;????????? ??????????
        rjmp test_sw0        ; ?????? sw1
        cbi PORTB,led
        rcall delay2
        sbi PORTB,led
wait_1:
        sbis PINA,sw1
        rjmp wait_1
        rjmp test_sw0
delay1:                                ; ?????????????? 1 ?
        ret
delay2:                                ; ?????????????? 2 ?
        rcall delay1
        rcall delay1
        ret

```

При вызове подпрограммы или функции в тек помещается адрес возврата. По условию задания начало стека установлена в конец секции данных. Работы стека продемонстрирована на рисунках 2.1 — 2.2

40:	rjmp test_sw0			
+0000001A:	CFF1	RJMP	PC-0x000E	Relative jump
@0000001B:	delay1			
42:	ret			
+0000001B:	9508	RET		Subroutine return
@0000001C:	delay2			
44:	rcall delay1			
+0000001C:	DFFE	RCALL	PC-0x0001	Relative call subroutine
45:	rcall delay1			
+0000001D:	DFFD	RCALL	PC-0x0002	Relative call subroutine
46:	ret			
+0000001E:	9508	RET		Subroutine return
----- No Source -----				

Рисунок 2.1 – Дизассемблер программы

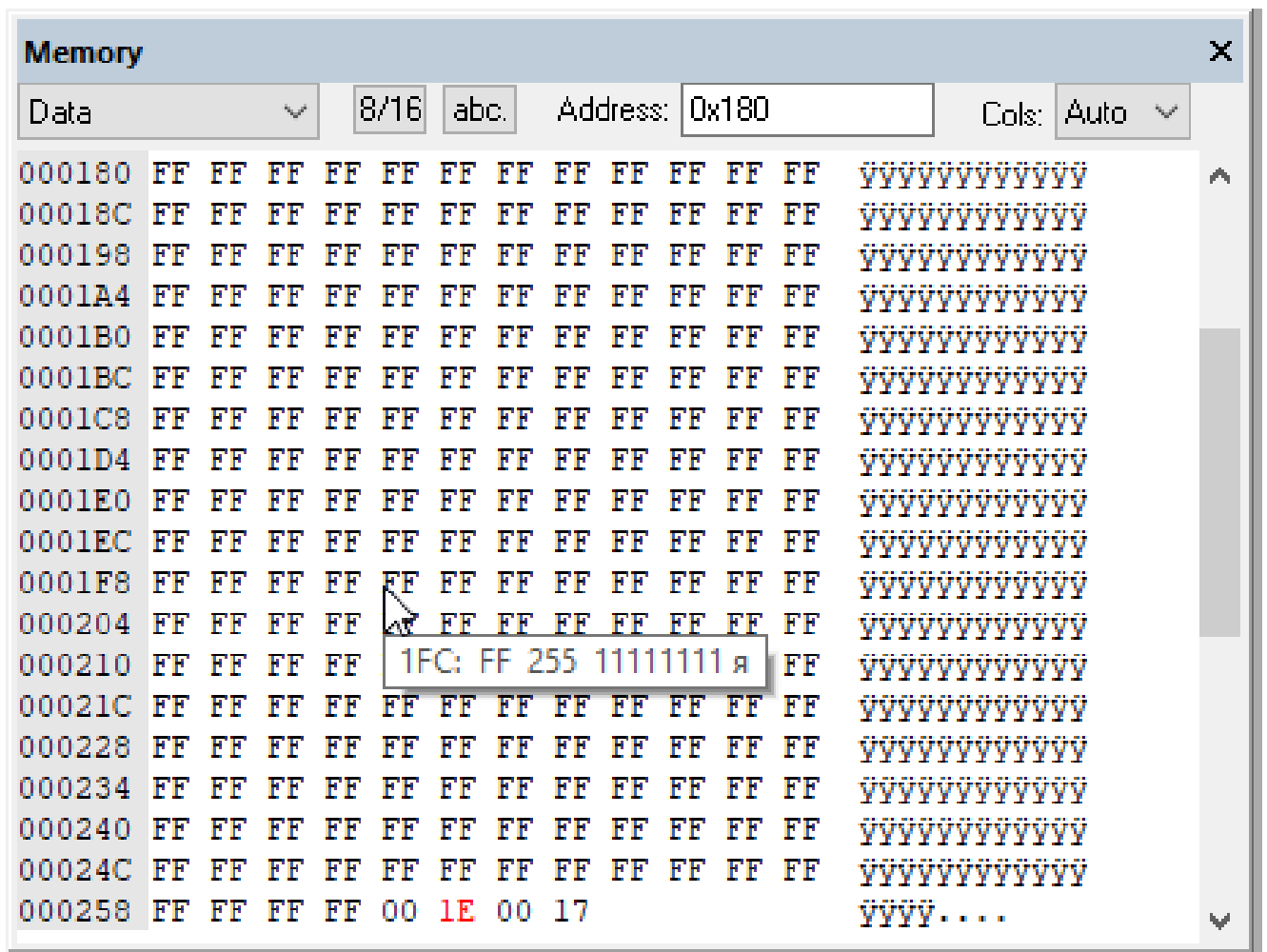


Рисунок 2.2 – Состояние стека при вызове подпрограммы или функции

2.2 Задание 2

Вносим изменения и дополнения в исходный текст программы 2.1, касающиеся обработки прерываний. На этапе инициализации указываются область

стека для сохранения адресов возврата, при необходимости адреса векторов прерываний и сами векторы, маска прерываний, общее разрешение прерываний. Завершаем инициализацию переводом процессора в фоновый режим ожидания:

Листинг 2.2 – Код реализации прерывания на плате ATmega8515

```
.include "m8515def.inc"      ;файл определений для ATmega8515
.def temp = r16              ;временный регистр
.equ led = 0                 ;0-о бит порта PB
.equ sw0 = 2                 ;2-й бит порта PD
.equ sw1 = 3                 ;3-й бит порта PD
.org $000
    ;***Таблица векторов прерываний, начиная с адреса $000***
    rjmp INIT                ;обработка сброса
    rjmp led_on1             ;на обработку запроса INT0
rjmp led_on2                 ;на обработку запроса INT1
;***Инициализация SP, портов, регистра маски***
INIT:    ldi temp,$5F        ;установка
        out SPL,temp        ; указателя стека
        ldi temp,$02        ; на последнюю
        out SPH,temp        ; ячейку ОЗУ
        ser temp            ;инициализация выводов
        out DDRB,temp       ; порта PB на вывод
        out PORTB,temp      ;погасить СД
        clr temp            ;инициализация
        out DDRD,temp       ; порта PD на ввод
        ldi temp,0b00001100 ;включение подтягивающих
        out PORTD,temp      ; резисторов порта PD
        ldi temp,((1<<INT0)|(1<<INT1));разрешение прерываний
        out GICR,temp       ; в 6,7 битах регистра маски GICR
        ldi temp,0          ;обработка прерываний
        out MCUCR,temp      ; по низкому уровню
        sei                 ;глобальное разрешение прерываний
loop:
        nop                 ;режим ожиданий
        rjmp loop
led_on1:
        cbi PORTB,led
        rcall delay1
        sbi PORTB,led
```

```

wait_0:      sbis pind,sw0
             rjmp wait_0
             reti
led_on2:
             cbi PORTB,led
             rcall delay2
             sbi PORTB,led
wait_1:
             sbis pind,sw1
             rjmp wait_1
             reti
delay1:
             nop
;для подпрограммы задержки 1 с
             ret
delay2:      ;подпрограмма задержки 2 с
             rcall delay1
             rcall delay1
             ret

```

Состояние регистров и стека продемонстрировано на рисунках 2.3 — 2.5. Результат работы программы был отлажен в среде протейс, как показано на рисунке 2.6

000228	FF FF FF FF FF FF FF FF FF FF FF FF	----- ÿÿÿÿÿÿÿÿÿÿÿÿ
000234	FF FF FF FF FF FF FF FF FF FF FF FF	ÿÿÿÿÿÿÿÿÿÿÿÿ
000240	FF FF FF FF FF FF FF FF FF FF FF FF	ÿÿÿÿÿÿÿÿÿÿÿÿ
00024C	FF FF FF FF FF FF FF FF FF FF FF FF	ÿÿÿÿÿÿÿÿÿÿÿÿ
000258	FF FF 00 24 00 1D 00 14	ÿÿ.\$....

Рисунок 2.3 – Состояние стека при вызове подпрограммы или функции

8:	rjmp INIT		;обработка сброса
+00000000:	C002	RJMP	PC+0x0003 Relative jump
9:	rjmp led_on1		;на обработку запроса INTO
+00000001:	C013	RJMP	PC+0x0014 Relative jump
10:	rjmp led_on2		;на обработку запроса INT1
+00000002:	C018	RJMP	PC+0x0019 Relative jump
@00000003:	INIT		
12:	INIT: ldi temp,\$5F		;установка
+00000003:	E50F	LDI	R16,0x5F Load immediate
13:	out SPL,temp		; указателя стека
+00000004:	BF0D	OUT	0x3D,R16 Out to I/O location
14:	ldi temp,\$02		; на последнюю
+00000005:	E002	LDI	R16,0x02 Load immediate
15:	out SPH,temp		; ячейку 039
+00000006:	BF0E	OUT	0x3E,R16 Out to I/O location
16:	ser temp		;инициализация выводов
+00000007:	EF0F	SER	R16 Set Register
17:	out DDRB,temp		; норма PB на вывод
+00000008:	BB07	OUT	0x17,R16 Out to I/O location
18:	out PORTB,temp		;подаем СД
+00000009:	BB08	OUT	0x18,R16 Out to I/O location
19:	clr temp		;инициализация
+0000000A:	2700	CLR	R16 Clear Register
20:	out DDRD,temp		; норма PD на вывод
+0000000B:	BB01	OUT	0x11,R16 Out to I/O location
21:	ldi temp,0b00001100		;включение `подтягивающих`
+0000000C:	E00C	LDI	R16,0x0C Load immediate
22:	out PORTD,temp		; резисторов норма PD
+0000000D:	BB02	OUT	0x12,R16 Out to I/O location
23:	ldi temp,((1<<INT0) (1<<INT1));		разрешение прерываний
+0000000E:	EC00	LDI	R16,0xC0 Load immediate
24:	out GICR,temp		; в 6,7 битах регистра маски GICR
+0000000F:	BF0B	OUT	0x3B,R16 Out to I/O location
25:	ldi temp,0		;обработка прерываний
+00000010:	E000	LDI	R16,0x00 Load immediate
26:	out MCUCR,temp		; по низкому уровню

Рисунок 2.4 – Дизассемблер программы

2.3 Задание 3

Подготовим программу соответствующую заданному алгоритму работы. При инициализации помимо общих директив устанавливаем исходный управляющий код в регистре индикации, нулевой разряд которого инициирует зажигание светодиода, настраиваем на вывод порт микроконтроллера и указатель стека. В цикле алгоритма на каждой итерации выполняем вывод в порт микроконтроллера управляющего слова, временную задержку, затем циклический сдвиг влево управляющего слова. Программные коды основной программы и обработки прерываний дополняем операциями, связанными с обработкой внешних запросов от кнопок, учитывая заданные входы прерываний (INT1 - линия порта PD3, адрес вектора прерываний \$002, маска – бит 7 в регистре GICR; INT2 - линия порта PE0, адрес вектора прерываний \$00D, маска – бит 5 в регистре GICR).

Листинг 2.3 – Код реализации прерывания на плате ATmega8515

```
;?????????: SW0-PD0 , SW2-PD2 , LED-PB
;*****
;.include "8515def.inc"
.include "m8515def.inc"
.def temp = r16
.def reg_led = r20
.def key = r21
.equ START = 0
.org $000

rjmp INIT
.org $002
rjmp START_PRESSED ;????????? ?????????? ?????????? INT1(START)
.org $00D
rjmp STOP_PRESSED ; ?????????? ?????????? ?????????? INT2(STOP)

INIT:
    ldi temp,$5F
    out SPL,temp
    ldi temp,$02
    out SPH,temp
```

```

ldi reg_led,0x3F
sec ;C=1
clt ;T=0
clr key

ser temp
out DDRB,temp
out PORTB,temp
clr temp
out DDRD,temp
ldi temp,0x08
out PORTD, temp
clr temp
out DDRE,temp
ldi temp,0x1
out PORTE,temp
ldi temp,(1<<INT2)| (1<<INT1)
out GICR,temp
ldi temp,0x00
out MCUCR,temp
out MCUCSR,temp
sei
WAIT:
sbrs key,0
rjmp WAIT
out PORTB,reg_led
rcall DELAY

brts LEFT

;sbrs reg_led,1
mov r27, reg_led
andi r27, 0x01
brne R_OFFSET
set
ldi reg_led,0x03
clc
rjmp WAIT

```

```

        ;end if

R_OFFSET:
    ror reg_led
    rjmp WAIT

LEFT:
    ;sbrc reg_led,6
    mov r27, reg_led
    andi r27, 0x3F
    brne L_OFFSET
    clt
    ldi reg_led,0x3F
    sec
    rjmp WAIT
    ;end if
L_OFFSET:
    rol reg_led
    rjmp WAIT

STOP_PRESSED:
    ldi key,0
    reti

START_PRESSED:
    ldi key,1
    reti

DELAY: ldi r17,11
d1: ldi r18,255
d2: ldi r19,236
d3: dec r19
    brne d3
    dec r18
    brne d2
    dec r17
    brne d1
    ret

```

На рисунке 2.7 продемонстрировано состояние программы в процессе отлад-

ки. На данном рисунке можно отпределить работу стека, изменения состояния регистров и флагов.

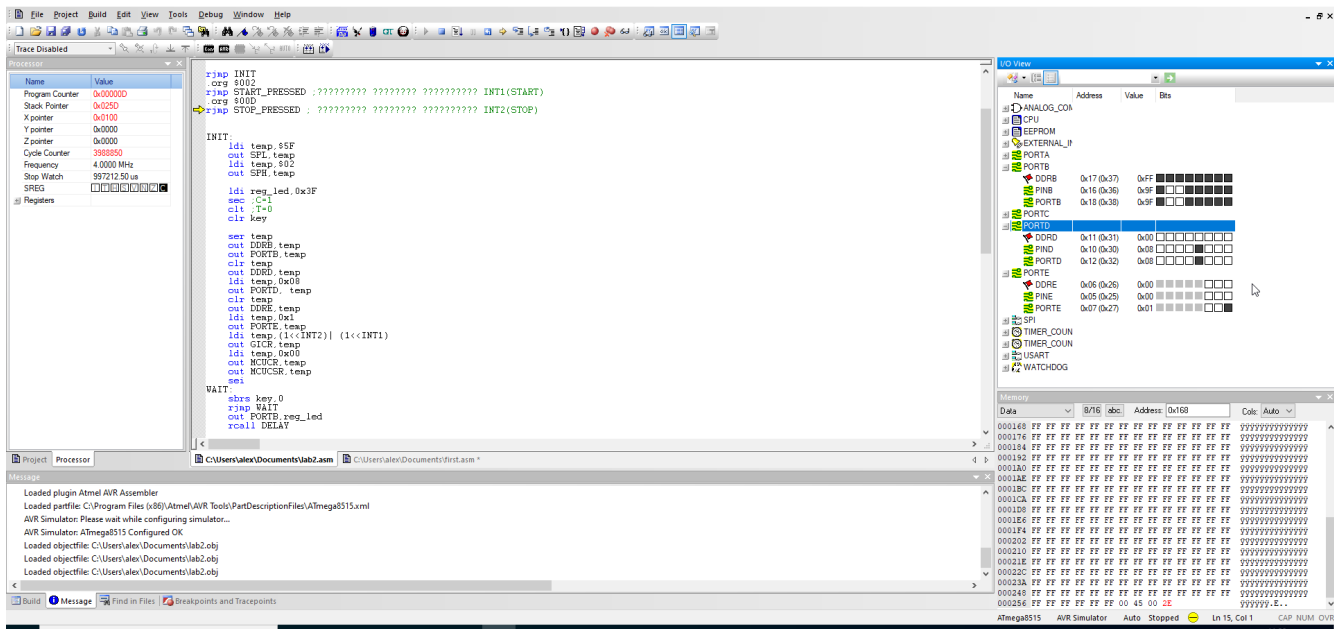


Рисунок 2.7 – Состояние программы в процессе отладки

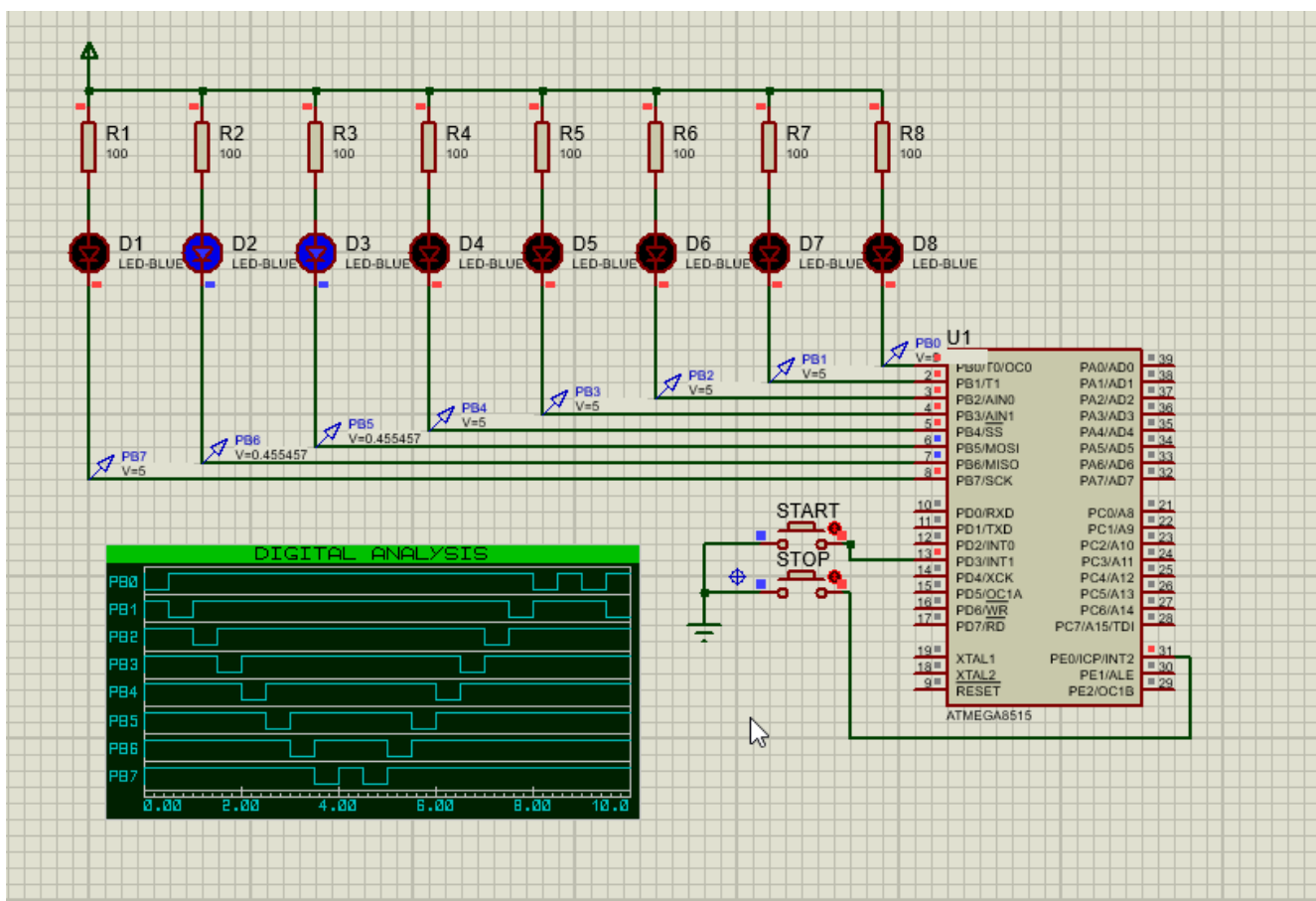


Рисунок 2.8 – Схема прерываний в протеусе

2.4 Задание 4

Запустив программу Proteus ISIS, собрать проект, включающий микроконтроллер, 2 кнопки и 8 светодиодов. Объединим два запроса прерываний от кнопок с помощью диодной сборки для передачи на вход прерывания микроконтроллера INT2. Обработка прерывания в этом случае начинается с программной идентификации источника запроса прерывания.

Листинг 2.4 – Код реализации прерывания на плате ATmega8515

```
.include "m8515def.inc"
.def temp = r16
.def reg_led = r20
.def key = r21
.equ START = 0
.org $000
rjmp INIT
.org $00D
rjmp EO_PRESSED ; ?????????? ?????????? ???????????? INT2()
INIT:
    ldi temp,$5F
    out SPL,temp
    ldi temp,$02
    out SPH,temp

    ldi reg_led,0x3F
    sec ;C=1
    clt ;T=0
    clr key

    ser temp
    out DDRB,temp
    out PORTB,temp
    clr temp
    out DDRA,temp
    ldi temp,0x03
    out PORTA,temp
    clr temp
    out DDRE,temp
    ldi temp,0x1
```

```

    out PORTE,temp
    ldi temp,(1<<INT2)
    out GICR,temp
    ldi temp,0x00
    out MCUCR,temp
    out MCUCSR,temp
    sei
WAIT:
    sbrs key,0
    rjmp WAIT
    out PORTB,reg_led
    rcall DELAY
    brts LEFT

    mov r27, reg_led
    andi r27, 0x01
    brne R_OFFSET
    set
    ldi reg_led,0x03
    clc
    rjmp WAIT
R_OFFSET:
    ror reg_led
    rjmp WAIT
LEFT:
    mov r27, reg_led
    andi r27, 0x3F
    brne L_OFFSET
    clt
    ldi reg_led,0x3F
    sec
    rjmp WAIT
L_OFFSET:
    rol reg_led
    rjmp WAIT
EO_PRESSED:
    sbis PINA, 0
    ldi key,0
    sbis PINA, 1

```

```

        ldi key,1
        reti
DELAY:  ldi r17,11
d1:     ldi r18,255
d2:     ldi r19,236
d3:     dec r19
        brne d3
        dec r18
        brne d2
        dec r17
        brne d1
        ret

```

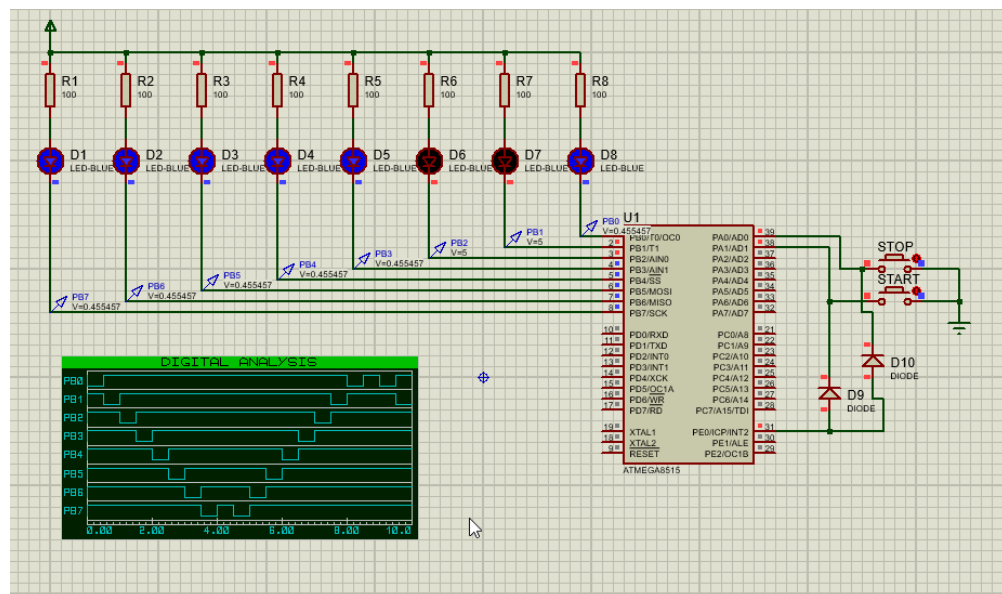


Рисунок 2.9 – Схема прерываний в протеусе

3 Вывод:

В результате выполнения лабораторной работы были изучены системы прерываний микроконтроллеров AVR, а именно прерывания INT0, INT1 и INT2, освоены системы команд микроконтроллеров AVR, был получен опыт работы со стеком при вызове подпрограмм и обработчиков прерываний, а именно понимание работы стека при записи адресов в пошаговом режиме, и с программированием внешних прерываний. Также, были получены навыки работы со средой моделирования ISIS Proteus, сборка схем для моделирования и построение осциллограммы