

Лабораторная работа 2.

Обработка внешних прерываний в микроконтроллерах AVR

Цель работы:

- изучение системы прерываний микроконтроллеров AVR,
- освоение системы команд микроконтроллеров AVR,
- ознакомление с работой стека при вызове подпрограмм и обработчиков прерываний,
- программирование внешних прерываний.

Введение

Контроллер прерываний обрабатывает внешние прерывания и прерывания от периферийных устройств микроконтроллера (таймеров, портов последовательного ввода-вывода, аналогового компаратора и др.). Все прерывания являются маскируемыми. Адреса и маски прерываний указаны в табл.1. В ней использованы обозначения:

- INT0, INT1 – сигналы внешних прерываний, поступающие по линиям порта PD2, PD3. Маски внешних прерываний представлены разрядами INT0 и INT1 (6-й и 7-й разряды соответственно регистра GIMSK микроконтроллера AT90S8515 или регистра GICR микроконтроллера ATmega8515). Сигналы внешних прерываний устанавливают в «1» флаги прерываний INTF0 и INTF1 (6-й и 7-й разряды соответственно регистра GIFR);

- прерывания от таймеров T1, T0 имеют адреса с \$003 по \$007. Маскирование прерываний от таймеров осуществляется битами регистра TIMSK;

- последующие адреса прерываний для запросов от последовательных каналов ввода – вывода SPI, UART и аналогового компаратора;

- остальные прерывания (INT2, T/C0 COMP и др.), помеченные знаком «*», поддерживаются только микроконтроллером ATmega8515.

Из табл. 1 видно, что таблица векторов прерываний моделей ATx8515 располагается в начальной области памяти программ, начиная с адреса \$0001. Размер таблицы зависит от типа модели. Следует также иметь в виду, что микроконтроллеры семейства Mega допускают перемещение таблицы векторов из начальной области памяти программ в начало области загрузчика, которая располагается в конце программной памяти. Для этого используют два младших бита регистра GICR микроконтроллеров ATmega8515: IVSEL (1-й разряд) и IVCE (0-й разряд). Если IVSEL=0, таблица векторов располагается в начале памяти программ, в противном случае, при IVSEL=1, – в начале области загрузчика. Конкретное значение начального адреса области загрузчика определяется конфигурационными ячейками BOOTSZ1 и BOOTSZ2. Бит IVCE используется для разрешения изменения бита IVSEL.

Таблица 1. Адреса и маски прерываний ATx8515

Запрос	Адрес	Маска	Флаг
RESET	000	–	–
INT0	001	GIMSK.6/ GICR.6	GIFR.6
INT1	002	GIMSK.7/ GICR.7	GIFR.7
T/C1 CAPT	003	TIMSK.3	TIFR.3
T/C1 COMPA	004	TIMSK.6	TIFR.6
T/C1 COMPB	005	TIMSK.5	TIFR.5
T/C1 OVF	006	TIMSK.7	TIFR.7
T/C0 OVF	007	TIMSK.1	TIFR.1
SPI STC	008	SPCR.7	SPSR.7
UART RXC	009	UCR.7/ UCRB.7	USR.7/ UCSRA.7
UART DRE	00A	UCR.5/ UCSRB.5	USR.5/ UCSRA.5
UART TXC	00B	UCR.6/ UCSRB.6	USR.6/ UCSRA.6
ANA COMP	00C	ACSR.3	ACSR.4
INT2*	00D	GICR.5	GIFR.5
T/C0 COMP*	00E	TIMSK.0	TIFR.0

Положение вектора в таблице прерываний определяет приоритет соответствующего прерывания. Запрос с меньшим адресом имеет более высокий приоритет. Флаг общего прерывания I расположен в регистре состояния микроконтроллера SREG (бит 7). При I=1 и единичном значении маски запроса прерывание данного типа разрешено. При поступлении запроса устанавливается флаг прерывания в одном из регистров ввода – вывода, который может вызвать аппаратное прерывание. Состояние флага может быть также опрошено программой.

Обработка прерывания начинается после завершения текущей команды, для чего может понадобиться несколько тактов в зависимости от типа выполняемой команды. При обработке прерывания разряд I в регистре SREG сбрасывается в состояние 0, запрещая обработку всех остальных запросов. В стеке сохраняется адрес возврата и выполняется переход по вектору прерывания на первую команду обработчика прерывания. При выходе из прерывающей программы разряд I вновь устанавливается в состояние 1, разрешая обработку прерываний. Программа, выполняемая при пуске микроконтроллера и использующая вектор запроса RESET,

не зависит от состояния разряда I. Для обработки прерываний она должна выполнить команду разрешения прерывания SEI, устанавливающую флаг I в состояние 1.

Обработка внешних прерываний

В качестве входов внешних прерываний используются входы портов с альтернативной функцией. Это входы PD2, PD3 для прерываний INT0, INT1 и вход PE0 для прерывания INT2 в микроконтроллере ATmega8515. Запросы внешних прерываний INT0, INT1 могут быть представлены низким уровнем сигнала прерывания (L), переходом от высокого уровня сигнала к низкому (HL - по отрицательному фронту), переходом от низкого уровня сигнала к высокому (LH - по положительному фронту), запрос INT2 только переходами (LH) и (HL). В зависимости от типа запроса в регистре управления микроконтроллера MCUCR необходимо установить биты ISCx0 и ISCx1 согласно табл.2 для каждого из прерываний INTx (x=0,1) и определить бит ISC2 в регистре EMCUCR для прерывания INT2. При ISC2=0 прерывание осуществляется по отрицательному фронту, при ISC2=1 – по положительному фронту.

Таблица 2. Таблица выбора типа запроса

ISCx1	ISCx0	Тип запроса
0	0	L
0	1	-
1	0	HL
1	1	LH

Ниже приведена программа переключения светодиодов с использованием внешнего прерывания от кнопки STOP.

Согласно поставленным требованиям в блок инициализации микроконтроллера по сравнению с программой 1.1 (лабораторная работа 1) внесён ряд изменений:

- добавлен вектор прерываний по адресу \$001;
- указатель стека установлен на последнюю ячейку ОЗУ;
- разрешено внешнее прерывание INT0 (по сигналу «0» на линии порта PD2) и прерывания вообще.

Так как внешнее прерывание INT0 представлено сигналом на входе порта PD2, в качестве кнопки STOP используем кнопку SW2 и программируем PD2 на ввод. Пример программы приведен ниже. Задержка представлена подпрограммой DELAY. Программа работает аналогично программе 1.1, но нажатие кнопки STOP вызывает прерывание.

Программа 2.1

```
; *****
;Программа 2.1 для поочерёдного переключения светодиодов (СД)
;при нажатии на кнопку START (SW0).
```

```

;После нажатия на кнопку STOP (SW2) переключение прекращается и
;возобновляется с места остановки при повторном нажатии
;на кнопку START.
;Соединения: SW0-PD0, SW2-PD2, LED-PB
;*****
;.include "8515def.inc"           ;файл определений для AT90S8515
;.include "m8515def.inc"         ;файл определений для ATmega8515
.def temp = r16                  ;временный регистр
.def reg_led = r20               ;состояние регистра светодиодов
.equ START = 0                   ;0-ой вывод порта PD

.org $000
;***Векторы прерываний***
rjmp INIT                        ;обработка сброса
rjmp STOP_PRESSED               ;обработка внешнего прерывания INT0 (STOP)
;***Инициализация МК***
INIT:    ldi reg_led,0xFE
         ldi temp,$5F           ;установка
         out SPL,temp           ; указателя стека
         ldi temp,$02           ; на последнюю
         out SPH,temp           ; ячейку ОЗУ
         sec                    ;C=1
         set                    ;T=1
         ser temp               ;инициализация выводов
         out DDRB,temp          ; порта PB на вывод
         out PORTB,temp         ;погасить СД
         clr temp               ;инициализация 0-ого и 2-ого выводов
         out DDRD,temp          ; порта PD на ввод
         ldi temp,0x05          ;включение 'подтягивающих'
         out PORTD,temp         ; резисторов порта PD
         ldi temp,(1<<INT0)     ;разрешение прерывания INT0
         out GICR,temp          ; (6 бит GICR или GIMSK)
         ldi temp,0x00          ;обработка прерывания INT0
         out MCUCR,temp         ; по низкому уровню
         sei                    ;глобальное разрешение прерываний
WAITSTART: sbic PIND,START      ; ожидание нажатия
         rjmp WAITSTART        ; кнопки START
LOOP:    out PORTB,reg_led      ;включение СД

```

```

    rcall DELAY          ;задержка
    ser temp             ;выключение
    out PORTB,temp       ; светодиода
    brts LEFT            ;переход, если флаг T установлен
    sbrs reg_led,0        ;пропуск следующей команды,
                        ; если 0-й разряд reg_led установлен

    set                  ;T=1
    ror reg_led           ;сдвиг reg_led вправо на 1 разряд
    rjmp LOOP
LEFT:    sbrs reg_led,7    ;пропуск следующей команды,
                        ;если 7-й разряд reg_led установлен

    clt                  ;T=0
    rol reg_led          ;сдвиг reg_led влево на 1 разряд
    rjmp LOOP

;***Обработка прерывания от кнопки STOP***
STOP_PRESSED:
WAITSTART_2:                ;ожидание
    sbic PIND,START        ; нажатия
    rjmp WAITSTART_2      ; кнопки START
    reti

;*** Задержка ***
DELAY:    ldi r17,8
d1:       ldi r18,255
d2:       ldi r19,255
d3:       dec r19
          brne d3
          dec r18
          brne d2
          dec r17
          brne d1
          ret

```

Практическая часть

Задание 1. Проверить работу вышеприведенной программы в шаговом режиме работы, наблюдая и фиксируя состояния программного счетчика Program Counter, указателя стека Stack Pointer (на вкладке I/O View, структура Processor) и ячеек области стека (в окне памяти данных, начиная с адреса \$025F), а также сопоставляя загружаемые адреса с адресами выполняемых

команд в окне дизассемблера, при вызове подпрограммы задержки DELAY и обработчика прерывания путем замыкания кнопки STOP в регистре PIND.

Проанализировать работу стека для следующих случаев:

- а) при вызове подпрограммы задержки после замыкания кнопки START,
- б) эмулируя вызов обработчика прерывания путем замыкания кнопки STOP до вызова подпрограммы DELAY,
- в) эмулируя прерывание после входа в подпрограмму задержки.

Для все трех случаев снять скриншоты состояний стека, программного счетчика, указателя стека и фрагменты программы в окне дизассемблера с адресами и мнемониками команд в точках прерываний и при вызове DELAY. Объяснить наблюдаемые результаты.

Восстановить программу для работы в реальном времени на плате STK500. После трансляции загрузить ее в микроконтроллер и проверить ее работу на плате.

Задание 2. Изменить программу 2.1, добавив второе внешнее прерывание INT1 (вывод PD3) от кнопки START, удалив прежнюю связь кнопки с выводом PD0. Из табл. 1 самостоятельно выбрать адрес вектора прерывания и бит регистра маски прерываний в соответствии с используемым микроконтроллером.

Отладить программу с помощью симулятора, подготовить ее для работы в реальном времени на плате STK500, загрузить в микроконтроллер и проверить ее работу.

Задание 3. Изменить программу для микроконтроллера ATmega8515, заменив прерывание INT0 на INT2.

Задание 4. Запустив программу Proteus ISIS, собрать проект, включающий микроконтроллер, 2 кнопки и 8 светодиодов.

Выбрав из библиотеки компонентов Component/PickDevices/Micro микроконтроллер ATx8515, в окно редактора вводим микроконтроллер. Добавляем периферийные устройства: кнопки (Button) из библиотеки компонентов Component/Active и светодиоды (Led или Led-Bargraph). Присоединяем выводы кнопки: один - к выводу порта PDx, второй – к общей шине GROUND, выбрав ее из списка Inter-sheet Terminal на панели инструментов слева. Сохранить проект на диске в папке с программой.

С помощью команд меню Source/(Add/Remove Source files...) добавляем в проект файл с расширением *.asm, указав используемый транслятор (AVRASM2). Выполняем компиляцию с помощью команды Build all (Компоновать все).

В меню *Debug* запустить проект на выполнение (*Start* и затем *Execute*) и проверить его работу, поочередно нажимая кнопки START и STOP.

Оформление отчета

Отчет должен содержать:

- а) базовую программу (2.1) с одним внешним прерыванием;
- б) скриншоты стековой области памяти данных (Data), указателя стека, программного счетчика при вызове подпрограммы задержки и обработчика прерываний с объяснением наблюдаемых результатов;
- в) подготовленную программу для обработки 2-х внешних прерываний;
- г) схемы алгоритмов основной программы, обработчиков прерываний и подпрограммы задержки;
- д) схему проекта («скриншот» из окна редактора Proteus) с текущим состоянием линейки светодиодов.
- е) выводы по работе.

Требования при защите: уметь отвечать на вопросы по общей обработке прерываний, текстам приведенных в отчете программ, по работе стека, отвечать на контрольные вопросы.

Контрольные вопросы

1. Какая линия порта микроконтроллера АТМega8515 используется для внешнего прерывания INT2? По какому адресу в таблице прерываний располагается вектор прерывания INT2?
2. Охарактеризуйте каждое из внешних прерываний?
3. Какие команды и регистры нужно задействовать для управления внешними прерываниями?
4. Какое из 3-х внешнее прерывание имеет более высокий приоритет?
5. Как изменится схема проекта и программа обработки прерываний в случае подключения обоих кнопок к одному входу прерывания?

Задачи для самостоятельного программирования при подготовке к рубежному контролю

1. Написать и промоделировать (в среде Proteus или VMLab) программу, которая при нажатии кнопки SW_i, подключенной к выводу порта P_x, кратковременно включает светодиод LED_j на 40 мс.
2. Написать и промоделировать программу, которая при первом нажатии кнопки SW_i включает, а при повторном нажатии выключает светодиод LED_j.
3. Промоделировать работу выходного порта микроконтроллера:
 - а) в режиме двоичного счета, б) в режиме двоично-десятичного счета,

в) в режиме обратного двоичного счета, г) в режиме обратного двоично-десятичного счета.

4. Вывести на семисегментный индикатор последовательность десятичных цифр 0, 1, 2,...9.

Проверить работу программы в среде Proteus или VMLab.

5. Вывести на семисегментный индикатор номер нажатой кнопки.

6. Вывести на светодиоды LED0 – LED3 двоичный код номера нажатой кнопки. Проверить работу программы в симуляторе. В окне редактора присоединить к выходному порту микроконтроллера 4-входовой семисегментный индикатор с встроенным преобразователем 4-разрядного кода в семиразрядный. Наблюдать работу схемы, поочередно нажимая каждую из восьми кнопок.

7. Подобрать необходимую частоту регенерации, при которой последовательно включаемые светодиоды группы из 4 светодиодов (в позициях цифры 4 одного семисегментного индикатора) будут восприниматься глазом так же, как по команде одновременного включения этих светодиодов. Проверить работу программы с помощью симулятора Proteus или VMLab.

8. Написать и промоделировать в Proteus программу “Кодового Замка”, которая после ввода 4-х разрядного двоичного PIN-кода с помощью кнопок SW0 - SW3 осуществляет сравнение с заданным PIN-кодом (номером студента в журнале группы) и включает один светодиод при правильном вводе. При 3-х неправильных попытках введения PIN-кода кнопочная клавиатура должна быть заблокирована, а все светодиоды включены.

Рекомендуемая литература

1. В.Я. Хартов Микроконтроллеры AVR. Практикум для начинающих. Издательство МГТУ им. Н.Э. Баумана. 2012 г.