**4 ЛАБОРАТОРНАЯ РАБОТА №4**

**ЗНАКОСИНТЕЗИРУЮЩИЙ СВЕТОДИОДНЫЙ ИНДИКАТОР**

**Цель работы**

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

**4.1 Краткие теоретические сведения**

Знакосинтезирующий индикатор HG1 реализован на плате расширения в виде матрицы полупроводниковых светодиодов, состоящей из 5 столбцов и 7 строк. Управление светодиодной матрицей осуществляется по линиям РА0 – РА4 и РС0 – РС6. Например, для того, чтобы засветить точку с координатами [1;1], необходимо выставить уровень логической единицы на линии РА0, и уровень логического нуля на линии РС0.

Непосредственно индикация осуществляется следующим образом. Сначала выставляется информация на строки индикатора. Затем на некоторое время зажигаем крайний левый столбец. Меняем состояние строк и зажигаем следующий столбец и т.д. Если повторять указанную процедуру достаточно часто, то формируемое изображение будет казаться цельным и немигающим.

Ниже, на рисунке 4.1, изображена функциональная схема подключения знакосинтезирующего светодиодного индикатора к микроконтроллеру, реализованная в учебно-отладочном стенде EV8031/AVR. Из этого рисунка видно, что для обращения к шине данных знакосинтезирующего светодиодного индикатора (по сути, к строкам индикатора) необходимо произвести запись во внешнюю ячейку памяти по адресу 0x8002 (8002h). При этом нулевой бит (если он равен 0) отвечает за зажигание светодиода в самой нижней строке выбранного столбца, первый – во второй строке снизу и т.д. Выбор конкретного столбца индикатора осуществляется записью необходимой комбинации битов во внешнюю ячейку памяти по адресу 0x8000 (8000h). При этом нулевой бит (если он равен 1) отвечает за выбор крайнего правого столбца индикатора, первый – второго справа и т.д.

**4.2 Порядок выполнения работы**

4.2.1 Запустите IDE AVR Studio 4.

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

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

4.2.4 Ниже, в поле Location, укажите путь по которому будет располагаться проект.

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

4.2.6 Принудительно (если это не сделано самой AVR Studio 4) установить флажок Create initial file.

4.2.7 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!

4.2.8 Нажать кнопку Finish для завершения создания проекта.

4.2.9 В появившемся окне написать программу на языке С или ассемблер с учётом варианта задания

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

4.2.11 При наличие сообщений об ошибках или предупреждениях вернуться к предыдущему пункту и внести необходимые исправления. В случае некорректной работы программы выполнить её отладку средствами меню Debug.

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

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

4.2.14 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!

4.2.15 Нажмите на кнопку Connect...

4.2.16 В появившемся окне сразу перейдите на вкладку Program. В поле Location укажите путь к файлу с расширением \*.hex.

4.2.17 Нажмите кнопку Program. При этом происходит программирование (полное стирание и запись) Flash-памяти (памяти программ) микроконтроллера.

4.2.18 Визуально оцените правильность работы знакосинтезирующего светодиодного индикатора.

**4.3 Исходный текст программы**

.include "m8515def.inc"

.def VidL = r0

.def VidH = r1

.def temp = r16 ;регистр временного хранения

.def kadr\_addr = r17

.def kadr\_counter = r18

.def row = r19

.def col = r20

.def counter = r21 ;счетчик цикла в подпрограмме генерации задержки

.def long\_delay\_low =r24 ;младший байт счетчика длинной задержки

.def long\_delay\_high =r25 ;старший байт счетчика длинной задержки

.EQU synthes\_5x7\_row\_dyn\_7seg\_control = 0x8002

;адрес регистра маски выбора столбцов знакосинтезирующего индикатора

.EQU synthes\_5x7\_col = 0x8000

.CSEG ;определяем начало сегмента кода

.ORG 0x0000 ;определяем адрес начала сегмента кода в памяти

rjmp Init ; вектор прерывания по сбросу

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 ;установка SP на последний адрес SRAM

sbi ACSR,7

ldi temp, 0b10000000

out MCUCR, temp

ldi temp, low(Video\*2)

mov VidL, temp

ldi temp, high(Video\*2)

mov VidH, temp

ldi XL,low(synthes\_5x7\_row\_dyn\_7seg\_control)

ldi XH,high(synthes\_5x7\_row\_dyn\_7seg\_control)

;заносим адрес регистра, отвечающего за столбцы индикатора, в указатель

ldi YL,low(synthes\_5x7\_col)

ldi YH,high(synthes\_5x7\_col)

;\*\*\* Переходим в бесконечный цикл вывода картинки \*\*\*

Full\_kartina: ;цикл вывода полной картины

clr kadr\_addr

Kadr: ;начало цикла отображения новой картинки (нового кадра)

ldi kadr\_counter, 25

Main\_cycle: ;цикл вывода кадра (перебор всех столбцов, повторяется 25 раз)

mov ZL, VidL ;устанавливаем указатель Z на начало всей картинки

mov ZH, VidH

clr temp ;обнуляем регистр temp (для реализации двухбайтного сложения)

;вычисляем младший байт адреса начала текущего кадра во Flash-памяти

add ZL, kadr\_addr

;вычисляем старший байт адреса начала текущего кадра во Flash-памяти

adc ZH, temp

lpm row, Z+

st X, row ;выводим этот код на строки индикатора

ldi col, 0b00010000 ;строб перебора столбцов (1-й выбран)

st Y, col ;зажигаем нужный столбец

rcall long\_delay ;задержка на время зажигания одного столбца

lpm row, Z+

st X, row ;выводим этот код на строки индикатора

ldi col, 0b00001000 ;строб перебора столбцов (2-й выбран)

st Y, col ;зажигаем нужный столбец

rcall long\_delay ;задержка на время зажигания одного столбца

lpm row, Z+

st X, row ;выводим этот код на строки индикатора

ldi col, 0b00000100 ;строб перебора столбцов (3-й выбран)

st Y, col ;зажигаем нужный столбец

rcall long\_delay ;задержка на время зажигания одного столбца

lpm row, Z+

st X, row ;выводим этот код на строки индикатора

ldi col, 0b00000010 ;строб перебора столбцов (4-й выбран)

st Y, col ;зажигаем нужный столбец

rcall long\_delay

lpm row, Z+

st X, row ;выводим этот код на строки индикатора

ldi col, 0b00000001 ;строб перебора столбцов (5-й выбран)

st Y, col ;зажигаем нужный столбец

rcall long\_delay ;задержка на время зажигания одного столбца

dec kadr\_counter ;уменьшаем счетчик количества отображений одного кадра

brne Main\_cycle ;если текущий кадр не был отображен 25 раз,

;то продолжаем выводить его снова

inc kadr\_addr ;а если текущий кадр уже был отображен 25 раз,

;перемещаем указатель начала кадра на 1 байт дальше

cpi kadr\_addr, 76;если еще не вся картинка была отображена,

brne Kadr ;то переходим к отображению нового (следующего) кадра

rjmp Full\_kartina ;а если уже вся - повторяем все с самого начала

long\_delay:

ldi long\_delay\_low,0x2E ;загрузка в регистровую пару коэффициента задержки

ldi long\_delay\_high,0x00 ;(002Eh), это будет задержка на 4,99 мс

long\_loop: ;тело цикла занимает 796 + 2 + 2 = 800 тактов

rcall short\_delay ;короткая задержка

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

;вычитание из пары числа 1 (декремент длинного счетчика)

brne long\_loop ;если не 0,повторить цикл

ret ;возврат в основную программу

;\*\*\* Подпрограмма короткой задержки (нужна для генерации длинных задержек) \*\*\*

short\_delay: ;вся подпрограмма занимает ровно 796 тактов вместе с rcall и ret

nop

ldi counter,0xC5 ;счетчик цикла

short\_loop:

nop

dec counter

brne short\_loop ;команда ветвления по флагу нуля (зацикливание)

ret ;возврат в основную программу

Video:

.DB 0xFF, 0b10111111, 0b10111111, 0b10000000, 0b10111111, 0b10111111 ;T

.DB 0xFF, 0b10001100, 0b11101110, 0b11101110, 0b11101110, 0b10000000 ;Y

.DB 0xFF, 0b10000000, 0b10110111, 0b10110111, 0b10110111, 0b10000111 ;P

.DB 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,

.DB 0xFF, 0b10000000, 0b10111111, 0b10111111, 0b10111111, 0b10111111 ;Г

.DB 0xFF, 0b10000000, 0b10110111, 0b10110111, 0b10110111, 0b10000111 ;P

.DB 0xFF, 0b10000000, 0b11111100, 0b11100011, 0b10011111, 0b10000000 ;И

.DB 0xFF, 0b10000000, 0b10111111, 0b10111111, 0b10111111, 0b10111111 ;Г

.DB 0xFF, 0b11000001, 0b10111110, 0b10111110, 0b10111110, 0b11000001 ;0

.DB 0xFF, 0b10000000, 0b10110111, 0b10110111, 0b10110111, 0b10000111 ;P

.DB 0xFF, 0b10000000, 0b11111100, 0b11100011, 0b10011111, 0b10000000 ;И

.DB 0xFF, 0b10000000, 0b11111100, 0b10100011, 0b11011111, 0b10000000 ;Й

.DB 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,

**4.4 Особенности IDE AVRStudio выявленные в ходе выполнения лабораторной работы**

В ходе лабораторной работы никаких новых особенностей IDE AVRStudio не было выявлено.

**Выводы**

В ходе лабораторной работы были изучены принципы отображения информации на знакосинтезирующем светодиодном индикаторе в учебно-отладочном стенде EV8031/AVR. Была написана программа, которая выводит бегущую строку состоящую из собственного имени и фамилии. Преимуществом знакосинтезирующего светодиодного индикатора является возможность отображать существенно более разнообразные изображения нежели на семисегментном индикаторе.