**1 ЛАБОРАТОРНАЯ РАБОТА №1**

**ИЗУЧЕНИЕ ОТЛАДОЧНОГО СТЕНДА EV8031/AVR, СИСТЕМЫ КОМАНД ОДНОКРИСТАЛЬНОГО МИКРОКОНТРОЛЛЕРА ATmega8515. ИНДИКАТОРЫ НА ПОЛУПРОВОДНИКОВЫХ СВЕТОДИОДАХ**

**Цель работы**

Ознакомиться с функциональными возможностями и внутренней структурой отладочного стенда EV8031/AVR. Изучить внутреннюю архитектуру и структуру микроконтроллера ATmega8515, его систему команд. Изучить функциональные возможности IDE AVR Studio 4, научиться создавать и компилировать, выполнять пошаговую трассировку программ, написанных на языке С или ассемблер, программировать указанный микроконтроллер. Научиться отображать информацию на полупроводниковом светодиодном индикаторе.

**1.1 Краткие теоретические сведения**

Простейшими приборами отображения информации в цифровых устройствах являются светодиоды.

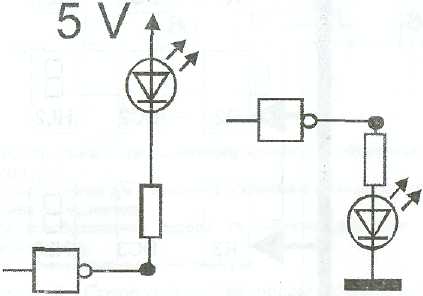


Рисунок 1.1 – Схема подключения светоизлучающего диода

Полупроводниковый светодиод – это излучающий полупроводниковый прибор с одним или несколькими электрическими переходами, предназначенный для непосредственного преобразования электрической энергии в энергию некогерентного светового излучения. Действие полупроводникового светодиода базируется на спонтанном излучении, образующемся вследствие рекомбинации некомпенсированных носителей заряда, которые инжектируют через p-n переход.

В полупроводниковых светодиодах используется свойство р-n перехода излучать свет в видимой части спектра при протекании через него прямого тока (Iпр=5-20мА, Uпр=2-3B). Схема реализации индикатора на полупроводниковых светодиодах представлена выше на рисунке 1.1.

В учебно-отладочном стенде EV8031/AVR доступ к светодиодному индикатору (линейке светодиодов HL1 – HL8) осуществляется как к ячейке внешней памяти, которая расположена по адресу 0xA006 (A006h). Каждый из светодиодов зажигается записью логической единицы в соответствующий разряд.

**1.2 Порядок выполнения работы**

1.2.1 Запустите IDE AVR Studio 4.

1.2.2 Активируйте создание нового проекта посредством выбора пункта New Project из пункта Project главного меню AVR Studio 4.

1.2.3 Выберите тип создаваемого проекта: проект будет создаваться на языке ассемблер (пункт Atmel AVR Assembler) или на языке С (пункт AVR GCC).

1.2.4 Ниже, в поле Location, укажите путь по которому будет располагаться проект. Для удобства дальнейших манипуляций с проектом необходимо сохранить его в отдельном каталоге (настоятельно рекомендуется D:\AVR\_projects).

1.2.5 Указать имя проекта (на английском языке) в поле Project name.

1.2.6 Принудительно (если это не сделано самой AVR Studio 4) установить флажок Create initial file. Желательно также установить флажок Create folder, что позволит создать отдельный каталог (по пути, указанном в поле Location) для сохранения всех файлов проекта внутри него. После указанных действий нажать кнопку Next >>.

1.2.7 В появившемся окне Debug platform выбрать пункт AVR Simulator. В окне Device выбрать пункт ATmega8515. Выбор любых отличающихся от приведенных выше пунктов в указанных окнах приведёт к дальнейшим ошибкам и невозможности выполнения лабораторной работы!

1.2.8 Нажать кнопку Finish для завершения создания проекта.

1.2.9 В появившемся окне написать программу на языке С или ассемблер с учётом варианта задания, который указан в таблице 1.1.

Таблица 1.1 – Варианты заданий

|  |  |
| --- | --- |
| № | Описание задания |
| 10 | Реализовать алгоритм индикации №8 (смотри таблицу 1.9 ниже). Длительность индикации одного состояния (Ti) – 4 с. |

Таблица 1.2 – Алгоритм индикации №8

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Т1** | **Т2** | **Т3** | **Т4** | **Т5** | **Т6** | **Т7** | **Т8** | **Т9** | **Т10** | **Т11** | **Т12** | **Т13** | **Т14** | **Т15** |
| **VD1** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **VD2** | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| **VD3** | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| **VD4** | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| **VD5** | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| **VD6** | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| **VD7** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **VD8** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

1.2.10 Произвести компиляцию проекта посредством выбора пункта Build из одноимённого меню (либо нажать кнопку F7).

1.2.11 При наличие сообщений об ошибках или предупреждениях вернуться к предыдущему пункту и внести необходимые исправления.

1.2.12 Обязательно выполнить пошаговую отладку и трассировку скомпилированной программы средствами меню Debug перед программированием микроконтроллера.

1.2.13 Проверьте наличие подключения USB-кабеля программатора к соответствующему разъёму системного блока.

1.2.14 Для загрузки откомпилированного проекта в микроконтроллер выберите пункт меню Tools -> Program AVR -> Connect...

1.2.15 В появившемся окне Platform выберите пункт AVRISP mkII. В окне Port выберите пункт USB. Выбор любых отличающихся от приведенных выше пунктов в указанных окнах приведёт к невозможности программирования микроконтроллера!

1.2.16 Нажмите на кнопку Connect...

1.2.17 В появившемся окне сразу перейдите на вкладку Program. В поле Location укажите путь к файлу с расширением \*.hex. Он расположен в каталоге (подкаталогах) проекта, а его имя совпадает с именем проекта. Этот файл содержит машинные коды, которые будут выполняться микроконтроллером после загрузки в него файла \*.hex.

1.2.18 Нажмите кнопку Program. При этом происходит программирование (полное стирание и запись) Flash-памяти (памяти программ) микроконтроллера. Количество гарантированных операций стирания/записи этой памяти достаточно малое, поэтому не следует нажимать кнопку Program без надобности и бездумно перепрограммировать микроконтроллер!

1.2.19 Визуально оцените правильность работы светодиодного индикатора.

**1.3 Исходный текст программы**

.include "m8515def.inc"

.def temp = r16

.def long\_delay\_low =r24

.def long\_delay\_high =r25

.def counter = r17

.def value = r18

.EQU cycles\_nmb = 15

.EQU led\_line = 0xA006

.EQU t1 = 0b10101010

.CSEG

.ORG 0x0000

rjmp Init

reti;rjmp EXT\_INT0 ; IRQ0 Handler

reti;rjmp EXT\_INT1 ; IRQ1 Handler

reti;rjmp TIM1\_CAPT ; Timer1 Capture Handler

reti;rjmp TIM1\_COMPA ; Timer1 Compare A Handler

reti;rjmp TIM1\_COMPB ; Timer1 Compare B Handler

reti;rjmp TIM1\_OVF ; Timer1 Overflow Handler

reti;rjmp TIM0\_OVF ; Timer0 Overflow Handler

reti;rjmp SPI\_STC ; SPI Transfer Complete Handler

reti;rjmp USART\_RXC ; USART RX Complete Handler

reti;rjmp USART\_UDRE ; UDR0 Empty Handler

reti;rjmp USART\_TXC ; USART TX Complete Handler

reti;rjmp ANA\_COMP ; Analog Comparator Handler

reti;rjmp EXT\_INT2 ; IRQ2 Handler

reti;rjmp TIM0\_COMP ; Timer0 Compare Handler

reti;rjmp EE\_RDY ; EEPROM Ready Handler

reti;rjmp SPM\_RDY ; Store Program memory Ready

Init:

ldi temp, low(RAMEND)

out SPL, temp

ldi temp, high(RAMEND)

out SPH, temp

sbi ACSR, 7

ldi temp, 0b10000000

out MCUCR, temp

ldi ZL, low(led\_line)

ldi ZH, high(led\_line)

Infinite\_loop:

ldi counter, 1

ldi value, 0b10000000

ldi temp, 0b10000000

while\_loop:

cpi counter, cycles\_nmb

breq end\_while\_loop

st Z, value

cpi counter, 0x08

brlt incr

sub value, temp

lsl temp

rjmp end\_if

incr:

lsr temp

add value, temp

end\_if:

inc counter

rcall long\_delay

rjmp while\_loop

end\_while\_loop:

rjmp Infinite\_loop

long\_delay:

ldi long\_delay\_low,0x00

ldi long\_delay\_high,0x90

rcall short\_delay

sbiw long\_delay\_high:long\_delay\_low,0b00000001)

brne long\_loop

ret

.EXIT

**1.4 Особенности функционирования IDE AVR Studio 4**

В ходе лабораторной работы было выявлено, что IDE AVR Studio 4 позволяет во время отладки следить за содержимым во всех видах памяти, а также регистрах, это окно можно вызвать с использованием комбинации клавиш Alt + 4, также можно получить дизассемблер программы.

**Выводы**

В ходе лабораторной работы были изучены основы работы с лабораторным отладочным стендом EV8031/AVR. Также были получены практические навыки создания проекта, редактирования файлов исходных текстов и отладки программ в IDE AVR Studio 4. Было написано и отлажено приложени,е которое управляет линейкой светодиодов на учебном стенде с определенным алгоритмом.