Лабораторная работа № 11

# Использование EEPROM-памяти данных микроконтроллера

Цель работы:

1. Изучение принципов работы с энергонезависимой (EEPROM) памя- тью данных микроконтроллера.
2. Программирование работы с EEPROM-памятью данных МК

ATmega8535.

Оборудование и программное обеспечение:

1. Лабораторный макет MK8535 в составе: модуль базовый МБ8535 +

модуль сменный МС01.

1. Кабель соединительный для программирования МК.
2. Персональный компьютер.
3. Интегрированная среда программирования CodeVisionAVR.

# Подготовка к работе

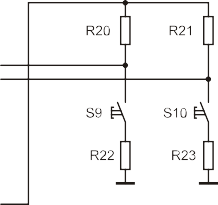
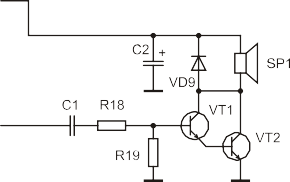
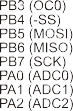
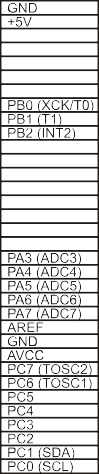
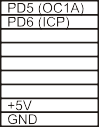
Структурная схема лабораторного макета приведена на рис. 68. Рассмотрим и выполним следующий пример. Заполним первые 256 байт EEPROM-памяти (начиная с адреса 0) возрастающими значениями от 0 до

255. Затем будем последовательно считывать эти ячейки и выводить зна- чения на светодиодный двоичный индикатор, подключенный к порту A. Мы должны увидеть ту же последовательность двоичных чисел (0…255).

Во время записи в EEPROM двоичные значения чисел выводятся также и на светодиодный индикатор. Процесс записи заканчивается гудком динамика и затем начинается индикация значений, считанных начиная с адреса 0. Чтение из EEPROM происходит со скоростью 1 раз в секунду.

Ниже приведен исходный текст программы для МК ATmega8535. В нем опущены те строки, автоматически сформированные CodeWizardAVR, которые не относятся непосредственно к данному примеру.

# EECR |= 0x04; // Разрешить запись в EEPROM EECR |= 0x02; // Начать запись в EEPROM



**}**

# unsigned char EEPROM\_read(unsigned int Addr)

**{**

**while (EECR & 0x02); // Ждать завершения**

**// предыдущей записи**

**EEAR = Addr; // Установить адрес**

**EECR |= 0x01; // Выполнить чтение**

**return EEDR; // Забрать данные из EEDR**

**}**

# void main(void)

**{**

**unsigned int i;**

**DDRA = 0xFF; // Настроить порт A PORTA = 0x00; // на вывод**

**DDRB = 0x08; // Настр. линию 3 PORTB = 0x00; // порта B на вывод**

**for (i=0; i<256; i++) // Запись чисел 0..255**

**{ // в EEPROM-память,**

**EEPROM\_write(i, i); // начиная с адреса 0 PORTA = i;**

Рис. 68. Структурная схема лабораторного макета. Для модуля МС01 показана только та часть схемы, которая используется в данной работе

# #include <mega8535.h> #include <delay.h>

**void EEPROM\_write(unsigned int Addr, unsigned char Data)**

**{**

**while (EECR & 0x02); // Ждать завершения**

**// предыдущей записи**

**EEAR = Addr; // Установить адрес**

**EEDR = Data; // Установить данные**

**}**

# for (i=150; i>0; i--) // Гудок

**{**

# PORTB.3 = i & 0x01;

**delay\_ms(1);**

**}**

**i = 0; // Адрес = 0**

**while (1) // Вывод на индикацию**

**{ // 256 значений**

**PORTA = EEPROM\_read(i); // EEPROM-памяти, i++; // начиная с адреса 0 if (i >= 0xFF) i = 0;**

**delay\_ms(1000);**

**};**

**}**

# Задание

1. Создать проект и реализовать рассмотренный выше пример про- граммы.
2. Вместо модуля МС01 установить модуль МС03 и реализовать про- ект «Кодовый замок». Код замка хранится в EEPROM-памяти. Пользователь вводит код с клавиатуры и в случае совпадения на ЖК-индикатор выводится «OPEN».

# РАБОТА С МОДУЛЕМ ЗНАКОСИНТЕЗИРУЮЩЕГО ЖКИ

Для отображения цифро-буквенных сообщений в приборах, по- строенных на МК, удобно использовать знакосинтезирующие ЖКИ, выполненные в виде функционально законченных модулей. Выпускают ЖКИ, различающиеся как размерами, так и числом отображаемых сим- волов. Наличие встроенной подсветки позволяет использовать их в лю- бых условиях освещенности.

Модульные ЖКИ, как правило, имеют унифицированные интер-

фейсы, совместимые с микропроцессорными системами. Для вывода

информации на ЖКИ, подключенный к МК, достаточно выполнить пра- вила, предписываемые интерфейсом модуля, т.е. программным спосо- бом сформировать на выводах порта МК нужные комбинации сигналов в соответствии с временными диаграммами.

# Модуль знакосинтезирующего ЖКИ типа 1602

Жидкокристаллические знакосинтезирующие модули типа 1602

представляют собой двухстрочные ЖК-индикаторы. Каждая строка

содержит по 16 знакомест в виде матриц 5×8 точек + курсор, что позво- ляет отображать любые символы. Интервалы между символами шири- ной в одну отображаемую точку. На рис. 69 показан ЖКИ, знакогенера- тор которого содержит русские буквы.



Рис. 69. Модуль ЖКИ типа 1602

Модули ЖКИ содержат контроллеры управления

ЖК-панелью,

как правило, аналогичные HD44780 фирмы Hitachi или KS0066 фирмы Samsung. Модули выпускаются со светодиодной или люминесцентной подсветкой. Структурная схема модуля ЖКИ представлена на рис. 70.

Модули управляются по 4-битному или 8-битному интерфейсу. Разрядность интерфейса задается пользователем при помощи соответ- ствующих команд. Команды интерфейса, временные диаграммы и пр. можно, при необходимости, узнать из технического описания модуля ЖКИ. Мы же воспользуемся тем, что в библиотеке CodeVisionAVR име- ются функции для работы со стандартными модулями ЖКИ. Эти функции рассчитаны на 4-битную передачу данных, в результате для подключения модуля достаточно лишь одного порта микроконтроллера. Для того, чтобы сообщить функциям, к какому порту подключен ЖКИ, нужно включить в программу следующий фрагмент (здесь 0x1B – адрес порта A):

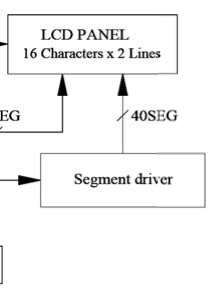
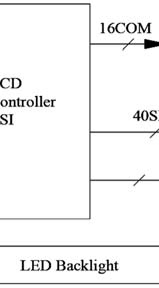
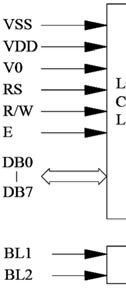


Рис. 70. Структурная схема модуля типа 1602

Назначение внешних выводов модуля ЖКИ типа 1602 описано

# #asm

**.equ lcd\_port #endasm**

# = 0x1B

в табл. 28.

*Таблица 28*

Библиотека подключается строкой **#include <lcd.h>**.

В табл. 29 приведено описание некоторых функций этой библиотеки.

*Таблица 29*

Назначение внешних выводов модуля ЖКИ типа 1602

|  |  |  |
| --- | --- | --- |
| № | Наимен. | Назначение |
| 1 | GND | Общий вывод (0V) |
| 2 | VCC | Напряжение питания (+5V) |
| 3 | V0 | Управление контрастностью |
| 4 | A0 | Адресный сигнал – выбор между передачей данных и коман- дами управления |
| 5 | R/W | Выбор режима – чтение/запись |
| 6 | E | Разрешение обращений к модулю, а также строб данных |
| 7 | DB0 | Шина данных (8-битный режим). Младший бит в 8-битном режиме |
| 8 | DB1 | Шина данных (8-битный режим) |
| 9 | DB2 | Шина данных (8-битный режим) |
| 10 | DB3 | Шина данных (8-битный режим) |
| 11 | DB4 | Шина данных (8- и 4-битный режимы). Младший бит в 4-  битном режиме |
| 12 | DB5 | Шина данных (8- и 4-битный режимы) |
| 13 | DB6 | Шина данных (8- и 4-битный режимы) |
| 14 | DB7 | Шина данных (8- и 4-битный режимы) |
| 15 | +BL | + питания подсветки |
| 16 | –BL | – питания подсветки |

Некоторые функции для работы с модулями ЖКИ

|  |
| --- |
| **void \_lcd\_ready(void);**  Готовность ЖКИ. |
| **void \_lcd\_write\_data(unsigned char data);**  Запись данных в ЖКИ. |
| **void lcd\_write\_byte(unsigned char addr, unsigned char data);**  Запись байта в знакогенератор ЖКИ или в память дисплея. |
| **unsigned char lcd\_read\_byte(unsigned char addr);**  Чтение байта из знакогенератора ЖКИ или из памяти дисплея. |
| **void lcd\_gotoxy(unsigned char x, unsigned char y);**  Устанавливает указатель ЖКИ на знакоместо **x** в строке **y** (**x**=0..39, **y**=0..3). |
| **void lcd\_clear(void);**  Очистка экрана ЖКИ. |
| **void lcd\_putchar(char c);**  Вывод символа на ЖКИ. |
| **void lcd\_puts(char \*str);**  Вывод строки, расположенной в SRAM микроконтроллера, на ЖКИ. |
| **void lcd\_putsf(char flash \*str);**  Вывод строки, расположенной во FLASH-памяти МК, на ЖКИ. |
| **unsigned char lcd\_init(unsigned char lcd\_columns);**  Инициализация контроллера ЖКИ. |