

УДК 004.451

## ОРГАНИЗАЦИЯ МНОГОЗАДАЧНОСТИ НА МИКРОКОНТРОЛЛЕРЕ ФИРМЫ ATMEL

**Львов С.А., Кратинов А.Г., Маркин Н.А., Волошенко А.Н.**

Восточноукраинский национальный университет имени Владимира Даля  
кафедра автоматизации и компьютерно-интегрированных технологий

E-mail: [kratinov@bigmir.net](mailto:kratinov@bigmir.net)

### *Аннотация*

*Львов С.А., Кратинов А.Г., Маркин Н.А., Волошенко А.Н. Организация многозадачности на микроконтроллере фирмы Atmel. Рассмотрены принципы построения программ для операционных систем реального времени, определены ограничения по структуре исходного кода, даны рекомендации по выбору аппаратных средств микроконтроллера, необходимых для функционирования ОСРВ, а также приведены примеры программной реализации исходного кода программы на языке С.*

### **Общая постановка проблемы**

Начиная с 1980-х годов микроконтроллеры стали внедряться в управление различными устройствами, постепенно вытесняя ТТЛ-логику. Сегодня микроконтроллерным управлением оснащено большинство выпускаемых устройств бытового и промышленного назначения, а количество задач, которые микроконтроллер должен решать параллельно, постоянно увеличивается. Вследствие этого управляющие программы, с точки зрения написания и реализации, становятся все сложнее. В то же время, проблема реализации многозадачности на микроконтроллерах на сегодняшний день остается исследованной не достаточно.

### **Анализ последних исследований и публикаций**

Реализация многозадачности связана с применением операционных систем реального времени, анализу которых посвящен ряд работ [1,2,3,4]. Однако в опубликованных работах не отражены особенности использования операционных систем в микроконтроллерах фирмы Atmel.

**Цель исследования** – реализация практического использования ОСРВ на микроконтроллере Atmega16 фирмы Atmel для достижения многопоточковой обработки данных.

### **Изложение основного материала исследований**

Операционная система реального времени, ОСРВ (англ. Real-Time Operating System) — тип операционной системы, которая реагирует в предсказуемое время на непредсказуемое появление внешних событий.

Любая программа, которая строится на операционной системе реального времени, должна быть разбита на отдельные процессы (задачи).

Задача - это часть программного кода, которая с точки зрения программиста выполняется одновременно с другими задачами, что обеспечивается разделением процессорного времени между ними. Каждая задача может быть представлена как независимое приложение, которое владеет уникальными ресурсами (регистры процессора, указатель стека и т.п.). В простейшем случае программа может содержать всего одну задачу,

например, опрос датчика, однако, в большинстве случаев программа включает в себя несколько задач, например, опрос датчика, опрос клавиатуры и индикация показаний.

К каждой задаче в ОСРВ выдвигается ряд требований, так любая задача в ОСРВ OSA [4] - это функция, тело которой содержит бесконечный цикл. Внутри цикла должен быть хотя бы один вызов сервиса ОС, который переключает контекст задач, иначе планировщик (а следовательно, и все остальные задачи) никогда не получит управление обратно.

Простейшая задача может выглядеть так:

```
void SimpleTask (void)
{
    for (;;)
    {
        OS_Yield();
    }
}
```

Или так:

```
void SimpleTask (void)
{
    for (;;)
    {
        OS_Delay(1);
    }
}
```

Задача может находиться в одном из 5-и состояний:

1. Не создана (не активна).
2. В ожидании.
3. В готовности.
4. Выполняется.
5. Приостановлена.

Кроме того, в ОСРВ OSA задача имеет приоритет от 0 (высший) до 7 (низший). Приоритет можно изменять в ходе выполнения программы.

Для примера возьмем микроконтроллер фирмы Atmel Atmega16. Микроконтроллер имеет богатую периферию и может выполнять до 16 миллионов операций в секунду.

Для запуска ОСРВ OSA потребуется лишь выделить некоторый объем памяти для хранения операционной системы. Точное значение требуемой памяти зависит от количества задач, которые будет обрабатывать ОСРВ в процессе своей работы, и включению дополнительных функций ОС.

Для формирования задержек (пауз) ОС требуется использование аппаратного таймера микроконтроллера. При срабатывании прерывания, вызванного переполнением таймера, операционная система отсчитывает так называемый «системный тик» (все задержки в ОСРВ OSA формируются именно при помощи системных тиков). В микроконтроллере Atmega16 за работу любого таймера отвечают 2 регистра TCNTX и TCCRХ, а также регистр TIMSK, отвечающий за разрешение прерываний. Например, микроконтроллер тактируется от внешнего кварцевого резонатора с частотой 16 мГц, предделитель таймера равен 256 (т.е. таймер учитывает лишь каждый 256 тактирующий импульс), начальное значение счетчика равно 193 (т.е. таймер ведет счет не от 0 до 255, а начиная со 193). Такие настройки позволят вызывать прерывания от переполнения таймера примерно каждую мс (0,000996 сек.), т.е. системный тик ОСРВ будет составлять 1 мс. Для формирования системных тиков ОС

рекомендуется использовать восьмибитный таймер.

Для примера практического использования операционной системы OSA на микроконтроллере Atmega16 представим, что есть задача снимать показания с 8 разных датчиков (опрашивать 8 каналов АЦП), выводить показания на семисегментный индикатор с динамической индикацией и опрашивать клавиатуру (2 кнопки, отвечающие за переключение активного датчика и кнопка старт/стоп измерений). В данной программе можно выделить 3 задачи:

1. Опрос АЦП
2. Вывод индикации на семисегментный индикатора
3. Опрос клавиатуры

Код задачи, отвечающей за опрос АЦП:

```
void get_adc_result(void)
{
    for(;;)
    {
        ADMUX=adc_chanel;
        ADCSRA|=64;
        while(!(ADCSRA&16));
        adc_result=ADCH;
        OS_Yield();
    }
}
```

Код задачи, отвечающей за вывод результатов на индикатор:

```
void light_data (void)
{
    for ( ;; )
    {
        PORTA=ind[adc_result/100];
        PORTD|=ind1;
        OS_Delay(5);
        PORTD&=~ind1;
        PORTA=ind[(adc_result%100)/10];
        PORTD|=ind2;
        OS_Delay(5);
        PORTD&=~ind2;
        PORTA=ind[adc_resul%10];
        PORTD|=ind3;
        OS_Delay(5);
        PORTD&=~ind3;
    }
}
```

Код задачи, отвечающей за опрос клавиатуры:

```
void buttons (void)
{
    for ( ;; )
    {
        if (!(PINC&(1<<0))) if (OS_Task_IsPaused(TaskPointer)) OS_Task_Pause(TaskPointer) else
        OS_Task_Continue(TaskPointer);
        if (!(PINC&(1<<1))) if (adc_chanel<=6) adc_chanel++; else adc_chanel=0;
    }
}
```

```
    if(!(PINC&(1<<2))) if (adc_chanel>=1) adc_chanel--; else adc_chanel=7;
    OS_Yield();
}
}
```

Код главного тела программы, содержащий лишь функции ОСРВ (код для настройки периферии опущен):

```
void main (void)
{
    OS_Init();
    OS_Task_Create(3, get_adc_result);
    OS_Task_Create(3, light_data);
    OS_Task_Create(3, buttons);
    OS_EI();
    OS_Run();
}
```

### Выводы

1. Операционная система реального времени позволяет программисту сосредоточить свои усилия на решении конкретных задач (алгоритмических, математических и т.п.), не отвлекаясь на задачи второстепенные. Она берет на себя:

- переключение между параллельными процессами (например, опрос клавиатуры, вывод информации на экран и т.п.);
- отсчет таймаутов, выдержку задержек;
- выбор готовой к выполнению задачи с наивысшим приоритетом и передача ей управления;

- обмен данными между задачами (с помощью семафоров, сообщений и пр.).

2. В результате исследования предложена операционная система OSA, которая показала устойчивую работу на микроконтроллере фирмы Atmel, однако, для ее использования следует соблюдать некоторые ограничения, например: задача в ОСРВ OSA является обычной функцией, тело которой должно содержать бесконечный цикл, внутри которого должен быть хотя бы один вызов сервиса переключения задач.

### Список литературы

1. Операционная система реального времени [Электронный ресурс] - URL: [http://ru.wikipedia.org/wiki/Операционная\\_система\\_реального\\_времени](http://ru.wikipedia.org/wiki/Операционная_система_реального_времени). Дата обращения: 10.03.12 г.

2. OSA: Документация [Электронный ресурс] – URL: <http://www.pic24.ru/doku.php/osa/ref/intro>. Дата обращения: 12.03.12 г.

3. Жданов А. А.. Операционные системы реального времени [Электронный ресурс]/ А. А. Жданов - URL: <http://www.asutp.ru/?p=600591>. Дата обращения: 10.03.12 г.

4. Е. Горошко. Операционные системы реального времени [Электронный ресурс]/ Горошко Е. - URL: <http://www.nsdevice.narod.ru/sprog/qnx/rtos.pdf> Дата обращения: 12.03.12 г.