**2 ЛАБОРАТОРНАЯ РАБОТА №2**

**СТАТИЧЕСКИЙ СЕМИСЕГМЕНТНЫЙ СВЕТОДИОДНЫЙ ИНДИКАТОР**

**Цель работы**

Ознакомиться с функциональными возможностями и внутренней структурой отладочного стенда EV8031/AVR. Изучить внутреннюю организацию статического семисегментного светодиодного индикатора, научиться отображать информацию на нём.

**2.1 Краткие теоретические сведения**

Для отображения цифровой информации наибольшее распространение получили семисегментные индикаторы, в которых изображение цифры состоит из семи линейных светодиодных сегментов расположенных в виде цифры 8.

На основе светодиодов и семисегментных индикаторов строятся подсистемы отображения информации. При построении подсистем отображения информации различают два подхода – динамическая и статическая схемы построения индикации.

Статическая индикация заключается в постоянной подсветке индикаторов НL1 – HLn от одного источника информации (см. рисунок 2.1).

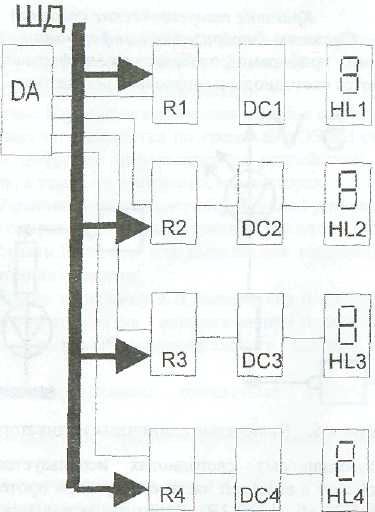


Рисунок 2.1 – Структурная схема статической индикации

DA – дешифратор адреса, необходим для выборки соответствующего регистра.

R1 – R4 – регистры, в которых временно хранится значение кода символа для отображения (соответствующий регистр выбирается с помощью DA).

DC1 – DC4 –преобразователи двоичного кода в семисегментный.

HL1 – HL4 – ceмисегментные индикаторы.

ШД – шина данных, по которой осуществляется передача данных для отображения.

**2.2 Порядок выполнения работы**

2.2.1 Запустите IDE AVR Studio 4.

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

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

2.2.4 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).

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

2.2.6 Принудительно (если это не сделано самой AVR Studio 4) установить флажок Create initial file. Желательно так же 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 >>.

2.2.7 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!

2.2.8 Нажать кнопку Finish для завершения создания проекта.

2.2.9 В появившемся окне написать программу на языке С или ассемблер с учётом варианта задания, который указан в таблице 2.1.

Таблица 2.1 – Варианты заданий

|  |  |
| --- | --- |
| № | Описание задания |
| 10 | На статическом семисегментном индикаторе (на всех знакоместах) реализовать "бегущую" строку, состоящую из десятичных цифр. Для этого организовать отображение информации по следующему алгоритму: 9876, 8765, 7654, 6543, 5432, 4321, 3210, 2109, 1098, 0987. Задержка на отображение одного состояния равна 1,5 с и организовывается программно. |

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

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

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

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

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

2.2.15 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!

2.2.16 Нажмите на кнопку Connect...

2.2.17 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.

2.2.18 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!

2.2.19 Визуально оцените правильность работы статического семисегментного светодиодного индикатора.

**2.3 Результат выполнения**

.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 right\_7\_seg = 0xB000

.EQU left\_7\_seg = 0xA000

.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 XL, low(right\_7\_seg)

ldi XH, high(right\_7\_seg)

ldi YL, low(left\_7\_seg)

ldi YH, high(left\_7\_seg)

Infinite\_loop:

ldi counter, 0

ldi ZH, high(numbers\*2)

ldi ZL, low(numbers\*2)

start\_loop:

cpi counter, 10

breq Infinite\_loop

lpm temp, Z+

st Y, temp

lpm temp, Z+

st X, temp

inc counter

rcall long\_delay

rjmp start\_loop

long\_delay:

ldi long\_delay\_low,0x00 ;

ldi long\_delay\_high,0x11

long\_loop:

rcall short\_delay

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

brne long\_loop

ret

short\_delay:

nop

ldi counter,0xC5 ;A±A·A?A?A·A?A? A¶A?A?A«A

short\_loop:

nop

dec counter

brne short\_loop

ret

numbers:

.db 0x98, 0x76, 0x87, 0x65, 0x76, 0x54, 0x65, 0x43, 0x54, 0x32, 0x43, 0x21, 0x32, 0x10, 0x21, 0x09, 0x10, 0x98, 0x09, 0x87

.EXIT

**2.3 Особенности функционирования IDE AVR Studio 4 выявленные в ходе выполнения лабораторной работы**

В ходе выполнения лабораторной работы никаких новых особенносте й IDE AVR Studio 4 не было выявлено.

**Выводы**

В ходе лабораторной работы была изучена работа с статическим семисегментным светодиодным индикатором в отладочном стенде EV8031/AVR. Была написана и отлажена программа которая выдает на семисегментном индикаторе бегущую строку с задержкой в 1.5с.