Лабораторная работа № 4

# Использование внешних прерываний в микроконтроллерах AVR

Цель работы:

1. Изучение механизма внешних прерываний микроконтроллеров AVR.
2. Исследование режимов работы входов прерываний 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.

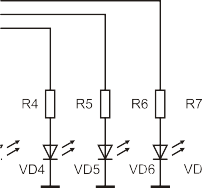
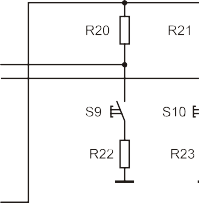
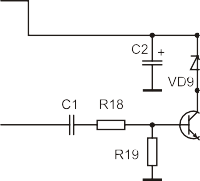
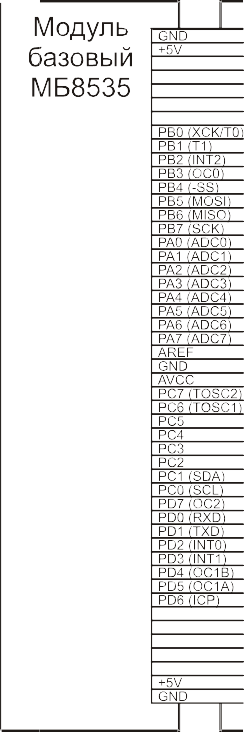
# Подготовка к работе

Рассмотрим и выполним пример, где на фоне основной программы, занятой генерацией гудка, по сигналам со входов внешних прерываний INT0 и INT1 происходит увеличение или уменьшение значения счетчика и отображение его на индикаторе.

Схема лабораторного макета приведена на рис. 35. Звуковой дина- мик SP1 подключен к выходу 3 порта B. Индикатором двоичного значе- ния счетчика служат светодиоды VD1…VD8, подключенные к выходам порта A. Источниками сигналов прерываний служат кнопки S9 и S10, подключенные ко входам INT0 и INT1. Как видно из схемы, при нажатии на кнопку формируется спадающий фронт сигнала, а при её отпускании – возрастающий. Пусть по нажатию кнопки S9 происходит увеличение зна- чения счетчика, а по нажатию кнопки S10 – уменьшение.

Создадим проект в CodeVisionAVR и зададим нужный режим работы микроконтроллера. В окне **CodeWizardAVR** на вкладке **Chip** выбрать тип микроконтроллера и рабочую частоту. На вкладке **Ports** задать режимы портов A, B и D: все линии порта A – на вывод, линию 3 порта B – на вывод, все линии порта D – на ввод. На вкладке **External IRQ** (рис. 36) отметить **INT0 Enabled** и **INT1 Enabled** и выбрать для них режим **Falling Edge**.

В созданном при помощи **CodeWizardAVR** тексте программы строки



# // External Interrupt 0 service routine interrupt [EXT\_INT0] void ext\_int0\_isr(void)

**{**

# // Place your code here

**}**

# // External Interrupt 1 service routine interrupt [EXT\_INT1] void ext\_int1\_isr(void)

**{**

# // Place your code here

**}**

представляют собой заготовки процедур обработки внешних пре- рываний для INT0 и INT1, оформленных в виде функций. На место тела функции можно будет вписать свой код обработки.

Строки

Рис. 35. Структурная схема лабораторного макета. Для модуля МС01 показана только та часть схемы, которая используется в данной работе



Рис. 36. Подготовка к работе входов INT0 и INT1

**// External Interrupt(s) initialization**

**// INT0: On**

**// INT0 Mode: Falling Edge**

**// INT1: On**

**// INT1 Mode: Falling Edge**

**// INT2: Off GICR |= 0xC0; MCUCR = 0x0A; MCUCSR = 0x00; GIFR = 0xC0;**

реализуют выбранный нами режим работы входов INT0 и INT1.

Строка

# #asm(«sei») // Global enable interrupts

представляет собой команду ассемблера SEI (разрешить все прерывания), вставленную в текст на языке Си.

Доработаем исходный текст программы так, чтобы реализовать наш пример. Для этого допишем код для генерации гудка и коды обра-

ботки прерываний. В качестве двоичного счетчика используем регистр

PORTA. Остальные строки оставим без изменений (здесь они не показаны):

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

**// Процедура обработки внешнего прерывания INT0 interrupt [EXT\_INT0] void ext\_int0\_isr(void)**

**{**

# if (PORTA < 0xff) ++PORTA;

**}**

# // Процедура обработки внешнего прерывания INT1 interrupt [EXT\_INT1] void ext\_int1\_isr(void)

**{**

# if (PORTA > 0x00) --PORTA;

**while (1) // Генерация гудка**

**{**

**PORTB.3 = 1;**

**delay\_us(500); // задержка 500 мкс**

**PORTB.3 = 0;**

**delay\_us(500); // задержка 500 мкс**

**}**

**}**

# Задание

1. Создать проект и реализовать рассмотренный выше пример программы.
2. Изменить код программы так, чтобы прерывания возникали при отпускании кнопок (т.е. по возрастающему фронту сигнала).
3. Изменить код программы так, чтобы по прерыванию от кнопок повышалась и понижалась частота гудка.

**}**

# void main(void)

**{**

**PORTA = 0x00; // Настроить все линии порта A DDRA = 0xFF; // на вывод**

**PORTD = 0x00; // Настроить все линии порта D DDRD = 0x00; // на ввод**

**PORTB = 0x00; // К выходу PB3**

**DDRB = 0x08; // подключен динамик**

**// Настройка режима прерываний:**

**// INT0: Разрешено**

**// режим INT0: по спадающему фронту**

**// INT1: Разрешено**

**// режим INT1: по спадающему фронту**

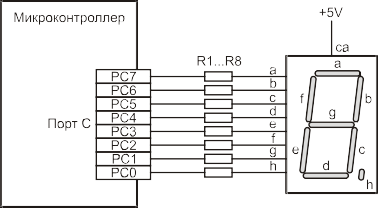
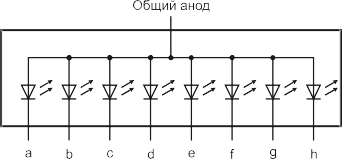
**// INT2: Запрещено GICR |= 0xC0; MCUCR = 0x0A; MCUCSR = 0x00; GIFR = 0xC0;**

**#asm(«sei») // разрешить все прерывания**

# ДИНАМИЧЕСКАЯ ИНДИКАЦИЯ

**Принцип динамической индикации**

Устройства на основе микроконтроллеров зачастую имеют мно- горазрядные индикаторы (дисплеи) для вывода цифровой или цифро- буквенной информации. На рис. 37, а показана внутренняя схема свето- диодного семисегментного индикатора. Сегменты a…g образуют цифру, сегмент h – точка.



а) б)

Рис. 37. Семисегментный индикатор и его подключение к МК

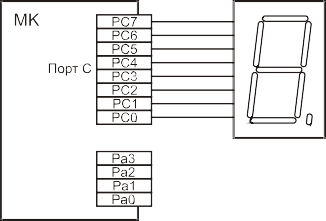
Схема подключения одного индикатора к порту МК напрямую, без дешифратора, приведена на рис. 37, б (дешифратор можно реализо- вать программно). Резисторы R1…R8 служат для ограничения тока све- тодиодов. Видно, что для одного индикатора требуется один 8-разрядный порт МК. Если дисплей многоразрядный, то, как правило, у МК не найдется такого числа свободных портов.

Сократить затраты поможет динамическая индикация, принцип которой состоит в следующем. Известно, что если световые вспышки повторяются чаше 20–25 раз в секунду, то человеческий глаз, вследствие инерционности, воспринимает их как непрерывное свечение. Тогда можно все индикаторы разрядов подключить параллельно к выходу одного дешифратора, но зажигать их поочерёдно, подавая при этом на дешифратор код очередной цифры. Если чередовать индикацию цифр с нужной частотой, то будет создана иллюзия слитного многоразрядного изображения.

Если используется микроконтроллер, то можно исключить и де- шифратор, реализовав его функцию программно. Тогда блок семисег-

ментных индикаторов можно подключить непосредственно к порту МК. На рис. 38 показан 4-разрядный дисплей, реализованный по схеме дина- мической индикации. Видно, что для подключения 4 индикаторов вме- сто 32 линий потребовалось лишь 12. Процесс вывода числа «2013» на 4-разрядный дисплей разложен по фазам.

Рис. 38. Последовательность вывода числа «2013»



С увеличением числа разрядов динамическая индикация стано- вится более выгодной. Следует, однако, учитывать, что доля времени на свечение отдельного разряда уменьшается, что приводит к снижению общей яркости. Скомпенсировать яркость можно увеличением импульс- ного тока светодиодов (разумеется, не превышая предельных значений).