

Особенности конфигурирования TCP/IP стека для 1GbE портов АМС-модулей TORNADO-Аxxx

Все АМС-модули [TORNADO-Аxxx](#) фирмы МикроЛАБ Системс (далее ТАxxx) с процессорами общего назначения и процессорами ЦОС имеют два гигабитных порта (1GbE) АМС интерфейса (АМС порт #0 и #1), предназначенные для внешнего управления АМС-модулями через IP сети LAN/WAN от ПК и Android-устройств. В зависимости от требований приложения пользователя, эти порты могут быть подключены к общей или разным IP сетям и сконфигурированы для работы с динамическими или статическими IP адресами.

В зависимости от типа подключения, могут потребоваться дополнительные настройки TCP/IP стека используемой операционной системы (ОС) в зависимости от модели хоста ОС (*End System* или *Host Model*) в соответствии с RFC1122:

- *слабая модель хоста (Weak Host Model)* – TCP/IP стек принимает IP пакет, если IP адрес назначения совпадает с одним из адресов, который присвоен сетевым интерфейсам на данном хосте.
- *усиленная модель хоста (Strong Host Model)* – IP пакет будет принят только с IP адресом, соответствующим адресу сетевого интерфейса, на который он пришел.

При использовании ОС со слабой моделью хоста (ОС *Linux*) возможны проблемы при одновременном подключении обоих гигабитных портов АМС-модулей ТАxxx к одной IP сети.

В данной статье будут рассмотрены особенности настройки 1GbE-портов АМС-модулей ТАxxx при подключении к IP сетям при использовании разных ОС и приведены примеры использования библиотеки функций сетевой коммуникации *TASDK_DTE*, входящей в состав среды разработки ПО *TASDK* ([Среда разработки ПО для АМС модулей TORNADO-Аxxx](#)).

Подключение сетевых гигабитных интерфейсов АМС-модулей ТАxxx к разным IP сетям

При отсутствии дополнительных требований, наиболее простым и рекомендуемым способом одновременного подключения обеих 1GbE-портов АМС-модулей ТАxxx является их подключение к отдельным IP сетям, например:

- АМС порт #0 (сетевой интерфейс «eth0») подключается к IP сети с базовым адресом 192.168.0.0 и маской 255.255.255.0. Это может быть отдельная физическая IP сеть или логическая IP подсеть внутри одной физической IP сети.
- АМС порт #1 (сетевой интерфейс «eth1») подключается к IP сети с базовым адресом 192.168.1.0 и маской 255.255.255.0. Это может быть также отдельная физическая IP сеть или логическая IP подсеть внутри одной физической IP сети.

Это способ одновременного подключения двух 1GbE-портов АМС-модулей ТАxxx не требует никаких дополнительных настроек. При этом нет необходимости выполнять дополнительную настройку TCP/IP стека используемой ОС (ОС *Linux*, операционные системы реального времени (ОСРВ) *FreeRTOS* и *Azure RTOS ThreadX*), все сетевые интерфейсы устройства будут функционировать одновременно и независимо друг от друга.

В случае, если сетевые интерфейсы настроены на получение динамического IP адреса, то DHCP сервер, работающий в соответствующей IP сети, обеспечивает выдачу уникального IP адреса автоматически. При использовании статической IP адресации необходимо сначала убедиться, что выбранные статические IP адреса не конфликтуют с уже существующими в IP сети. Это можно сделать, например, при помощи команды «*ping*».

Подключение сетевых гигабитных интерфейсов АМС-модулей ТАxxx к общей IP сети

Существует целый ряд приложений и аппаратуры, которые могут потребовать безусловного подключения обеих 1GbE-портов АМС-модулей ТАxxx к общей IP сети, причем неважно какой конкретно - физической или логической. Одним из таких примеров является аппаратура с резервированием логических каналов коммуникации. При простоте

аппаратной реализации, программная поддержка такого подключения оказывается не очевидной и зависящей от типа ОС, используемой при написании приложения для процессора AMC-модулей TAxxx.

Рассмотрим пример, когда оба AMC порты #0 и #1 (сетевые интерфейсы «eth0» и «eth1») подключаются к общей IP сети с базовым адресом 192.168.0.0 и маской 255.255.255.0. Варианты последующей настройки будут отличаться в зависимости от используемой на устройстве ОС.

Использование ОС Linux

При использовании ОС Linux возможны проблемы при одновременном подключении обоих 1GbE-портов AMC-модулей TAxxx к одной IP сети. Например, если для обоих 1GbE-портов были назначены IP адреса из одной IP сети и в дальнейшем пропадает соединение по одному из них (обычно первому), то устройство может быть недоступно при обращении к другому порту. Такое поведение TCP/IP стека является особенностью ОС Linux.

По умолчанию, Linux использует слабую модель хоста (Weak Host Model). В соответствии с RFC1122 – TCP/IP стек принимает IP пакет, если IP адрес назначения совпадает с одним из адресов, который присвоен сетевым интерфейсам на данной системе. Включение усиленной модели хоста в Linux не является стандартным действием и требует изменения целого ряда настроек TCP/IP стека и правил маршрутизации, что может привести к нестабильной работе TCP/IP стека ОС Linux.

В настоящий момент не существует стандартного общепринятого решения для включения усиленной модели хоста в ОС Linux. В связи с этим, предлагается настройка и подключение 1GbE-портов AMC-модулей TAxxx в разные IP сети при их одновременном использовании.

Вариантом решения данной проблемы является использование либо системного вызова `setsockopt()` BSD-сокетов ОС Linux, либо API функции `TASDK_DTE_set_socket_option()` библиотеки `TASDK_DTE_LINUX.LIB` среды `TASDK` с параметром `SO_BINDTODEVICE`. Данный параметр позволяет связать сокет с конкретным сетевым интерфейсом с именем «eth0» или «eth1».

Пример использования:

```
int32_t r;
TASDK_DTE_SOCKET_DD listen_socket_dd;
char str[8];
struct ifreq optval_set_bindtodevice;
int32_t optlen_set_bindtodevice = sizeof(optval_set_bindtodevice);
...
memset(&listen_socket_dd, 0, TASDK_DTE_SOCKET_DD_LEN);

/* Get local interface name for selected port */
sprintf(str, "eth%u", port);

/* Open listening connection */
if ((r = TASDK_DTE_server_open_connection(INADDR_ANY, SERVER_PORT,
TASDK_DTE_OPEN_CONNECTION_FLAGS_SET_DEFAULT, &listen_socket_dd)) != TASDK_DTE_OK)
    goto exit;

/* Bind the socket to the selected interface */
strcpy(optval_set_bindtodevice.ifr_name, str);
if ((r = TASDK_DTE_set_socket_option(listen_socket_dd.sock, SOL_SOCKET, SO_BINDTODEVICE,
&optval_set_bindtodevice, optlen_set_bindtodevice)) != TASDK_DTE_OK)
    goto exit;
printf("TASDK_DTE_set_socket_option() with SO_BINDTODEVICE at port %u is completed\n", port);
```

После этого пакеты, принятые только этим сетевым интерфейсом, будут обработаны данным сокетом. Следует отметить, что вызов `bind()` с локальным IP адресом сетевого интерфейса не приводит к такому же поведению.

Использование OCPB FreeRTOS и TCP/IP стека lwIP

TCP/IP стек lwIP позволяет использовать одновременное подключение нескольких сетевых интерфейсов устройства к общей IP сети без дополнительных настроек.

Для этого, при создании сокета и его привязки к локальному адресу необходимо передать локальный IP адрес сетевого интерфейса либо в функцию `bind()` стека lwIP, либо в API функции `TASDK_DTE_server_open_connection()` и `TASDK_DTE_client_open_connection()` библиотеки `TASDK_DTE_FREERTOS.LIB` среды `TASDK`.

Пример использования:

```
int32_t r;
TASDK_DTE_SOCKET_DD listen_socket_dd;
uint32_t local_ip_addr;
...
memset(&listen_socket_dd, 0, TASDK_DTE_SOCKET_DD_LEN);

/* Get local interface IP address */
local_ip_addr = netif_instance[port].ip_addr.addr;
```

```

/* Open listening connection */
if ((r = TSDK_DTE_server_open_connection(local_ip_addr, SERVER_PORT,
TSDK_DTE_OPEN_CONNECTION_FLAGS_SET_DEFAULT, &listen_socket_dd)) != TSDK_DTE_OK)
goto exit;

```

Еще одним вариантом привязки сокета к сетевому интерфейсу является использование либо системного вызова *setsockopt()*, либо API функции *TSDK_DTE_set_socket_option()* библиотеки *TSDK_DTE_FREERTOS.LIB* среды *TSDK* с параметром *SO_BINDTODEVICE*. Данный параметр позволяет связать сокет с конкретным сетевым интерфейсом, который по умолчанию имеет имя «en0» и «en1».

Использование OCPB Azure RTOS ThreadX и TCP/IP стеков NetX и NetX Duo

TCP/IP стеки *NetX* и *NetX Duo* для OCPB Azure RTOS ThreadX (Microsoft) позволяют использовать одновременное подключение нескольких сетевых интерфейсов устройства к общей IP сети без дополнительных настроек.

Для этого, при создании сокета и его привязки к локальному адресу необходимо передать локальный IP адрес сетевого интерфейса либо в функцию *bind()* опции BSD-сокетов для стеков *NetX* и *NetX Duo*, либо в API функции *TSDK_DTE_server_open_connection()* и *TSDK_DTE_client_open_connection()* библиотеки *TSDK_DTE_THREADX.LIB* среды *TSDK*. Локальный IP адрес сетевого интерфейса можно получить при помощи системного вызова *nx_ip_interface_address_get()* стеков *NetX* и *NetX Duo*.

Пример использования:

```

int32_t r;
TSDK_DTE_SOCKET_DD listen_socket_dd;
uint32_t local_ip_addr, local_netmask;
...
memset(&listen_socket_dd, 0, TSDK_DTE_SOCKET_DD_LEN);

/* Get IP address from selected port */
if ((status = nx_ip_interface_address_get(&ip_instance, port, &local_ip_addr, &local_netmask)) !=
NX_SUCCESS)
{
    printf("Error: nx_ip_interface_address_get() failed with error %u", status);
    ...
}

/* It seems nx_ip_interface_address_get() returns IP address in host byte order */
local_ip_addr = htonl(local_ip_addr);
/* Open listening connection */
if ((r = TSDK_DTE_server_open_connection(local_ip_addr, SERVER_PORT,
TSDK_DTE_OPEN_CONNECTION_FLAGS_SET_DEFAULT, &listen_socket_dd)) != TSDK_DTE_OK)
goto exit;

```

Использование OCPB TI SYS/BIOS и TCP/IP стека NDK для процессоров ЦОС TMS320C6678 AMC-модулей TORNADO-A6678xx

TCP/IP стек *NDK* для OCPB TI DSP SYS/BIOS фирмы Texas Instruments для процессоров ЦОС TMS320C6678 стандартно не поддерживает одновременное подключение нескольких сетевых интерфейсов устройства к общей IP сети. В связи с этим, он был существенно доработан фирмой МикроЛАБ Системс и уже в составе среды ПО *TSDK* полностью поддерживает несколько сетевых интерфейсов устройства в общей IP сети.

Для этого, при создании сокета и привязки его к локальному адресу необходимо передать локальный IP адрес сетевого интерфейса либо в функцию *bind()* стека *NDK*, либо в API функции *TSDK_DTE_server_open_connection()* и *TSDK_DTE_client_open_connection()* библиотеки *TSDK_DTE_SYSBIOS.LIB* среды *TSDK*.

Пример использования:

```

int32_t r;
TSDK_DTE_SOCKET_DD listen_socket_dd;
uint32_t local_ip_addr;
...
memset(&listen_socket_dd, 0, TSDK_DTE_SOCKET_DD_LEN);

/* Open listening connection */
if ((r = TSDK_DTE_server_open_connection(local_ip_addr, SERVER_PORT,
TSDK_DTE_OPEN_CONNECTION_FLAGS_SET_DEFAULT, &listen_socket_dd)) != TSDK_DTE_OK)
goto exit;

```

Фирма «МикроЛАБ Системс»

Дубнинская ул, д. 83, оф. 612, г.Москва, 127591, Российская Федерация
тел. +7-(499)-900-6208
E-mail: info@mlabsys.ru WEB: www.mlabsys.ru