

## Обзор Nios II Simple Socket Server

В следующих разделах описан Nios II Simple Socket Server:

- "Унифицированные имена в программе" на стр. 1-16 - Здесь определяются унифицированные имена, используемые в учебных файлах.
- "Архитектура программы" на стр. 1-17 - Здесь описывается архитектурная модель приложения Nios II и её взаимодействие с остальными программными компонентами Nios II системы.
- "Ресурсы MicroC-OS/II" на стр. 1-19 - Здесь описываются задачи, очередность, флаги событий и семафоры, которые реализованы в программе Nios II Simple Socket Server.
- "Инициализация стека NicheStack TCP/IP" на стр. 1-20 - Здесь описываются необходимые задачи и функции из учебного материала для создания и эксплуатации соединения с гнездом Изернет TCP/IP.
- "Подробности реализации Nios II Simple Socket Server" на стр. 1-21 - Здесь описываются структуры, использующие это соединение с гнездом, и функции для каждого программного компонента, включая main(), инициализацию MicroC/OS-II и детали программных компонентов SSS и светодиодов.

### Унифицированные имена в программе

В проекте Nios II Simple Socket Server используются аббревиатуры заглавными буквами для идентификации общих ресурсов для каждого программного модуля, и строчные буквы с подчёркиванием для индикации частных ресурсов или функций для внутреннего использования в программном модуле. В табл. 1-2 отображаются идентификаторы с заглавными буквами.

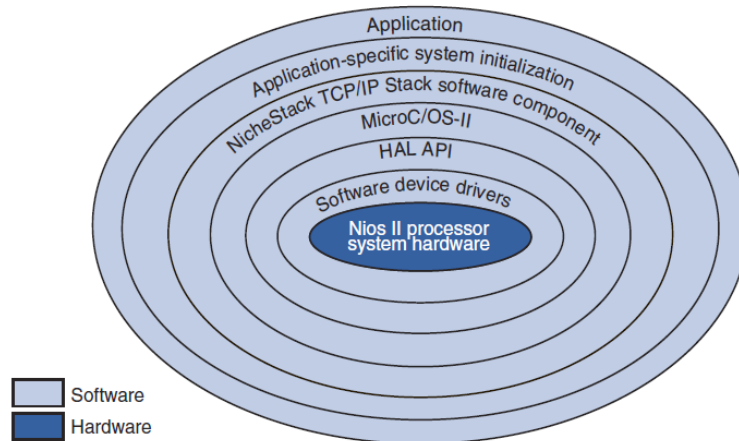
**Табл. 1-2. Аббревиатуры и имена программных модулей**

Аббревиатура	Имя
SSS	Программный модуль Nios II Simple Socket Server
LED	Программный модуль управления светодиодами
OS	Программный модуль MicroC/OS-II RTOS

## Архитектура программы

Диаграмма в виде луковицы на рис. 1-10 отображает архитектурные слои программного приложения Nios II MicroC/OS-II.

Figure 1-10. Layered Software Model

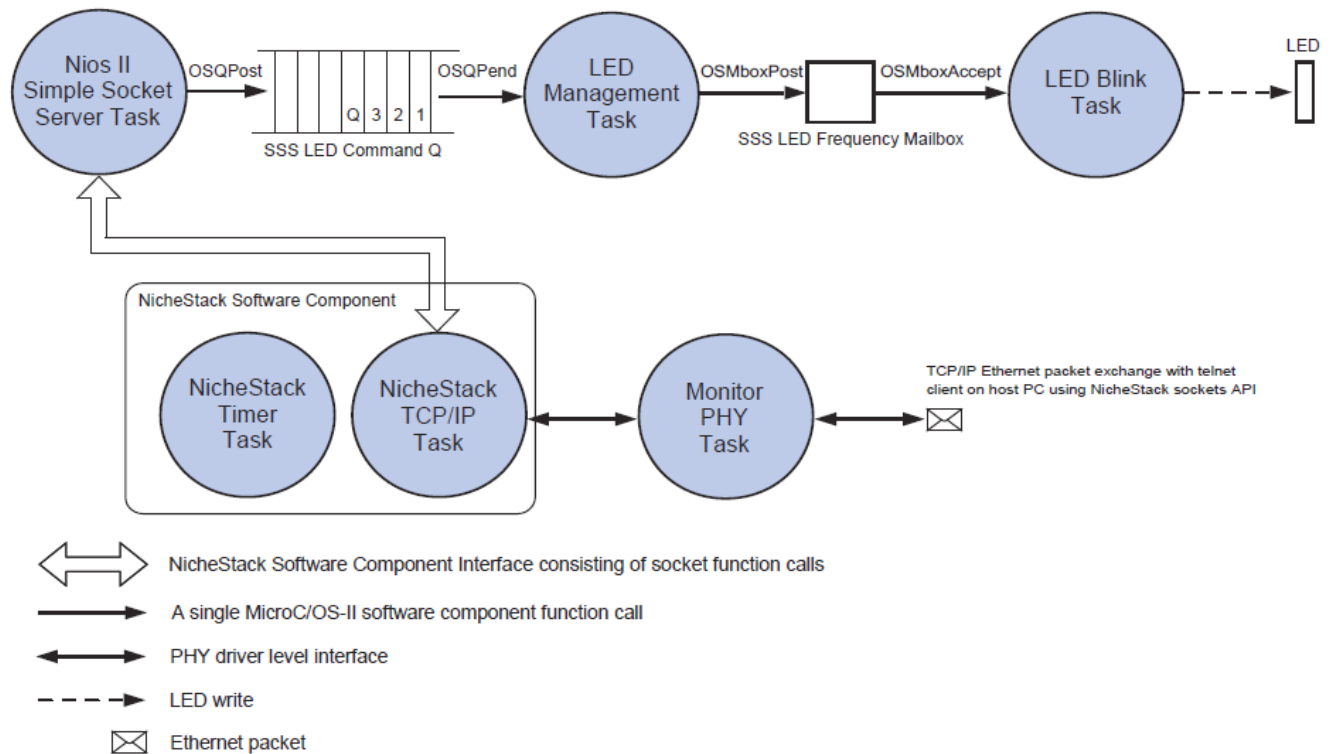


Каждый слой представляет собой законченную реализацию, позволяющую абстрагировать данные для следующего внешнего слоя. Ниже описаны слои:

- Nios II processor system hardware (Аппаратная часть системы с процессором Nios II) - Ядро этой диаграммы, отображает программный процессор Nios II и его аппаратную периферию, реализованную в FPGA.
- Software device drivers (Программные драйверы устройств) - Слой программных драйверов устройств содержит программные функции, управляющие Изернет и аппаратной периферией. В этих драйверах содержится информация об аппаратной реализации периферийных устройств, которая абстрагируется для следующих слоёв.
- HAL API - Уровень аппаратной абстракции (HAL) Altera интерфейса прикладного программирования (API) предлагает стандартный интерфейс для драйверов устройств, организует родственный интерфейс переносимой операционной системы (POSIX API) для следующих слоёв.
- MicroC/OS-II - Слой MicroC/OS-II RTOS предлагает многозадачность и сервисы взаимодействия между задачами для сетевого стека NicheStack TCP/IP и Nios II Simple Socket Server.
- NicheStack TCP/IP Stack software component - Слой программного компонента NicheStack TCP/IP Stack предлагает сетевые сервисы для слоёв приложения и слоя специального приложения инициализации системы через API гнезда (sockets API).
- Application-specific system initialization - Слой специального приложения инициализации системы содержит функции инициализации для программ MicroC/OS-II и NicheStack TCP/IP Stack, выведенные из main(), также в нём создаются все задачи для приложений, все семафоры, очередность и каждый флаг внутренних задач для ресурсов коммуникации RTOS.
- Application (Приложение) - Самый верхний слой приложения содержит задачу Nios II Simple Socket Server и задачи управления светодиодами.

На рис. 1-11 показана структура примера проекта, описанного во "Введении" на стр. 1-1. В следующих параграфах описываются задачи, изображённые на рисунке.

**Figure 1-11. Nios II Simple Socket Server Data Flow Diagram**



На рисунке показано состояние системы после инициализации. Когда программный компонент NicheStack TCP/IP Stack принимает Изернет пакет с командой для светодиода, посланной из программы клиента telnet, то NicheStack TCP/IP Stack обрабатывает входящий Изернет пакет с помощью протокола TCP/IP и представляет данные пакета в виде задачи для сервера гнезда (socket server), используя API гнезда. Затем извлекаются задачи управления светодиодом, и посылается команда LED, находящаяся в пакете данных, в очередь команды LED на исполнение.

## Ресурсы MicroC/OS-II

В этой главе описываются задачи, очередность, флаги событий и семафоры, реализованные в приложении Nios II Simple Socket Server.

### Задачи

В табл. 1-3 перечислены задачи MicroC/OS-II, реализованные в приложении Nios II Simple Socket Server.

**Табл. 1-3. Задачи MicroC/OS-II для приложения Nios II Simple Socket Server**

<b>Задача</b>	<b>Описание</b>
SSSInitialTask()	Инсталлирует все ресурсы MicroC/OS-II. Инициализирует стек NicheStack TCP/IP и RTOS структуры и задачи примера Nios II Simple Socket Server.
SSSNiosIISimpleSocketServerTask()	Управляет соединением с сервером гнезда и вызывает соответствующие процедуры управления соединением с гнездом.
LEDManagementTask()	Управляет LEDBlinkTask командами, принятыми в очередности MicroC/OS-II под именем SSSLEDCommandQ. Почтовый ящик MicroC/OS-II под именем SSSLEDFreqMailbox, передаёт скорость мигания для задачи мигания светодиодом.
LEDBlinkTask()	Мигает светодиодом на оценочной плате, основываясь на частоте, принятой через почтовый ящик MicroC/OS-II.
SSSMonitorPhyTask()	Контролирует состояние сети на физическом уровне и обслуживает соединения по сети.

Приложение создаёт задачи, перечисленные в табл. 1-3. Сетевой стек NicheStack TCP/IP создаёт два дополнительных программных компонента уровня задач: основную задачу, работающую с сетевым стеком, и задачу хронометража, использующую основную задачу. Приложение создаёт основную задачу (tk\_netmain) для NicheStack TCP/IP Stack в функции netmain() с приоритетом TK\_NETMAIN\_TPRIO. Приложение создаёт задачу хронометража (tk\_nettick) в вызове функции netmain() с уровнем приоритета TK\_NETTICK\_TPRIO. За подробной информацией об этих задачах, о том, как задать для них приоритеты и размер стека, обратитесь к главе "Важный концепт NicheStack TCP/IP Stack" на стр. 1-22.

### **Ресурсы коммуникаций внутренних задач**

Следующие глобальные обработчики (или указатели) создают и управляют ресурсами коммуникаций внутренних задач. Все ресурсы, начинающиеся с SSS, означают общие ресурсы, предоставляемые Nios II Simple Socket Server, которые распределены между программными модулями. Функция SSSCreateOSDataStructs, вызываемая из SSSInitialTask() декларирует и создаёт эти ресурсы в `niosII_simple_socket_server.c`.

#### **■ SSSLEDCommandQ**

SSSLEDCommandQ является очередью сообщений MicroC/OS-II, которая посылает команды от задачи simple socket server к задаче контроля над светодиодом на оценочной плате Altera, LEDManagementTask().

#### **■ SSSLEDFreqMailbox**

SSSLEDFreqMailbox является обработчиком для MicroC/OS-II почтового ящика скорости мигания светодиодом. Этот почтовый ящик сообщает скорость мигания светодиодом задаче контроля над светодиодом под именем LEDManagementTask(), а задача мигания светодиодом на оценочной плате называется LEDBlinkTask(). Задача LEDManagementTask() сообщает скорость мигания в соответствии с командой, посланной от задачи SSSNiosIISimpleSocketServerTask(), когда команда "Мигнуть светодиодом" приходит на гнездо TCP/IP.

### **Инициализация стека NicheStack TCP/IP**

Как написано в секциях "Задачи для NicheStack TCP/IP стека" и "Инициализация стека" главы "[Изернет и TCP/IP стек NicheStack - Версия под Nios II](#)" в [настойной книге программиста Nios II](#), NicheStack TCP/IP Stack должен быть проинициализирован кодом приложения Nios II Simple Socket Server, путём вызова следующих функций NicheStack:

- `alt_iniche_init()`, вызывается из SSSInitialTask() в `iniche_init.c`
- `netmain()`, вызывается из SSSInitialTask() в `iniche_init.c`

Вам необходимо предоставить три функции NicheStack: `get_mac_addr()`, `get_board_mac_addr()` и `get_ip_addr()`. Для этого учебного пособия эти функции находятся в файле `iniche_init.c`.

Если функция `get_board_mac_addr()` не может подобрать правильный MAC адрес, обратитесь к инструкции пользователя вашей оценочной платы за информацией по восстановлению MAC адреса, или за аппаратным кодом MAC адреса в функции `get_mac_addr()`.

Во время инициализации, задача под именем SSSInitialTask() вызывает функции инициализации `alt_iniche_init()` и `netmain()` в нужной последовательности, затем ожидает, пока NicheStack TCP/IP Stack перейдёт в полностью рабочее состояние (ожидание перехода глобальной переменной `iniche_net_ready` в состояние TRUE), прежде чем создавать задачу уровня приложения SSSNiosIISimpleSocketServerTask().

Задача SSSNiosIISimpleSocketServerTask() определена в файле `niosII_simple_socket_server.c` и создаётся с приоритетом `SSS_NIOS_II_SIMPLE_SOCKET_SERVER_TASK_PRIORITY`.

Вы сможете использовать задачу `SSSInitialTask()` в вашем собственном сетевом приложении MicroC/OS-II и NicheStack TCP/IP Stack.

### Подробности реализации Nios II Simple Socket Server

В этой главе описаны детали структуры `simple socket server` (простого сервера гнезда), задачи и функции.

Приложение Nios II Simple Socket Server использует следующую структуру для управления каждым конкретным подключением к гнезду:

```
typedef struct SSS_SOCKET {
    enum { READY, COMPLETE, CLOSE } state;
    int fd;
    int close;
    INT8U rx_buffer[SSS_RX_BUF_SIZE]; /* circular buffer */
    INT8U *rx_rd_pos; /* position we've read up to */
    INT8U *rx_wr_pos; /* position we've written up to */
} SSSConn;
```

Функция `main()` в этом приложении (расположенная в файле `iniche_init.c`) выполняет следующие действия:

- Вызывает `OSTimeSet()`
- Вызывает `OSTaskCreateExt`, которая включается при `SSSInitialTask()`
- Вызывает `alt_uCOSIIErrorHandler()`
- Вызывает `OSStart()` для начала многопоточного режима

В примерах Micrium MicroC/OS-II предлагается использовать одиночную задачу для инициализации остальных приложений. При таком подходе проверяется стек на корректную инициализацию разрешённых свойств. В этом учебном пособии, задача `SSSInitialTask()` (расположенная в `iniche_init.c`) инициализирует программу NicheStack TCP/IP Stack, инициализирует структуры данных операционной системы и запускает любую определённую пользователем сетевую задачу и обычные задачи. Задача `SSSInitialTask()` выполняет следующие специальные действия:

- Вызывает `alt_iniche_init()` для выполнения предварительной инициализации NicheStack Networking Stack.
- Вызывает `netmain()` для инициализации и запуска NicheStack Networking Stack.
- Инсталлирует `ssstask()` и `sssphytask()` (через `TK_NEWTASK`) для запуска сетевой задачи Nios II Simple Socket Server.
- Вызывает `SSSCreateOSDataStructs()` для создания структуры данных (ресурсов `SSSLEDCommandQ` и `SSSLEDFreqMailbox` RTOS) для приложения Nios II Simple Socket Server.
- Вызывает `SSSCreateTasks()` для создания не зависимой от NicheStack TCP/IP Stack задачи, включая задачи для светодиодов (LED).
- Вызывает `OSTaskDel()` для само удаления задачи.

Задача `SSSNiosIISimpleSocketServerTask()` (расположенная в `niosII_simple_socket_server.c`) выполняет следующие действия:

- Создаёт гнездо для обслуживания TCP/IP соединений, связывается с гнездом и ожидает запросы на TCP/IP соединение от клиента.
- Вызывает `SSS_handle_accept()` для входящего TCP/IP соединения.
- Вызывает `SSS_handle_receive()` для обслуживания TCP/IP соединения. Если вам требуется несколько TCP/IP соединений, вы сможете изменить эту задачу, чтобы создать другие задачи, обрабатывающие каждое конкретное TCP/IP соединение.
- Вызывает `SSS_reset_connection()`, `SSS_send_menu()`, `SSS_exec_command()`.
- Когда задача `SSSNiosIISimpleSocketServerTask()` принимает пакет данных, задача извлекается, и передаёт команды для светодиодов в `LEDManagementTask()` посредством `SSSLEDCommandQ`.

Далее перечислены задачи для светодиодов (расположенные в **leds.c**):

- Задача `LEDManagementTask()` потребляет и исполняет команды для светодиодов, полученные посредством `SSSLEDCommandQ`. Задача читает `SSSLEDCommandQ` в виде команды входящего сообщения от `SSSNiosIISimpleSocketServerTask()`, преобразует команду в частоту пульсации светодиодов, и отправляет эту частоту пульсации в почтовый ящик `SSSLEDFreqMailbox` для задачи `LEDBlinkTask()`.
- Задача `LEDBlinkTask()` мигает светодиодом в зависимости от частоты пульсации, установленной задачей `LEDManagementTask()` в почтовом ящике `MicroC/OS-II SSSLEDFreqMailbox`.