Лабораторная работа №6

Тема: ПРЕРЫВАНИЯ

Цель: разработать программное обеспечение использовав систему аппаратных прерываний.

теоретические сведения

Прерывание - аппаратная остановка выполнения кода для выполнения отдельной процедуры, связанной с конкретным событием.

Описание регистров таймера и процессору находится в разделе дополнительной литературы.

Пример кода с использованием прерываний от таймеров:

/ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*

подключение библиотек

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\* /

#include <avr / io.h>

#include <avr / interrupt.h>

/ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*

Переопределения обработчиков прерываний

Для облегчения использовано макрос ISR ()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\* /

// Переполнения счетчика timer1

ISR (TIMER1\_OVF\_vect) {

    // Переключение второй лапки порта на противоположное состояние

    PORTD = PORTD ^ 0x02;

}

// Переполнение счетчика timer0

ISR (TIMER0\_OVF\_vect) {

    // Переключение первой лапки порта в противоположное состояние

    PORTA = PORTA ^ 0x01;

}

/ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*

Главная функция Main

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\* /

int main (void) {

    // Порт настраиваем на вывод напряжения

    DDRD = 0xFF;

    PORTD = 0xFF;

    // Включаем прерывания по переполнению таймеров 1 и 0

    TIMSK = (1 << TOIE0) | (1 << TOIE1)

    // Счетчик timer0 ставим в начальное значение 0

    TCNT0 = 0x00;

    // Старт счетчика timer0 с делением основной

    // Тактовой частоты на 1024

    TCCR0 = (1 << CS02) | (1 << CS00)

    // Пусть 16 бит таймер тоже имеет преддильник 1024

    TCCR1B | = (1 << CS10) | (1 << CS12)

    // Позволяется прерывания: флаг I = 1

    sei ();

    while (true) {

      // Во время между прерываниями можно переходить в режим

      // Энергосбережения

      // Sleep ();

    }

}

#include <avr/io.h>

#include <avr/interrupt.h>

char temp = 0;

// прерывание первой кнопки

ISR(INT0\_vect)

{

temp++;

OCR0 = temp;// инкремент

}

// прерывание второй кнопки

ISR(INT1\_vect)

{

temp--;

OCR0 = temp;// deкремент

}

// прерывание по совпадению

ISR(TIMER0\_COMP\_vect) { PORTA |=(1<<0); }

// прерывание по переполнению

ISR(TIMER0\_OVF\_vect) { PORTA &=~(1<<0); }

int main(void)

{

DDRA |=(1<<0); // порт на выход

OCR0 = 0;

TCCR0 = 0b01001011; // 0-2 предделитль 64 6-3 бит работа ШИМа

TIMSK |= (1<<1); // разрешение уходить в вектор при совпадении

TIMSK |= (1<<0); // разрешение уходить в вектор при переполнении

MCUCR &=~(1<<0); // установка значений для прерывания

MCUCR &=~(1<<0); // по низкому уровню

GICR |=(1<<6); // разрешение прерывания INT0

GICR |=(1<<7); // разрешение прерывания INT1

asm ("sei"); // разрешение глобального прерывания

while (1)

{

asm("nop");

}

}

Контрольные вопросы:

1. Как настроить частоту счетчика таймера?
2. Как задать максимальное значение счетчика таймера, после которого счет начинается с нуля?
3. Как переключить таймер в режим работы генератора импульсов?
4. Как задать ширину импульсов?
5. Где используется широтно-импульсная модуляция?

Задание:

1. Разработать программу управления яркостью светодиодов с использованием прерываний от таймера. Дополнительное задание: реализовать опрос кнопок для ручного регулирования яркости светодиода.

2. Разработайте блок-схему алгоритма работы.

3. Создайте программное обеспечение, и проверьте работу программы на симуляторе.

4. Ответьте на контрольные вопросы.

(Полный перечень ассемблерных команд процессора предоставлено в файле atmega8.pdf на сервере с материалами к предмету).

Контрольные вопросы:

1. Как разрешить прерывание от таймера по переполнению?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Для какого таймеру можно ограничить верхнее значение счетчика?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Можно комбинировать архитектуру циклического изменения задач и архитектуру работы по прерыванием?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Придумайте задачу, где поочередное выполнение двух задач является более выгодным, чем прерывание?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Придумайте задачу, где архитектура работы по прерываниям имеет преимущества за архитектуру поочередного выполнения?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_