

Лекция 8 (17.04.20)

# Язык Ассемблера для программирования микроконтроллеров AVR

# Основные команды языка Ассемблера для программирования микроконтроллеров AVR

Команда	Описание	Пример
<i>SER</i>	Установка бит регистра	<i>SER R20 // R20=0xFF</i>
<i>CLR</i>	Сброс бит регистра	<i>CLR R20 // R20=0x00</i>
<i>INC</i>	Увеличение значения регистра на единицу	<i>INC R20 //R20=R20+1</i>
<i>DEC</i>	Уменьшение значения регистра на единицу	<i>DEC R20 //R20=R20-1</i>
<i>AND</i>	Логическое умножение (конъюнкция)	<i>AND R20,R21//R20=R20&amp;R21</i>
<i>OR</i>	Логическое сложение (дизъюнкция)	<i>OR R20,R21 //R20=R20 R21</i>

Команда	Описание	Пример
<i>EOR</i>	Логическое исключаяющее «ИЛИ»	<i>EOR R20, R21 // R20 = R20 <math>\oplus</math> R21</i>
<i>COM</i>	Побитовая инверсия	<i>COM R20 // R20 = <math>\sim</math>R20</i>
<i>SBI</i>	Установка бита в регистре порта	<i>SBI PORTB, 4 // установлен // четвертый бит в // регистре PORTB</i>
<i>CBI</i>	Очистка бита в регистре порта	<i>CBI PORTB, 4 // очищен // четвертый бит в // регистре PORTB</i>
<i>MOV</i>	Копирование значения одного регистра в другой	<i>MOV R20, R21 // R20 = R21</i>
<i>LDI</i>	Загрузка константы в регистр	<i>LDI R20, 0x0F // R20 = 0x0F</i>
<i>IN</i>	Чтение регистра порта	<i>IN R20, PIND // R20 = PIND</i>

Команда	Описание	Пример
<i>OUT</i>	Запись значения регистра в регистр порта	<i>OUT PORTB, R20 // PORTB=R20</i>
<i>JMP</i>	Переход	<i>//бесконечный цикл</i> <i>START:</i> <i>JMP START</i>
<i>RJMP</i>	Относительный переход	<i>//код программы расположен с</i> <i>//адреса 0x0000</i> <i>.org 0x0000 rjmp PROGR</i> <i>...</i> <i>PROGR:</i>
<i>RCALL</i>	Относительный вызов подпрограммы	<i>// вызов подпрограммы</i> <i>RCALL delay</i> <i>...</i> <i>// задержка</i> <i>delay:</i> <i>LDI R19,250</i> <i>M3: LDI R20, 0</i> <i>      M2: LDI R21,0</i> <i>      M1: INC R21</i> <i>      BRNE M1 ;</i> <i>      INC R20</i> <i>      BRNE M2</i> <i>INC R19</i> <i>BRNE M3</i> <i>RET // возврат из</i> <i>      // подпрограммы delay</i>
<i>RET</i>	Возврат из подпрограммы	
<i>BRNE</i>	Перейти по метке, если результат последней операции не равен нулю	

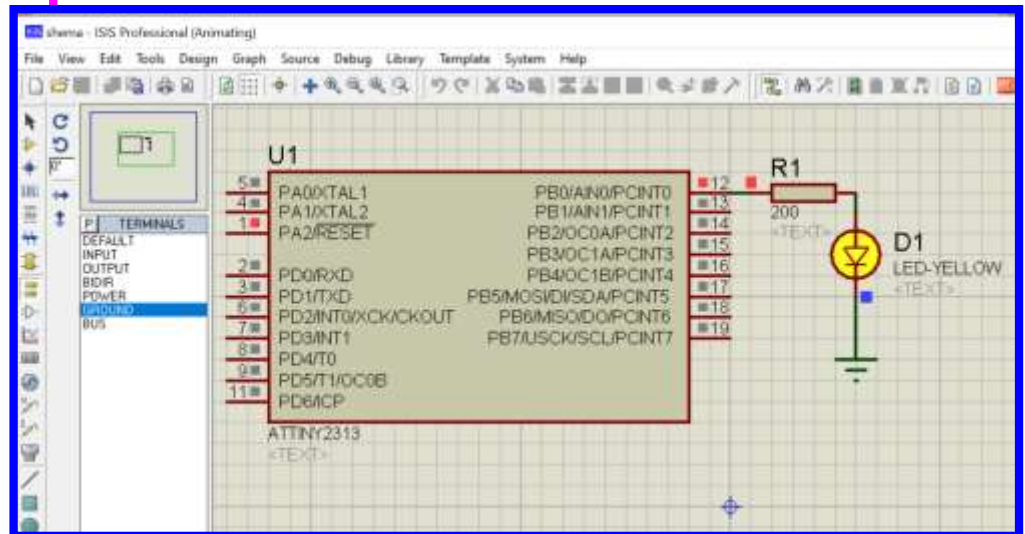
Команда	Описание	Пример
<i>BREQ</i>	Перейти по метке, если результат последней операции равен нулю	<i>MM: ldi R20,0b00000001</i> <i>out PORTB,R20</i>  ... <i>DEC ii</i> <i>BREQ MMM</i>
<i>CLI</i>	Глобальный запрет прерываний	<i>CLI</i>
<i>SEI</i>	Глобальное разрешение прерываний	<i>SEI</i>
<i>LSL</i>	Логический сдвиг влево значения в регистре	<i>LSL R20 // R20 = R20&lt;&lt;1</i>
<i>LSR</i>	Логический сдвиг вправо значения в регистре	<i>LSR R20 // R20 = R20&gt;&gt;1</i>
<i>SBRC</i>	Пропустить следующую команду, если указанный бит в регистре очищен	<i>SBRC R20, 5</i> <i>INC R20</i>

Команда	Описание	Пример
<i>RETI</i>	Возврат из прерывания	<pre> .org 0x0000 rjmp PROGR // ВП INT0 расположено // по адресу 0x0001 rjmp INT_0 // ВП INT1 расположено // по адресу 0x0002 rjmp INT_1 PROGR: ldi R21,4  ... INT_0: LDI R20,1 CPSE R21,R20 DEC R21 RETI INT_1: SBRS R20,3 INC R20 RETI // возврат из прерывания </pre>
<i>CPSE</i>	Сравнение значений регистров. Если равны, то пропускаем следующую команду	
<i>SBRS</i>	Пропустить следующую команду, если указанный бит в регистре установлен	

# Примеры программ для микроконтроллера ATtiny2313

```
1 .include "tn2313def.inc"
2 .org 0x0000
3
4 // установка бита DDB0 регистра DDRB
5 SBI DDRB,0
6
7 // установка бита PORTB0 регистра PORTB
8 SBI PORTB,0
9
10 START:
11     // установка бита PORTB0 регистра PORTB
12     SBI PORTB,0
13     RCALL delay
14
15     // сброс бита PORTB0 регистра PORTB
16     CBI PORTB,0
17     RCALL delay
18
19     RJMP START
20
21 // задержка
22 delay:
23 LDI R19,250
24 M3: LDI R20, 0
25     M2: LDI R21,0
26         M1: INC R21
27         BRNE M1
28         INC R20
29         BRNE M2
30 INC R19
31 BRNE M3
32 RET
```

## Программа «Мигающий светодиод»



```

1  .include "tn2313def.inc"
2  .org 0x0000
3
4  // настройка выводов PB0-PB7 на вывод данных
5  SER R16 ;
6  out DDRB, R16
7
8  // включение светодиода LED0
9  ldi R16,0b00000001
10 out PORTB, R16
11
12 START:
13     // пропустить команду если 7-ой бит регистра PORTB очищен
14     sbic PORTB,7
15     ldi R16,0b00000001
16
17     // переключение светодиодов
18     LSL R16
19     OUT PORTB, R16
20
21     // задержка
22     rcall delay
23
24     rjmp START

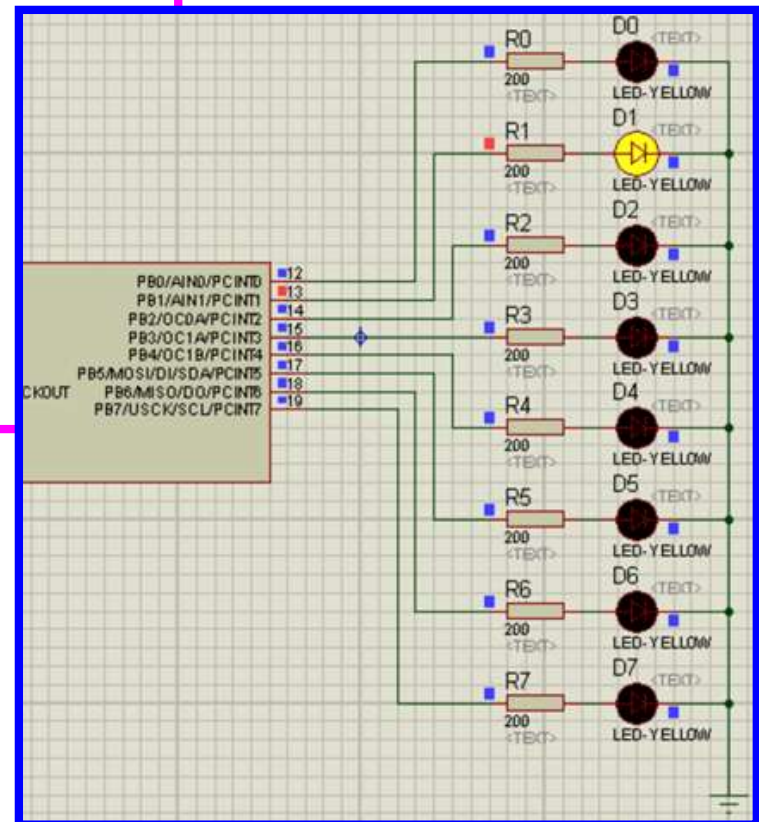
```

```

25
26 // задержка
27 delay:
28 LDI R19,250
29     M3: LDI R20, 0
30         M2: LDI R21,0
31             M1: INC R21
32                 BRNE M1
33                     INC R20
34                         BRNE M2
35                             INC R19
36                                 BRNE M3
37     ret

```

## Программа «Светодиодная дорожка»





# Таблица векторов прерывай микроконтроллера ATtiny2313

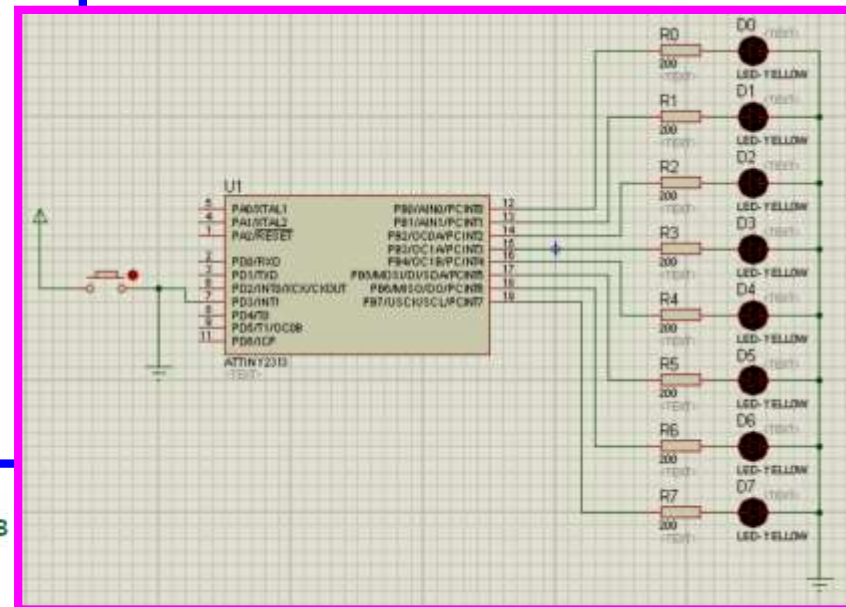
Vector No.	Program Address	Source	Interrupt Definition
1	0x0000	RESET	External Pin, Power-on Reset, Brown-out Reset, and Watchdog Reset
2	0x0001	INT0	External Interrupt Request 0
3	0x0002	INT1	External Interrupt Request 1
4	0x0003	TIMER1 CAPT	Timer/Counter1 Capture Event
5	0x0004	TIMER1 COMPA	Timer/Counter1 Compare Match A
6	0x0005	TIMER1 OVF	Timer/Counter1 Overflow
7	0x0006	TIMER0 OVF	Timer/Counter0 Overflow
8	0x0007	USART0, RX	USART0, Rx Complete
9	0x0008	USART0, UDRE	USART0 Data Register Empty
10	0x0009	USART0, TX	USART0, Tx Complete
11	0x000A	ANALOG COMP	Analog Comparator
12	0x000B	PCINT	Pin Change Interrupt
13	0x000C	TIMER1 COMPB	Timer/Counter1 Compare Match B
14	0x000D	TIMER0 COMPA	Timer/Counter0 Compare Match A
15	0x000E	TIMER0 COMPB	Timer/Counter0 Compare Match B
16	0x000F	USI START	USI Start Condition
17	0x0010	USI OVERFLOW	USI Overflow
18	0x0011	EE READY	EEPROM Ready
19	0x0012	WDT OVERFLOW	Watchdog Timer Overflow

```

1  .include "tn2313def.inc"
2
3  .cseg
4  // код программы расположен с адреса 0x0000
5  .org 0x0000 rjmp PROGR
6
7  // ВП INT1 расположено по адресу 0x0002
8  .org 0x0002 rjmp INT_1
9
10 PROGR:
11
12 // установка бит в регистре DDRB
13 SER R16
14 out DDRB, R16
15
16 // настройка вызова прерывания INT1 (передний фронт)
17 ldi R16,0b00001100
18 out MCUCR, R16
19
20 //разрешение прерывания INT1
21 ldi R16,0b10000000;
22 out GIMSK,R16
23
24 // глобальное разрешение прерываний
25 SEI
26
27 // бесконечный цикл
28 START:
29 rjmp START
30
31 INT_1:
32     //переключение светодиодов
33     in R17, PORTB
34     com R17
35     out PORTB, R17
36     RETI

```

**Пример** программы на языке Ассемблера, демонстрирующей работу ВП INT1: включение/выключение светодиодов (выводы микроконтроллера PB0-PB7) нажатием кнопки (вывод микроконтроллера INT1).



```

1  .cseg
2  // код программы расположен с адреса 0x0000
3  .org 0 rjmp PROGR
4  // ВП INT0 расположено по адресу 0x0001
5  rjmp INT_0
6  // ВП INT1 расположено по адресу 0x0002
7  rjmp INT_1
8
9  .set s=0;
10
11 PROGR:
12 // настройка выводов PB2 и PB3 на вывод данных
13 sbi DDRB,2
14 sbi DDRB,3
15
16 //разрешение прерываний INT0 и INT1
17 ldi R16,0b11000000
18 out GIMSK,R16
19
20 //настройка прерываний INT0 и INT1 (передний фронт)
21 ldi R16,0b00001111
22 out MCUCR, R16
23
24 // глобальное разрешение прерываний
25 SEI
26
27 // бесконечный цикл
28 START:
29 rjmp START

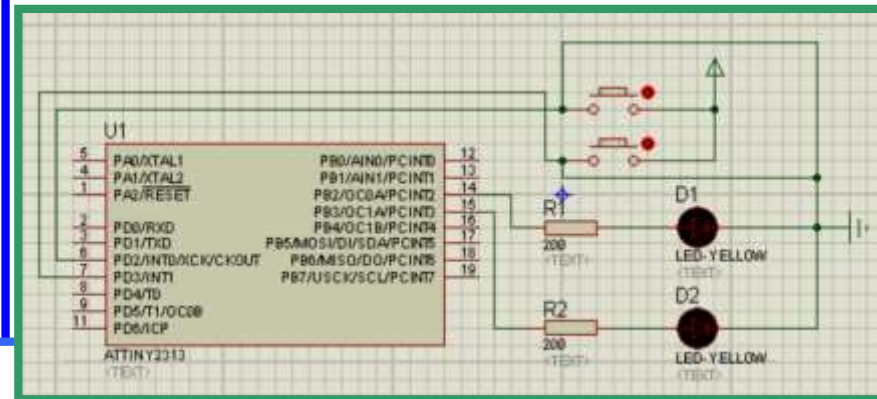
```

```

30
31 INT_0:
32     // включение/выключение светодиода
33     in R17, PORTB
34     ldi R16, 0b00000100
35     eor R17, r16 // исключающее ИЛИ
36     out PORTB, R17
37     RETI
38
39 INT_1:
40     // включение/выключение светодиода
41     in R17, PORTB
42     ldi R16, 0b00001000
43     eor R17, r16 // исключающее ИЛИ
44     out PORTB, R17
45     RETI

```

**Пример** программы на языке Ассемблера, демонстрирующей работу ВП INT0 и INT1: включение/выключение светодиода (вывод микроконтроллера PB2) нажатием кнопки (вывод микроконтроллера INT0) и включение/выключение светодиода (вывод микроконтроллера PB3) нажатием кнопки (вывод микроконтроллера INT1) с сохранением состояния светодиодов.



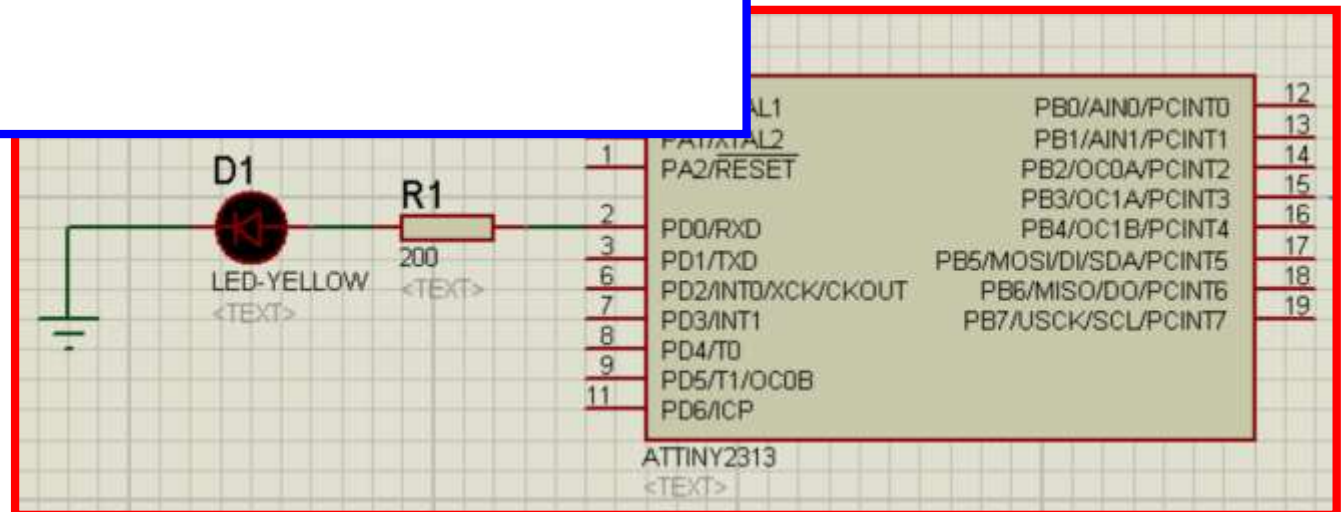


```

1  .include "tn2313def.inc"
2
3  // настройка вывода PD0 на вывод данных
4  SBI DDRD,0
5
6  // запуск T/C T0 (предварительный делитель 1024)
7  LDI R20, 5
8  OUT TCCR0B, R20
9
10 START:
11 // считывание состояния CP TCNT0 в PCH
12 in R20,TCNT0
13
14 // пропустить установку бита PORTD0 регистра PORTD
15 // (вкл. светодиода) если 7ой бит в регистре TCNT0 очищен
16 SBRC R20,7
17 SBI PORTD,0
18
19 // пропустить сброс бита PORTD0 регистра PORTD (выкл.
20 // светодиода) если 7ой бит в регистре TCNT0 установлен
21 SBRS R20,7
22 CBI PORTD,0
23
24 RJMP START

```

Пример программы, демонстрирующей работу T/C T0 в режиме «Normal»: переключение светодиода (вывод микроконтроллера PD0) через 128 отсчетов счетчика

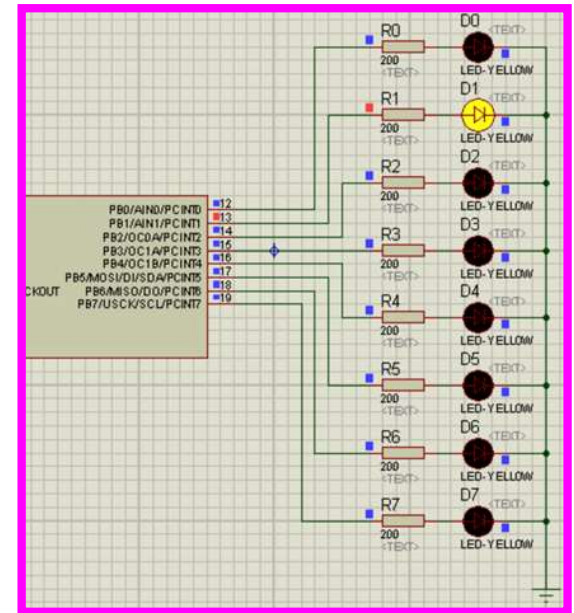


```

1 .include "tn2313def.inc"
2 //регистр общего назначения r16 получает имя r
3 .def r=r16
4
5 .cseg
6 //код программы расположен с адреса 0x0000
7 .org 0x0000 rjmp PROGR
8
9 // прерывание по совп. в канале A расположено по адресу 0x000D
10 .org 0x000D rjmp T0_COMPA
11
12 PROGR:
13     // установка бит в регистре DDRB
14     SER r20
15     OUT DDRB, r20
16
17     // установка PORTB0 в регистре PORTB (вкл. светодиода)
18     LDI R, 0b00000001
19     OUT PORTB, R
20
21     // разрешение прерывания по совпадению в канале A
22     LDI R20, 0b00000001
23     OUT TIMSK, R20
24
25     //настройка режима CTC с предельным
26     // значением СЧ TCNT0 = 200
27     LDI R20, 200
28     OUT OCR0A, R20
29     LDI R20, 0b00000010
30     OUT TCCR0A, R20

```

Пример программы на языке Ассемблера, демонстрирующей работу Т/С Т0 в режиме CTC и прерывания по совпадению в канале А: светодиодная дорожка из светодиодов (выводы микроконтроллера PB0-PB7); переключение светодиодов через 200 отсчетов счетчика.



```

31
32 // запуск Т/С Т0 (предварительный делитель 1024)
33 LDI R20, 0b00000101
34 OUT TCCR0B, R20
35
36 // глобальное разрешение прерываний
37 SEI
38
39 START:
40 RJMP START
41
42 T0_COMPA:
43     //логический побитовый сдвиг влево
44     LSL R16
45
46     // пропустить запись в POH R16 значения 0b1
47     // если сброшен бит PORTB7 регистра PORTB
48     SBIC PORTB, 7
49     ldi R16, 0b00000001
50
51     //запись значения из POH R16 в PORTB
52     OUT PORTB, R16
53 RETI

```