Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

Кафедра «Электронные вычислительные машины»

ОТЧЕТ

По лабораторная работа №5

**Обработка прерываний. Внешние прерывания**

По дисциплине «ЭВМ и периферийные устройства»

Выполнили:

Проверили:

Рязань 20

**Цель:** изучить процессы прерываний и их виды, получить навыки работы с

системой прерываний на примере микроконтроллера AVR ATmega328P.

Таблица 1 – Трасса программы из примера 1. начало

|  |  |  |  |
| --- | --- | --- | --- |
| № | Строки программы | Расшифровка | Результат |
| 1 | .include"m328Pdef.inc" | Подключение библиотеки с определениями для МК Atmega328p |  |
| 2 | .org 0 | Определение вектора прерывания с 0 адресом |  |
| 3 | jmp Reset | Переход к метке Reset | При сбросе - переход к Reset |
| 4 | .org 0x006 | Определение вектора прерывания с 0x006 адресом |  |
| 5 | jmp mPCINT0 | Переход к метке mPCINT0 | При прерывании PCINT0 осуществляется переход к mPCINT0 |
| 6 | .org 0x00A | Определение вектора прерывания с 0x00A адресом |  |
| 7 | jmp mPCINT2 | Переход к метке mPCINT2 | При прерывании  PCINT2  осуществляется переход к mPCINT2 |
| 8 | Reset: |  |  |
| 9 | ldi r16,low(RAMEND) | Запись в r16 младшего бита RAMEND | r16:=low(RAMEND) |
| 10 | out spl, r16 | Запись в младший бит sp значения из r16 | spl:=low(RAMEND) |
| 11 | ldi r16,high(RAMEND) | Запись в r16 старшего бита RAMEND | r16:=high(RAMEND) |
| 12 | out sph, r16 | Запись в старший бит sp значения из r16 | spl:=high(RAMEND) |
| 13 | ldi r16, 0x07 | Запись в r16 0x07 | r16:=0x07 |
| 14 | sts PCICR, r16 | Сохранение в PCICR значения из r16 | PCICR:=0x07 |
| 15 | ldi r16, 0xFF | Запись в r16 0xFF | r16:=0xFF |
| 16 | sts PCMSK0, r16 | Сохранение в PCMSK0 значения из r16 | PCIMSK0:=0xFF |
| 17 | sts PCMSK2, r16 | Сохранение в PCMSK2 значения из r16 | PCIMSK2:=0xFF |
| 18 | sei | Разрешение прерываний | Бит прерываний в sreg устанавливается в 1 |
| 19 | m: inc r17 | Инкремент значения в r17 | r17:=r17+1 |
| 20 | jmp m | Переход на метку m |  |
| 21 | mPCINT0: |  |  |
| 22 | in r20,PINB | Запись из порта PINB в r20 | r20:=PINB |
| 23 | Mov r22,r18 | Копирование из r18 в r22 | r22:=r18 |
| 24 | eor r22,r20 | r22=r22 xor r20 | r22=r22 xor r20 |

Таблица 1 – Трасса программы из примера 1. завершение

|  |  |  |  |
| --- | --- | --- | --- |
| 25 | mov r18,r20 | Копирование из r20 в r18 | r18:=r20 |
| 26 | reti | Возврат из прерывания |  |
| 27 | mPCINT2: |  |  |
| 28 | in r21,PIND | Запись из порта PIND в r21 | r21:=PIND |
| 29 | mov r23,r19 | Копирование из r19 в r23 | r23:=r19 |
| 30 | eor r23,r21 | r23= r23 xor r21 | r23=r23 xor r21 |
| 31 | mov r19,r21 | Копирование из r21 в r19 | r19:=r21 |
| 32 | reti | Возврат из прерывания |  |

момент срабатывания прерывания на снимке экрана при работе Saleae Logic на рисунках 1,2.



Рисунок 1 – Нажатие на кнопку



Рисунок 2 – Отпускание кнопки

**Задание 1.**

Измените код из примера таким образом, чтобы система реагировала на прерывание INT0 переключением между двумя светодиодами (при нажатии на кнопку один диод зажигается, второй гаснет, при следующем нажатии наоборот). Для работы используйте светодиоды RX – PD0, TX – PD1.

Подготовьте программы для работы в режиме срабатывания прерывания по

уровню и по фронту. Проанализируйте их работу в отладчике и отметьте разницу в отчёте. Переключите кнопку из PD3 в PD2 (используйте рис. 2). Поочерёдно загрузите программы в контроллер и проанализируйте работу.

Коды программ:

Таблица 2 – Задание 1. начало

|  |  |
| --- | --- |
| По уровню | По фронту |
| .include"m328Pdef.inc"  .org 0  jmp Reset  .org 0x004  jmp Int\_1  Reset:  ldi r16, High(RAMEND) | .include"m328Pdef.inc"  .org 0  jmp Reset  .org 0x004  jmp Int\_1  Reset:  ldi r16, High(RAMEND) |

Таблица 2 – Задание 1. завершение

|  |  |
| --- | --- |
| out sph, r16  ldi r16, Low(RAMEND)  out spl, r16  ldi r16, 0x04 // 01 срабатывание по любому изменению уровня сигнала  sts EICRA, r16  ldi r16, 0x02// 01 прерывание int1  out EIMSK, r16  sbi ddrd, 0 // 0-ой бит на вывод  sbi ddrd, 1 // 1 ой бит на вывод  sbi portd, 3 // установка 2ого бита  ldi r20, 0x09  ldi r21, 0x03  sbi portd, 0  sei  main: jmp main  Int\_1:  in r22, pind  andi r22, 0b100  cpi r22, 0  brne m1  //cpi  eor r20, r21  out portd,r20  m1:  reti | out sph, r16  ldi r16, Low(RAMEND)  out spl, r16  ldi r16, 0x0c // 11 срабатывание по фронту  sts EICRA, r16  ldi r16, 0x02// 01 прерывание int0  out EIMSK, r16  sbi ddrd, 0 // 0-ой бит на вывод  sbi ddrd, 1 // 1 ой бит на вывод  sbi portd, 3 // установка 2ого бита  ldi r20, 0x09  ldi r21, 0x03  sbi portd, 0  sei  main: jmp main  Int\_1:  eor r20, r21  out portd,r20  reti |

При прерывании по фронту прерывание срабатывает при смене сигнала с 0 на 1, при прерывании по уровню прерывание происходит по любому изменению уровня сигнала, т.е. и в момент нажатия на кнопку и в тот момент, когда нажатие прекратилось.

**Задание 2.** Измените программу таким образом, чтобы система реагировала

на прерывания INT0 и INT1 по фронту импульса, изменяя состояние светодиодов TX (в обработчике INT0) и RX (в обработчике INT1) на противоположное. Запустите программу в отладчике. Вызовите прерывания INT0 и INT1) одновременно. Проанализируйте приоритет прерываний. Загрузите программу в контроллер и проанализируйте её работу.

Код программы:

.include"m328Pdef.inc"

.org 0

jmp Reset

.org 0x002

jmp Int\_0

.org 0x004

jmp Int\_1

Reset:

ldi r16, High(RAMEND)

out sph, r16

ldi r16, Low(RAMEND)

out spl, r16

ldi r16, 0x0f // 1111 срабатывание по фронту

sts EICRA, r16

ldi r16, 0x03// 11 прерывание int0 и int1

out EIMSK, r16

sbi ddrd, 0 // 0-ой бит на вывод

sbi ddrd, 1 // 1 ой бит на вывод

sbi portd, 2 // установка 2 бита

sbi portd, 3 // установка 3 бита

ldi r21, 0x01

ldi r22, 0x02

sbi portd, 0

sbi portd, 1

sei

main: jmp main

Int\_0:

in r20, portd

eor r20, r21

out portd,r20

reti

Int\_1:

in r20, portd

eor r20, r22

out portd,r20

Reti

Более приоритетным прерыванием является INT0 (PD2). для того чтобы прога работала корректно нажатия на кнопки должны быть достаточно длительными и не одновременными, иначе будет срабатывать только более приоритетное прерывание.

**Задание 3.** Измените код программы из примера 2 (на языке C), оставив

разрешёнными прерывания только от указанных в таблице выводов контроллера. Обработчики прерываний измените таким образом, чтобы на вывод PORTB5 (светодиод L) выводилось текущее значение старшего бита счётчика count и сохранялось до следующего прерывания.

Вариант 9: PB5, PD3.

Так как по заданию нам требуется совершать прерывания по PB5, т.е. установить этот бит порта на ввод, мы не сможем выводить через этот бит информацию на светодиод. Исходя из этого было решено установить PB5 на вывод.

Код задания 3 на C версия 1:

|  |  |
| --- | --- |
| #include <avr/io.h>  #include <avr/interrupt.h>  char x, count, outt;  ISR(PCINT0\_vect)  {  char h = (count & 0b10000000) >> 3;  outt=PORTB&0xdf;  outt|=h;  PORTB = outt;  }  ISR(PCINT2\_vect)  {  char h = (count & 0b10000000) >> 3;  outt=PORTB&0xdf;  outt|=h;  PORTB = outt;  }  int main(void)  {  x=0;  DDRB = 0x20;  DDRD = 0x00;  DDRB |= 0x20;  DDRD |= 0x08;  PCICR = 0x05;    PCMSK0 = 0x20;  PCMSK2 = 0x08;  sei();  while (1)  {  PORTC = count;  count++;  }  } | //подключение шапки  //прерывание по вектору 0  //Старший бит счетчика в 5 бит h  //устанавливаем в 5 бит 0  //устанваливаем в 5 бит h  //прерывание по вектору 2  //Старший бит счетчика в 5 бит h  //устанавливаем в 5 бит 0  //устанваливаем в 5 бит h  //тк PB это светодиод и нам надо выводить его значение на логику  //чтение бита из порта PD3  //поддтяжка вверх  //поддтяжка вверх  //разрешение прерываний (разрешены группы прерываний 0 и 2 B и D)  //разрешено только PCINT0 (PB5)  //разрешено только PCINT2 (PD3)  //разрешение прерываний  //увеличение счётчика count |

Так же был выполнен код для варианта: PB1, PD3.

|  |  |
| --- | --- |
| #include <avr/io.h>  #include <avr/interrupt.h>  char x, count, outt;  ISR(PCINT0\_vect)  {  char h = (count & 0b10000000) >> 3;  outt=PORTB&0xdf;  outt|=h;  PORTB = outt;  }  ISR(PCINT2\_vect)  {  char h = (count & 0b10000000) >> 3;  outt=PORTB&0xdf;  outt|=h;  PORTB = outt;  }  int main(void)  {  x=0;  DDRB = 0x20;  DDRD = 0x00;  DDRB |= 0x02;  DDRD |= 0x08;  PCICR = 0x05;  PCMSK0 = 0x02;    PCMSK2 = 0x08;  sei();  while (1)  {  PORTC = count;  count++;  }  } | //подключение шапки  //прерывание по вектору 0  //Старший бит счетчика в 5 бит h  //устанавливаем в 5 бит 0  //устанваливаем в 5 бит h  //прерывание по вектору 2  //Старший бит счетчика в 5 бит h  //устанавливаем в 5 бит 0  //устанваливаем в 5 бит h  //тк PB это светодиод и нам надо выводить его значение на логику  //чтение бита из порта PD3  //поддтяжка вверх  //поддтяжка вверх  //разрешение прерываний (разрешены группы прерываний 0 и 2 B и D)  //разрешено только PCINT0 (PB1) (PB1 вместо PB5, потому что PB5 и нам надо его выводить)  //разрешено только PCINT2 (PD3)  //разрешение прерываний |

Заключение: изучили процессы прерываний и их виды, получили навыки работы с системой прерываний на примере микроконтроллера AVR ATmega328P.