

PUMPKIN™

REAL-TIME SOFTWARE

750 Naples Street • San Francisco, CA 94112 • (415) 584-6360 • <http://www.pumpkininc.com>
• Перевод: Андрей Шлеенков • <http://andromega.narod.ru> • <mailto:andromega@narod.ru> •

RM-MCC18

Справочное руководство

Справочное руководство *Salvo* для компилятора *Microchip MPLAB-C18*



Salvo™

The RTOS that runs in tiny places.™

Введение

Данное руководство предназначено для пользователей Salvo, использующих микроконтроллеры PIC18 PICmicro® компании Microchip (<http://www.microchip.com>) с компилятором Си MPLAB-C18 компании Microchip (<http://www.microchip.com>).

Связанные документы

При построении приложений Salvo с компилятором Си Microchip MPLAB-C18, вместе с данным руководством должны использоваться следующие документы Salvo:

Руководство пользователя Salvo (Salvo User Manual)
Приложение AN-12 (Application Note AN-12) - устаревшее
Приложение AN-25 (Application Note AN-25)

Замечание: Пользователям MPLAB-C18 настоятельно рекомендуется обновить Microchip MPLAB IDE до версии 6.30 или более поздней. Вместо AN-12 следует использовать AN-25.

Примеры проектов

Примеры проектов Salvo, с использованием компилятора Си Microchip MPLAB-C18 и среды разработки Microchip MPLAB IDE версий 5 или 6, могут быть найдены в следующих директориях каждого дистрибутива Salvo для Microchip PICmicro® MCUs:

```
\salvo\ex\ex1\syse  
\salvo\tut\tu1\syse  
\salvo\tut\tu2\syse  
\salvo\tut\tu3\syse  
\salvo\tut\tu4\syse  
\salvo\tut\tu5\syse  
\salvo\tut\tu6\syse
```

Свойства

Таблица 1 иллюстрирует основные особенности реализации Salvo для компилятора Си Microchip MPLAB-C18.

основное	
доступные дистрибутивы	Salvo Lite, LE & Pro for Microchip PICmicro® MCUs
поддерживаемые устройства	PIC18 PICmicro® MCUs
заголовочные файлы	portmcc.h
другие специфические для процессора файлы	portpic18.c
имена поддиректорий проекта	SYSE
salvocfg.h	
автоопределение компилятора?	да ¹
библиотеки	
поддиректория \salvo\lib	mcc18
класс хранения по умолчанию	auto
переключение контекста	
метод	при помощи OSCtxSw(label)
_OSLabel() требуется?	нет
объем автоматических переменных и параметров функций в задачах	не ограничен
память	
поддерживаемые модели памяти	small и large
поддерживаемые модели стека	single-bank и multi-bank
прерывания	
управляются через	GIEL и/или GIEH биты. Управляются через опцию конфигурации OSPIC18_INTERRUPT_MASK
статус прерывания сохраняется в критических секциях?	да
используемый метод	нужные GIE биты сохраняются в программном стеке на входе, прерывания запрещаются, и нужные GIE биты восстанавливаются из программного стека при выходе
степень вложенности	не ограничена ²
альтернативные методы возможны?	да ³
отладка	
отладка в исходных кодах?	только при построении с исходным кодом
компилятор	
поддержка упакованных битовых полей?	нет
printf() / %p поддерживается?	нет / нет
va_arg() поддерживается?	да ⁴

Таблица 1: Особенности реализации Salvo для компилятора Си Microchip MPLAB-C18

Библиотеки

Номенклатура

Имена библиотек Salvo для компилятора Си Microchip MPLAB-C18 следуют соглашениям, показанным на примере имени библиотеки на Рисунке 1.

Пример имени библиотеки: **sfc18sna.lib**

СИМВОЛЫ	ЗНАЧЕНИЕ	ВОЗМОЖНЫЕ ВАРИАНТЫ
s	Salvo	
f	тип	f : freeware l : standard
c18	MPLAB-C18	
s	модель памяти	l : large (пространство памяти <=2 МБ) s : small (пространство памяти <=64 КБ)
n	тип памяти глобальных объектов Salvo	f : far – указатели 16 бит n : near – указатели 8 бит
a	конфигурация	a : многозадачность с задержками и событиями d : многозадачность с задержками e : многозадачность с событиями m : многозадачность только t : многозадачность с задержками, событиями и ожиданиями с таймаутом

Рисунок 1: Номенклатура библиотек Salvo для компилятора Си Microchip MPLAB-C18

Тип

Дистрибутив Salvo Lite содержит *свободные (freeware)* библиотеки. Все остальные дистрибутивы Salvo содержат *стандартные (standard)* библиотеки. Дополнительную информацию о типах библиотек см. в главе *Библиотеки* документа *Руководство пользователя Salvo*.

Целевой процессор

Архитектура ядра⁵ PIC18 согласуется со всеми членами семейства. Каждая библиотека Salvo, сгенерированная с малой моделью памяти, может быть использована с любым членом семейства PIC18, имеющим объем памяти не превышающий 32 К-слов (64 КБ). Библиотеки Salvo, созданные с большой моделью памяти могут быть использованы с любым членом семейства PIC18.

Модель памяти

Компилятор Си Microchip MPLAB-C18 поддерживает малую (small) и большую (large) модели памяти. При построении с библиотеками, модель памяти, примененная во всех исходных файлах, должна совпадать с использованной в библиотеках. Для компиляции с исходными файлами, одна и та же модель памяти должна быть использована во всех исходных файлах.

код модели памяти	описание
l / OSL:	Большая (Large) модель памяти. Пространство программной памяти имеет максимум 1М-слов (2 МБ).
s / OSS:	Малая (Small) модель памяти. Пространство программной памяти имеет максимум 32 К-слов (64 КБ).

Таблица 2: Модели памяти библиотек Salvo для компилятора Си Microchip MPLAB-C18

Замечание: В отличие от конфигурации библиотеки и вариантов опций, определенных в файле `salvocfg.h`, для построения с библиотеками для выбранной модели памяти ничего не определено. Поэтому особое внимание должно быть уделено установкам модели памяти, используемым для создания приложения. Модель памяти обычно определяется на основе установок среды разработки IDE (например, MPLAB).

Тип памяти для глобальных объектов Salvo

Тип памяти для глобальных объектов Salvo в приложении можно определить выбором соответствующей библиотеки. Объекты `near` имеют быстрый доступ, но будут расходовать ценную память RAM в банке Access Bank. Объекты типа `far` будут размещены в банковской памяти RAM с более медленным доступом. Коды глобальных объектов Salvo перечислены в Таблице 3.

код типа памяти	описание
f / OSF:	Объекты Salvo объявлены как тип <code>far</code> и будут размещены в банковской памяти RAM.
n / OSN:	Объекты Salvo объявлены как тип <code>near</code> и будут размещены в первых 128 байтах внутренней памяти RAM (в <code>access RAM</code>).

Таблица 3: Типы памяти для библиотек Salvo для компилятора Си Microchip MPLAB-C18

Код, требуемый для доступа к глобальным объектам Salvo (например, к блокам управления задач или `tcbs`), будет различен по размеру и скорости в зависимости от того, где размещены объекты.

Так как в архитектуре PIC18 существует только 128 байт `access RAM`, в больших приложениях может оказаться необходимым размещение глобальных объектов Salvo в банковской RAM.

Конфигурация

С разными дистрибутивами Salvo поставляются разные конфигурации библиотек, позволяющие пользователю минимизировать объем ядра Salvo. См. главу *Библиотеки* документа *Руководство пользователя Salvo* для получения дополнительной информации о конфигурации библиотек.

Установки компиляции

Библиотеки Salvo для компилятора Си Microchip MPLAB-C18 созданы используя установки по умолчанию, описанные в главе *Библиотеки* документа *Руководство пользователя Salvo*. Специфические для процессора установки перечислены в Таблице 4.

ограничения компиляции	
макс. число задач	3
макс. число событий	5
макс. число флагов событий ⁶	1
макс. число очередей сообщений ⁷	1
специфические для процессора установки	
размер задержки	8 бит
холостой ход	разрешен
биты разрешения прерываний в течение критических секций	GIEH = GIEL = 0
указатели на сообщения	могут указывать на ROM или RAM
объекты Salvo	far
счетчик системного времени	доступен, 32 бита
приоритеты задач	разрешены
сторожевой таймер	очищается в OSSched()

Таблица 4: Установки и замены для библиотек Salvo для компилятора Си Microchip MPLAB-C18

Замечание: Ограничения компиляции библиотек Salvo могут быть изменены в меньшую сторону (все дистрибутивы Salvo) или в большую (все дистрибутивы Salvo кроме Salvo Lite) по сравнению со значениями по умолчанию. См. главу *Библиотеки* документа *Руководство пользователя Salvo*.

Доступные библиотеки

Существует 40 библиотек Salvo для компилятора Си Microchip MPLAB-C18. Каждый дистрибутив Salvo для Microchip PICmicro® MCU содержит также библиотеки Salvo из младших версий дистрибутивов.

Примеры salvocfg.h

Ниже приводятся примеры файлов конфигурации проекта salvocfg.h для различных дистрибутивов Salvo для PICmicro® MCU типа PIC18C452.

Компиляция с библиотеками Salvo Lite

```
#define OSUSE_LIBRARY           TRUE
#define OSLIBRARY_TYPE         OSF
#define OSLIBRARY_GLOBALS      OSF
#define OSLIBRARY_CONFIG       OSA
#define OSTASKS                 2
#define OSEVENTS               4
#define OSEVENT_FLAGS          0
#define OSMESSAGE_QUEUES       1
```

Листинг 1: Пример `salvocfg.h` для компиляции с библиотеками используя `sfc18lfa.lib`

Компиляция с библиотеками Salvo LE & Pro

```
#define OSUSE_LIBRARY           TRUE
#define OSLIBRARY_TYPE         OSL
#define OSLIBRARY_GLOBALS      OSF
#define OSLIBRARY_CONFIG       OSA
#define OSTASKS                 7
#define OSEVENTS               13
#define OSEVENT_FLAGS          3
#define OSMESSAGE_QUEUES       2
```

Листинг 2: Пример `salvocfg.h` для компиляции с библиотеками используя `slc18lfa.lib`

Компиляция с исходным кодом Salvo Pro

```
#define OSENABLE_IDLE_HOOK     TRUE
#define OSENABLE_SEMAPHORES    TRUE
#define OSTASKS                 9
#define OSEVENTS               17
#define OSEVENT_FLAGS          2
#define OSMESSAGE_QUEUES       4
```

Листинг 3: Пример `salvocfg.h` для компиляции с исходным кодом

Эффективность

Использование памяти

учебные примеры ⁸	всего ROM ⁹	всего RAM ¹⁰
tu1lite	644	34
tu2lite	976	36
tu3lite	1026	38
tu4lite	1914	47
tu5lite	2708	59
tu6lite	2988	62
tu6pro ¹¹	2770	58

Таблица 5: Требования памяти ROM и RAM для приложения Salvo, создаваемого компилятором Си Microchip MPLAB-C18

Специальные вопросы

Несовместимые оптимизации

Режим оптимизации компилятора Си Microchip MPLAB-C18, называемый *процедурная абстракция* является несовместимым с задачами Salvo из-за потенциального воздействия на переключатель контекста Salvo `OSCtxSw()`. Поэтому *данная оптимизация должна быть явным образом запрещена* (-Opa-) в каждом файле, содержащем задачи Salvo¹².

Замечание: В случаях, где эффект от процедурной оптимизации все-таки полезен для приложения, рекомендуется, чтобы были созданы специализированные модули исходных кодов (*.c), содержащие задачи Salvo и не содержащие других функций. Таким образом, область отключения оптимизации процедурных абстракций может быть ограничена только задачами Salvo, позволяя использование этой оптимизации для других функций там, где это целесообразно.

Классы хранения

Компилятор Си Microchip MPLAB-C18 поддерживает три различных глобальных класса хранения по умолчанию для локальных переменных и параметров функций: `auto`, `static` и `overlay`¹³. Класс хранения во всех модулях проекта должен совпадать для избежания ошибок времени компоновки.

Все библиотеки Salvo созданы с классом хранения по умолчанию `auto`. Чтобы построить приложение Salvo с другим классом хранения по умолчанию, необходим дистрибутив Salvo Pro для построения с исходными кодами с альтернативным классом хранения по умолчанию, или создания собственной библиотеки Salvo с альтернативным классом хранения по умолчанию¹⁴.

Различные классы хранения определяются во время компиляции с помощью опций командной строки `-sca`, `-scs` или `-sco`.

Стеки

По архитектурным соображениям, компилятор Си Microchip MPLAB-C18 передает параметры через программный стек, и использует аппаратный стек PIC18 для адресов возврата. Хотя компилятор поддерживает обе реентерабельные и статическую оверлейную модели, реализация Salvo для данного компилятора совместима только с реентерабельной моделью.

Размер стека

По умолчанию, Salvo сконфигурирован для работы с программным стеком MPLAB-C18 любого размера. Т.е. модель стека использует один или несколько банков (-Ls-). Если используется модель с одним банком (по умолчанию в MPLAB-C18), то параметр Salvo `OSMPLAB_C18_STACK_SIZE` может быть установлен в 256 или меньше и тем самым сократить используемый объем кода.

Предупреждение: Если выбрана модель стека MPLAB-C18 из нескольких банков с параметром `OSMPLAB_C18_STACK_SIZE` значением 256 байт или меньше, то генерируется ошибка.

Глобальные объекты Salvo

Размещение объектов в Near/Access или Banked RAM

С компилятором Си Microchip MPLAB-C18, глобальные объекты Salvo могут быть размещены полностью в одной из двух областей – в банковской или быстрой (access) памяти RAM. Если `OSMPLAB_C18_LOC_ALL_NEAR` установлено в `FALSE` (по умолчанию), все объекты Salvo размещаются в банковской памяти RAM. Для размещения всех объектов в быстрой (access) памяти RAM при построении с исходным кодом Salvo Pro необходимо установить `OSMPLAB_C18_LOC_ALL_NEAR` в `TRUE`.

Замечание: Из-за малого размера быстрой памяти access RAM в PIC18, размещение глобальных объектов Salvo в access RAM редко может быть подходящим решением.

Не существует никакой возможности отдельного размещения избранных глобальных объектов Salvo в быстрой (access) или банковской памяти RAM. Параметры конфигурации Salvo `OSLOC_XYZ` не применяются.

Ограничения на число задач, событий и т.п.

По умолчанию, каждый из больших глобальных объектов Salvo (например, массив блоков управления задачами, или `tcbs`) помещается в свою собственную *секцию данных*¹⁵. Это позволяет использовать максимально возможный размер массива (256 байт), поддерживаемый компилятором, и позволяет компилятору эффективно паковать все глобальные объекты Salvo в доступную память RAM.

Замечание: Из-за 256-байтного лимита на размер массивов, число задач Salvo, событий, очередей сообщений и т.д. ограничено не общим объемом памяти RAM процессора PIC18, а числом элементов, которые могут быть помещены в 256-байтный массив (т.е. в один банк).

Например, максимальное значение `OSTASKS` равно 42 в конфигурации Salvo, поддерживающей многозадачность и события, т.к. 42 задачи требуют массив размером `0xFC` (252) байта¹⁶. То же приложение также поддерживает до 51 события.

Процедуры обслуживания прерываний

По умолчанию, компилятор Си Microchip MPLAB-C18 предохраняет основной контекст в процедуре обработки прерывания (ISR). Чтобы корректно объявить процедуру обработки прерывания, которая вызывает сервис Salvo, параметр `save=` прагмы `interrupt` должен быть использован следующим образом:

```
#pragma interrupt ISR save=PROD,section(".tmpdata")

void ISR( void )
{
    ...
    OSTimer();
    ...
}
```

Замечание: Дополнительные функции в обработчике прерывания могут требовать дополнительных параметров `save=`. Дополнительно см. *MPLAB-C18 User's Guide*.

Управление прерываниями

Архитектура PIC18 поддерживает два различных уровня приоритетов прерываний. При разрешении используются два отдельных глобальных бита разрешения прерываний `GIEH` и `GIEL` для управления высоко- и низкоприоритетными прерываниями соответственно.

Прерывания автоматически запрещаются в критических секциях Salvo. По умолчанию, оба бита `GIEH` и `GIEL` сброшены (т.е. равны 0) в течение критических секций. Это управляется параметром `OSPIC18_INTERRUPT_MASK` – опцией конфигурации Salvo (по умолчанию: `0xC0`).

Пользователи Salvo Pro могут переконфигурировать способ запрета прерываний в критических секциях переопределением `OSPIC18_INTERRUPT_MASK` в файле проекта `salvocfg.h`. Например, если сервис Salvo (например, `OSTimer()`) вызывается только из низкоприоритетных прерываний, то величина `0x40` для `OSPIC18_INTERRUPT_MASK` обеспечит то, что только низкоприоритетные прерывания будут запрещены в критических секциях Salvo. В этой конфигурации на высокоприоритетные прерывания Salvo не влияет. Это полезно при использовании высокоприоритетных прерываний.

Замечание: Пользователи Salvo Pro имеют опцию генерации пользовательских библиотек с запрещением и разрешением прерываний, управляемых значениями не по умолчанию для `OSPIC18_INTERRUPT_MASK`. Для получения более подробной информации см. *Руководство пользователя Salvo*.

¹ Выполняется автоматически при помощи символа `__18CXX` определяемого компилятором.

² Архитектура PIC18 имеет аппаратный call-return стек глубиной 32 адреса.

³ Либо через встроенный ассемблер, либо через вызов функции.

⁴ Как MPLAB-C18 v.2.20.

⁵ Т.е. не периферийные спецрегистры, такие, как `TOSU|H|L`, `STKPTR`, `PCLATH|L`, `PCL`, `FSR0H|L` и т.п.

⁶ Каждый флаг события имеет свою память RAM, расположенную в своем блоке управления флагом события.

⁷ Каждая очередь сообщения имеет свою память RAM, расположенную в своем блоке управления очередью сообщений.

⁸ Salvo v.3.2.1 с MPLAB-C18 v.2.2x

⁹ В программных адресах (в словах).

¹⁰ В байтах, всех банках, `udata`. Не включает стек (по умолчанию: 0x100 байт). Глобальные объекты Salvo в банковской RAM (`far`).

¹¹ Построение с Salvo Pro несколько отличается от Salvo Lite при конфигурации – см. учебный `salvocfg.h`.

¹² По умолчанию все оптимизации MPLAB-C18 разрешены.

¹³ `overlay` применяется только к локальным переменным, но не к параметрам функции.

¹⁴ Это может потребовать изменений в системе Salvo `makefile`, т.к. компилятору необходимо передавать дополнительные параметры командной строки.

¹⁵ Как в Salvo v.3.2.4.

¹⁶ Например, с е-конфигурацией библиотеки.