Лабораторная работа № 3

# Ввод внешних сигналов в микроконтроллерах AVR

Цель работы:

1. Изучение способов ввода внешних двоичных сигналов в микро- контроллеры AVR.
2. Изучение способов устранения дребезга контактов при вводе сигналов.

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 две кнопки подключены к входным линиям порта МК. Ввод сигналов от кнопок происходит путем циклического программного опроса состояния входов порта. По сигналу от первой кнопки происходит увеличение значения двоичного счетчика и отображение его на индикаторе, а по сигналу от второй – уменьшение.

Схема лабораторного макета приведена на рис. 29. Кнопки S9 и S10 подключены к линиям 2 и 3 порта D. Индикатором значения счет- чика служат светодиоды VD1…VD8, подключенные к выходам порта A. В качестве двоичного счетчика используем регистр PORTA – ведь это, по существу, ячейка памяти. Как видно из схемы, при нажатии на кнопку формируется спадающий фронт сигнала, а при её отпускании – возраста- ющий. Пусть по нажатию кнопки S9 происходит увеличение значения 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 и D: все линии порта A – на вывод, все линии порта D – на ввод.

Заготовку текста программы, созданную **CodeWizardAVR**, допол- ним процедурами опроса кнопок (они помещены в бесконечный цикл **while(1)**).

# while (1)

**{**

# // ждем нажатия кнопок S9 или S10 while ((PIND.2 != 0) && (PIND.3 != 0));

**if (PIND.2 == 0) //если нажата кнопка S9**

**{**

**// инкремент счетчика и вывод на индикатор**

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

**while (PIND.2 == 0); //ждать отпускания S9**

**}**

# if (PIND.3 == 0) //если нажата кнопка S10

**{**

**// декремент счетчика и вывод на индикатор**

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

**while (PIND.3 == 0); //ждать отпускания S10**

**}**

**}**

**}**

Рис. 29. Структурная схема лабораторного макета. Для модуля МС01 показана только та часть схемы, которая используется в данной работе

# #include <mega8535.h> 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**

Откомпилируем и запустим эту программу на микроконтроллере. Скорее всего, мы обнаружим, что на одно нажатие кнопки может проис- ходить по несколько приращений счетчика, т.е. имеет место влияние дребезга контактов.

Чтобы избавиться от влияния дребезга контактов, введем про- граммные задержки как после обнаружения нажатия на кнопку, так и после её отпускания. Для этого воспользуемся процедурой delay\_ms() из библиотеки delay.h. Параметр процедуры – величина задержки в мил- лисекундах.

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

**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**

**while (1)**

**{**

# // ждем нажатия кнопок S9 или S10 while ((PIND.2 != 0) && (PIND.3 != 0));

**if (PIND.2 == 0) //если нажата кнопка S9**

**{**

**delay\_ms(20); // задержка 20 мсек**

**// инкремент счетчика и вывод на индикатор**

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

**while (PIND.2 == 0); // ждать отпускания S9 delay\_ms(20); // задержка 20 мсек**

**}**

# if (PIND.3 == 0) //если нажата кнопка S10

**{**

**delay\_ms(20); // задержка 20 мсек**

**// декремент счетчика и вывод на индикатор**

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

**while (PIND.3 == 0); // ждать отпускания S10 delay\_ms(20); // задержка 20 мсек**

# ВНЕШНИЕ ПРЕРЫВАНИЯ В МИКРОКОНТРОЛЛЕРАХ AVR

**Общие сведения**

Прерывание приостанавливает нормальный ход программы для выполнения приоритетной задачи, определяемой внутренним или внеш- ним событием микроконтроллера. При возникновении прерывания мик- роконтроллер сохраняет в стеке содержимое счетчика команд PC и за- гружает в него адрес соответствующего вектора прерывания. По этому адресу, как правило, находится команда безусловного перехода к под- программе обработки прерывания. Последней командой подпрограммы обработки прерывания должна быть команда RETI, которая обеспечивает возврат в основную программу и восстановление предварительно сохра- ненного счетчика команд (рис. 30, а).

**}**

**}**

**}**

Запустим доработанную программу на микроконтроллере. Теперь можно видеть, что единичному нажатию на кнопку соответствует одно- кратное приращение счетчика. Величины задержек можно при необхо- димости скорректировать, чтобы добиться надежной работы кнопок.

# Задание

1. Создать проект и реализовать рассмотренные выше примеры. Наблю- дать работу программы сначала в варианте без устранения дребезга контактов, затем – улучшенный вариант.
2. Изменить код программы так, чтобы изменения счетчика происходили в момент отпускания нажимаемых кнопок (т.е. по возрастающему фронту сигнала).
3. Создать новый проект, где для устранения дребезга контактов реали- зовать метод подсчета заданного числа совпадающих значений сигнала. Число N подобрать экспериментально.

а) б)

Рис. 30. Обработка прерывания (а), обработка вложенных прерываний (б)

Если во время обработки прерывания поступит другой запрос на прерывание с более высоким приоритетом (рис. 30, б), то текущая про- цедура, соответственно, будет приостановлена, и произойдет переход к новой процедуре. По завершении ее произойдет возврат к продолже- нию прерванной процедуры, а затем и к основной программе. Такие прерывания называют вложенными. Однако, если второй запрос будет иметь меньший приоритет, то ему придется дожидаться окончания те- кущей процедуры обработки прерывания.

Поскольку источниками прерываний, в частности, являются раз- личные периферийные устройства микроконтроллеров, количество пре- рываний зависит от конкретной модели.

# Таблица векторов прерываний

Иерархия векторов прерываний для микроконтроллера

ATmega8535 приведена в табл. 4.

*Таблица 4*

Таблица векторов прерываний микроконтроллера ATmega8535

|  |  |  |  |
| --- | --- | --- | --- |
| Источник | Описание | Номер вектора | Адрес |
| RESET | Внешний вывод RESET | 1 | $0000 |
| INT0 | Внешнее прерывание 0 | 2 | $0001 |
| INT1 | Внешнее прерывание 1 | 3 | $0002 |
| TIMER2 СOMP | Совпадение таймера/счетчика T2 | 4 | $0003 |
| TIMER2 OVF | Переполнение таймера/счетчика T2 | 5 | $0004 |
| TIMER1 CAPT | Захват таймера/счетчика T1 | 6 | $0005 |
| TIMER1 СОМРA | Совпадение «А» таймера/счетчика T1 | 7 | $0006 |
| TIMER1 СОМРВ | Совпадение «В» таймера/счетчика T1 | 8 | $0007 |
| TIMER1 OVF | Переполнение таймера/счетчика T1 | 9 | $0008 |
| TIMER0 OVF | Переполнение таймера/счетчика T0 | 10 | $0009 |
| SPI, STC | Передача по SPI завершена | 11 | $000А |
| USART, RXC | USART, прием завершен | 12 | $000B |
| USART, UDRE | Регистр данных USART пуст | 13 | $000C |
| USART, ТХС | USART, передача завершена | 14 | $000D |
| ADC | АЦП, преобразование завершено | 15 | $000E |
| ЕЕ RDY | EEPROM, готово | 16 | $000F |
| ANA COMP | Аналоговый компаратор | 17 | $0010 |
| TWI | Двухпроводный интерфейс TWI | 18 | $0011 |
| INT2 | Внешнее прерывание 2 | 19 | $0012 |
| TIMER0 COMP | Совпадение таймера/счетчика T0 | 20 | $0013 |
| SPM RDY | Готовность SPM | 21 | $0014 |

Все микроконтроллеры AVR имеют многоуровневую систему приоритетных прерываний. Младшие адреса памяти программ отведены под таблицу векторов прерываний. Каждому прерыванию соответствует адрес в этой таблице, который загружается в счетчик команд при воз- никновении прерывания, и таким образом происходит переход к под- программе обслуживания этого прерывания.

Положение вектора в таблице также определяет и приоритет соответствующего прерывания: чем меньше адрес, тем выше приори- тет прерывания. Как видно из табл. 4, RESET имеет наивысший прио- ритет, затем внешнее прерывание INT0 и т.д. Размер вектора прерыва- ния для микроконтроллера ATmega8535 составляет 1 байт, соответ- ственно для перехода к подпрограммам обработки прерываний исполь- зуются команды RJMP.

Если прерывания в работе микроконтроллера не предусматрива- ются, то на месте таблицы векторов прерываний может быть размещена часть основной программы.

# Обработка прерываний

Для глобального разрешения/запрещения прерываний предназна- чен флаг I регистра SREG. Дня разрешения прерываний он должен быть установлен в 1, а для запрещения сброшен в 0. Индивидуальное разре- шение или запрещение (маскирование) прерываний производится уста- новкой/сбросом соответствующих разрядов регистров масок прерыва- ний, рассматриваемых ниже.

При возникновении прерывания флаг I регистра SREG аппаратно сбрасывается, запрещая тем самым обработку следующих прерываний. Однако в подпрограмме обработки прерывания этот флаг можно снова установить в 1 для разрешения вложенных прерываний. При возврате из подпрограммы обработки прерывания (при выполнении команды RETI) флаг I устанавливается аппаратно.

Все имеющиеся прерывания можно разделить на два типа. Пре- рывания первого типа генерируются при наступлении некоторого собы- тия, в результате которого устанавливается флаг прерывания. Затем, если прерывание разрешено, в счетчик команд загружается адрес вектора соответствующего прерывания. При этом флаг прерывания аппаратно сбрасывается. Он также может быть сброшен программно, путем записи лог. 1 в разряд регистра, соответствующий флагу.

Прерывания второго типа не имеют флагов прерываний и генери- руются в течение всего времени, пока присутствуют условия, необходи-

мые для генерации прерывания. Соответственно, если условия, вызыва- ющие прерывание, исчезнут до разрешения прерывания, генерации пре- рывания не произойдет.

Следует помнить, что при вызове подпрограмм обработки преры- ваний регистр состояния SREG не сохраняется. Поэтому пользователь должен самостоятельно запоминать содержимое этого регистра при входе в подпрограмму обработки прерывания (если это необходимо) и восста- навливать его значение перед вызовом команды RETI.

Микроконтроллеры семейства Mega поддерживают очередь пре- рываний, которая работает следующим образом: если условия генерации одного или более прерываний возникают в то время, когда флаг общего разрешения прерываний сброшен (все прерывания запрещены), соответ- ствующие флаги устанавливаются в 1 и остаются в этом состоянии до установки флага общего разрешения прерываний. После разрешения прерываний выполняется их обработка в порядке приоритета.

Наименьшее время отклика для любого прерывания составляет 4 машинных цикла, в течение которых происходит сохранение счетчика команд в стеке. В течение последующих двух (для моделей ATmega8x и ATmega8515x) или трех циклов выполняется команда перехода к под- программе обработки прерывания. Если прерывание произойдет во время выполнения команды, длящейся несколько циклов, то генерация преры- вания произойдет только после выполнения этой команды. Если же пре- рывание произойдет во время нахождения микроконтроллера в «спя- щем» режиме, время отклика увеличивается еще на 4 машинных цикла.

Возврат в основную программу занимает 4 машинных цикла, в течение которых происходит восстановление счетчика команд из стека. После выхода из прерывания процессор всегда выполняет одну команду основной программы, прежде чем обслужить любое отложенное преры- вание.

# Внешние прерывания

Внешними называют прерывания, которые возникают при по- ступлении сигналов на входы INT0, INT1 или INT2. Для разреше- ния/запрещения внешних прерываний предназначен регистр **GICR** (General Interrupt Control Register – общий регистр управления прерыва- ниями). На рис. 31 показан формат этого регистра для микроконтроллера ATmega8535.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **INT1** | **INT0** | **INT2** |  |  |  | **IVSEL** | **IVCE** |
| R/W | R/W | R/W | R | R | R | R/W | R/W |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Рис. 31. Формат регистра GICR микроконтроллера ATmega8535

*Таблица 5*

Разряды 5, 6, 7 регистра GICR микроконтроллера ATmega8535

|  |  |  |
| --- | --- | --- |
| Бит | Название | Описание |
| 7 | INT1 | **Разрешение внешнего прерывания INT1**. Если этот разряд установлен в 1 и флаг I регистра SREG также установлен в 1, то разрешается внешнее прерывание с вывода INT1. Условие генерации прерывания определяется содержимым разрядов ISC11 и ISC10 регистра MCUCR |
| 6 | INT0 | **Разрешение внешнего прерывания INT0**. Если этот разряд установлен в 1 и флаг I регистра SREG также установлен в 1, то разрешается внешнее прерывание с вывода INT0. Условие генерации прерывания определяется содержимым разрядов ISC01 и ISC00 регистра MCUCR |
| 5 | INT2 | **Разрешение внешнего прерывания INT2**. Если этот разряд установлен в 1 и флаг I регистра SREG также установлен в 1, то разрешается внешнее прерывание с выводa INT2. Условие генерации прерывания определяется содержимым разряда ISC2 регистра MCUCSR |

Регистр **GIFR** (General Interrupt Flag Register – общий регистр флагов прерываний) предназначен для индикации наступления внешних прерываний. Формат этого регистра показан на рис. 32, а описание его разрядов приведено в табл. 6.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **INTF1** | **INTF0** | **INTF2** |  |  |  |  |  |
| R/W | R/W | R/W | R | R | R | R | R |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Рис. 32. Формат регистра GIFR микроконтроллера ATmega8535

*Таблица 6*

Разряды 5, 6, 7 регистра GIFR микроконтроллера ATmega8535

|  |  |  |
| --- | --- | --- |
| Бит | Название | Описание |
| 7 | INTF1 | **Флаг внешнего прерывания INT1**. Если в результате собы- тия на выводе INT1 сформировался запрос на внешнее преры- вание, этот разряд устанавливается в 1. Флаг сбрасывается аппаратно при запуске подпрограммы обработки прерывания или программно, записью в него лог. 1. Флаг INTF1 сброшен постоянно, если генерация прерывания должна происходить по НИЗКОМУ уровню на выводе INT1. |
| 6 | INTF0 | **Флаг внешнего прерывания INT0**. Если в результате собы- тия на выводе INT0 сформировался запрос на внешнее преры- вание, этот разряд устанавливается в «1». Флаг сбрасывается аппаратно при запуске подпрограммы обработки прерывания или программно, записью в него лог. 1. Флаг INTF0 сброшен постоянно, если генерация прерывания должна происходить по НИЗКОМУ уровню на выводе INT0. |
| 5 | INTF2 | **Флаг внешнего прерывания INT2**. Если в результате собы- тия на выводе INT2 сформировался запрос на внешнее преры- вание, этот разряд устанавливается в 1. Флаг сбрасывается аппаратно при запуске подпрограммы обработки прерывания или программно, записью в него лог. 1. Флаг INTF2 сброшен постоянно, если генерация прерывания должна происходить по НИЗКОМУ уровню на выводе INT2. |

Прерывания INT0 и INT1 могут быть сгенерированы по нараста- ющему/спадающему фронту сигнала или при появлении низкого уровня на входе. Условия генерации этих прерываний определяются состояни- ем младших 4-х разрядов регистра **MCUCR** (рис. 33) согласно табл. 7.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **SM2** | **SE** | **SM1** | **SM0** | **ISC11** | **ISC10** | **ISC01** | **ISC00** |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Рис. 33. Формат регистра MCUCR микроконтроллера ATmega8535

*Таблица 7*

Разряды 0…3 регистра MCUCR микроконтроллера ATmega8535

|  |  |  |
| --- | --- | --- |
| Бит | Название | Описание |
| 32 | ISC11 ISC10 | Определяют условие генерации внешнего прерывания INT1 |
| ISC11 | ISC10 | Условие |
| 0 | 0 | По низкому уровню на выводе INT1 |
| 0 | 1 | По любому из фронтов сигнала на выводеINT1 |
| 1 | 0 | По спадающему фронту сигнала на выводеINT1 |
| 1 | 1 | По нарастающему фронту сигнала на вы- воде INT1 |
| 10 | ISC01 ISC00 | Определяют условие генерации внешнего прерывания INT0 |
| ISC01 | ISC00 | Условие |
| 0 | 0 | По низкому уровню на выводе INT0 |
| 0 | 1 | По любому из фронтов сигнала на выводеINT0 |
| 1 | 0 | По спадающему фронту сигнала на выводеINT0 |
| 1 | 1 | По нарастающему фронту сигнала на вы- воде INT0 |

Обнаружение фронтов сигналов на выводах INT0 и INT1 осу- ществляется синхронно, поэтому минимальная длительность импульса, гарантирующая генерацию прерывания, составляет один период такто- вого сигнала микроконтроллера.

Если генерация прерывания должна происходить по низкому уровню, то он должен удерживаться на выводе до окончания выполне- ния текущей команды, в противном случае прерывание сгенерировано не будет.

Прерывание INT2 может быть сгенерировано только по нараста- ющему или спадающему фронту сигнала. Условие генерации прерыва- ния INT2 определяется состоянием разряда ISC2 регистра **MCUCSR** в модели ATmega8535. Если этот разряд сброшен в 0 (начальное состоя- ние), прерывание будет сгенерировано по спадающему фронту сигнала, если установлен в 1 – по нарастающему фронту. Обнаружение перепа-

дов сигнала на выводе INT2 осуществляется асинхронно, при этом минимальная длительность импульса, гарантирующая генерацию пре- рывания, составляет 50 нс.

В заключение скажем, что все внешние прерывания генерируются даже в том случае, если соответствующие выводы сконфигурированы как выходы. Эта особенность микроконтроллеров позволяет генериро- вать прерывания программно.

# Дребезг контактов и обработка прерываний

Сигнал от внешнего источника прерывания может быть подвер- жен влиянию дребезга контактов. Если, например, кнопка или клавиша подключена к входу внешнего прерывания, то при замыкании контактов будет запускаться процедура обработки этого прерывания. Однако, вследствие возможного дребезга контактов, за одно нажатие эта проце- дура может быть запущена неконтролируемое число раз. Чтобы избе- жать повторных запусков, процедура обработки прерывания должна сразу после запуска запретить внешние прерывания, выполнить задержку на время, заведомо большее длительности переходного процесса, выполнить предназначенные действия и затем, перед завершением, вновь разрешить внешние прерывания (см. рис. 34).

Рис. 34. Устранение влияния дребезга контактов при обработке внешнего прерывания