Создание настроек для драйверов устройств и пакетов программ

Генератор BSP позволяет вам публиковать настройки для отдельных драйверов устройств и пакетов программ. Такие настройки являются видимыми и могут изменяться любым пользователем BSP, если в BSP представлен ваш драйвер или пакет программ. Используйте команды интерфейса Tcl для создания настроек.

Команда Tcl, которая публикует настройки, очень полезна, если ваш драйвер или пакет программ имеет опции сборки или работы в режиме прогона, которые обычно задаются с помощью оператора #define или определений в makefile на стадии сборки программы. Настройки также могут добавить декларирование собственной переменной в BSP **Makefile**.

Настройки влияют на генерируемый BSP следующими способами:

- Настройки добавляются в качестве переменных в BSP **system.h** или **public.mk**, или в BSP makefile.
- Настройки сохраняются в файле настроек BSP, в имени которого содержится информация об иерархии, чтобы избежать конфликта имён.
- Значение по умолчанию, которые вы выбираете, назначается в настройки, так что конечный пользователь драйвера или пакета программ избавлен от необходимости задавать их во время создания или обновления BSP.
- Настройки отображаются в документе BSP **summary.html**, рядом с текстовым описанием на ваш выбор.

Используйте Tcl команду add_sw_setting для добавления настроек. За задания подробностей, необходимо снабдить команду add_sw_setting следующими аргументами в указанном порядке:

1. type – Тип данных, который контролирует форматирование установки значения в соответствующем генерируемом файле.
2. destination – Файл результатов в BSP.
3. displayName – Имя, которое используется для идентификации настроек во время изменения настроек BSP или просмотра документа BSP **summary.html**.
4. identifier – Концептуально, этот аргумент – это макрос, определённый на языке Си (текст, следующий за директивой #define), или имя переменной в makefile.
5. value – Значение по умолчанию, назначенное в настройках, если пользователь BSP ещё не изменил его вручную.
6. description – Текстовое описание, отображаемое в документе BSP **summary.html**.

Тип данных

Несколько доступных настроек данных контролируются аргументом type команды add_sw_setting. Это относится к типам данных, которые вы выражаете как директиву #define или как значения, объединённые в переменных makefile. Специфика настройки type зависит от структуры вашей программы или от потребностей сборки BSP. Доступные типы данных и их обычное использование приведены в табл. 7-5.

Табл. 7-5. Настройки типов данных (часть 1 из 2)

Тип данных	Настройка значений	Примечание
Булево выражение	boolean_define_only	Выражение, которое генерируется, если истинно, и отсутствует, когда ложно. Используйте булево выражение в ваших исходных файлах Си с помощью конструкции #ifdef <*setting*> ... #endif

Табл. 7-5. Настройки типов данных (часть 2 из 2)

Тип данных	Настройка значений	Примечание
Булево назначение	boolean	Выражение, которому назначается 1, если истинно, и 0, когда ложно. Используйте булево назначение в ваших исходных файлах Си с помощью конструкции #if <*setting*> ... #else
Символ	character	Выражение с одним символом, окружённое одиночными кавычками (').
Десятичное число	decimal_number	Выражение с нецитированным, неформатированным десятичным числом, например 123. Прекрасно подходит для определения значений в программе, которая, например, имеет буфер с конфигурируемым размером: int buffer[SIZE];
Число с двойной точностью	double	Выражение с плавающей точкой двойной точности, например 123.4
Число с плавающей точкой	float	Выражение с плавающей точкой одинарной точности, например 234.5
Шестнадцатеричное число	hex_number	Выражение числа с префиксом 0x, например 0x1000. Прекрасно подходит для задания адреса памяти и маски битов.
Строка в кавычках	quoted_string	Выражение строки в кавычках, например "Buffer".
Строка без кавычек	unquoted_string	Выражение строки без кавычек, например BUFFER.

Настройки файлов результатов

Аргумент destination команды add_sw_setting задаёт настройки и устанавливает их значения. Этот аргумент контролирует файл, в который сохраняется настройка в BSP. Генератор BSP форматирует установленное значение настройки, основываясь на файле определений и типе настройки. В табл. 7-6 показаны возможные настройки аргумента destination.

Табл. 7-6. Настройки файла результата

Файл результата	Значение настройки	Примечание
system.h	system_h_define	В большинстве случаев рекомендуется этот файл. Ваш исходный код должен использовать оператор #include <system.h>, чтобы сделать доступной выражение настройки. Настройка применяется к оператору #define в **system.h**.
public.mk	public_mk_define	Выражения применяются в качестве оператора -D в **public.mk** в соде флагов Си предпроцессора. Этот тип настройки относится напрямую к компилятору во время сборки, он становится видимым во время компиляции приложения и библиотек, связанных с BSP.
BSP makefile	makefile_variable	Настройки применяются в качестве переменных установок makefile в BSP makefile.

Определённые типы настроек не совместимы с типами файлов результата **public.mk** или **Makefile**.

За дополнительной информацией, обратитесь к главе "Справка по Nios II SBT" в настольной книге программиста под Nios II.

Настройки отображаемого имени

Настройка displayName контролирует, что вводит конечный пользователь драйвера или пакета программ (разработчик BSP), для контроля над настройками этого BSP. BSP применяет текст displayName после разделителя "." (точки) в имя вашего драйвера или пакета программ (как это определено в командах create_driver или create_sw_package). Например, если ваш драйвер называется my_peripheral_driver, а displayName вашей настройки - small_driver, BSP с вашим драйвером будет иметь настройку my_peripheral_driver.small_driver. Таким образом, каждый драйвер или пакет программ будет иметь собственную настройку в поле имени.

Настройка имени генерации

Настройка generationName команды add_sw_setting контролирует физическое имя настройки в генерируемых BSP файлах. Физическое имя связано с определением, сделанным в файлах **public.mk** и **system.h**, или в переменной, созданной в BSP **Makefile**. Настройка generationName – это текст, который использует ваша программа в условно компилируемом коде. Например, допустим, что ваша программа создаёт буфер следующим образом:

```
unsigned int driver_buffer[MY_DRIVER_BUFFER_SIZE];
```

Вы можете ввести существующий текст, MY_DRIVER_BUFFER_SIZE, в аргумент generationName.

Настройка значения по умолчанию

Аргумент value команды add_sw_setting хранит значение по умолчанию для вашей настройки. Аргумент value передаётся генерируемому BSP, если конечный пользователь драйвера или пакета (разработчик BSP) не изменит настройки перед генерацией BSP.

В аргументе value можно задать любую настройку, однако она должна являться значением по умолчанию для драйвера или пакета программ в Tcl скрипте, или быть введённой пользователем, конфигурирующим BSP, но быть совместимой с выбранной настройкой.

За дополнительной информацией, обратитесь к главе "Справка по Nios II SBT" в настольной книге программиста под Nios II.

Настройка текстового описания

Аргумент description команды add_sw_setting содержит текстовое описание настройки. Аргумент description является необходимым. Текст описания поместите в двойные кавычки (""). Текст описания применяется в документе **summary.html**, генерируемом BSP.

Настройка примера создания

В примере 7-5 реализована настройка для драйвера, имеющего два варианта функции, одна реализована как малый драйвер (минимальное кодовое покрытие), а другая является быстрым драйвером (эффективное исполнение).

Example 7–5. Supporting Driver Settings

```
#include "system.h"
#ifdef MY_CUSTOM_DRIVER_SMALL
int send_data( <args> )
{
// Small implementation
}
#else
int send_data( <args> )
{
// fast implementation
}
#endif
```

В примере 7-5 в Tcl файл вашего драйвера добавлена простая настройка булево выражения. Это средство позволяет BSP пользователю контролировать ваш драйвер в настройках интерфейса BSP. Когда пользователь устанавливает настройку как true или 1, BSP определяет MY_CUSTOM_DRIVER_SMALL либо в **system.h**, либо в BSP **public.mk** файле. Когда пользователь компилирует BSP, ваш драйвер компилируется с соответствующей процедурой, находящейся в объектном файле. Когда пользователь снимает эту настройку, MY_CUSTOM_DRIVER_SMALL не определяется.

Вы добавляете настройку MY_CUSTOM_DRIVER_SMALL в ваш драйвер с помощью Tcl команды add_sw_setting следующим образом:

```
add_sw_setting boolean_define_only system_h_define small_driver
MY_CUSTOM_DRIVER_SMALL false
"Enable the small implementation of the driver for my_peripheral"
```

Каждая Tcl команда должна занимать одну линию в Tcl файле. В этом примере это изменено, по причине ограничений размеров страницы.

Каждый аргумент имеет несколько вариантов. За подробной информацией об использовании и ограничениях, обратитесь к главе "Справка по Nios II SBT" в настольной книге программиста под Nios II.