

# Руководство по тестированию платы TE-STM32F107

## Содержание

Назначение .....	1
Требования .....	1
Тестовый стенд .....	1
Методика тестирования .....	2
Описание демостационной программы .....	5

## Назначение

Программа тестирования предназначена для проверки функционирования модулей TE-STM32F107, включая интерфейсы **ETHERNET**, **CAN1**, **CAN2**.

Итогом тестирования является заключение о работоспособности микроконтроллера и перечисленных интерфейсов платы.

Критерий тестирования интерфейса **ETHERNET** - отображение WEB-сайта, размещенного в памяти микроконтроллера на тестируемой плате на экране персонального компьютера, подключенного к локальной сети.

Критерий тестирования интерфейсов **CAN1** и **CAN2** - удачное/неудачное прохождение тестового обмена между разъемами CAN1 и CAN2 с получением результата тестирования (сообщения) с помощью протокола и TELNET.

## Требования

Для тестирования необходимы:

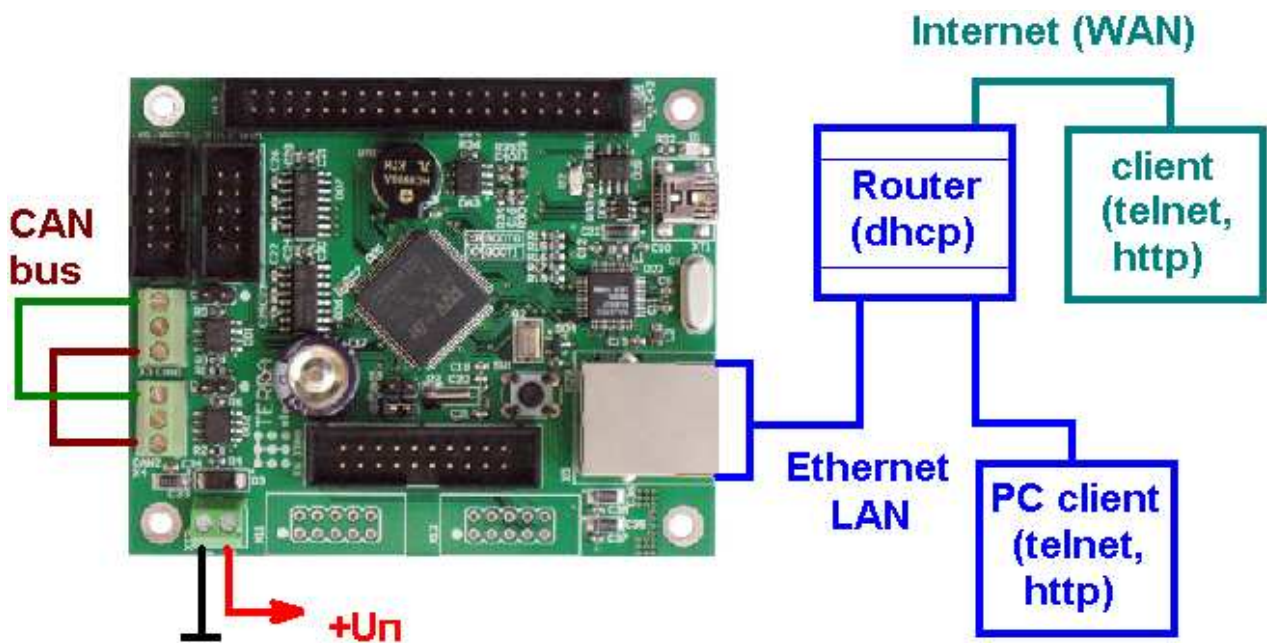
Персональный компьютер, подключенный к локальной сети, с установленным клиентом TELNET и WEB-проводником.

тестируемый модуль TE-STM32F107 с соединенными между собой интерфейсами CAN. В память программ микроконтроллера должен быть загружен тестовый пример WEB-сайта.

*В расширенном варианте, обеспечивающем тестирование USB и определенный уровень диагностики, необходим USB кабель для подключения платы к PC.*

## Тестовый стенд

Тестовый стенд представлен на рисунке.



### Методика тестирования

Перед сборкой стенда необходимо запрограммировать микроконтроллер модуля. В комплекте на диске присутствует файл собранной прошивки

/TE-STM32F107\_uIP\_demo/out/image.hex - прошивка в шестнадцатеричном формате INTEL, которую можно запрограммировать любым программатором, поддерживающим программирование stm32f107.

/TE-STM32F107\_uIP\_demo/out/image.bin – бинарный ELF образ прошивки, пригодный для программирования с помощью OpenOCD/ARM USB OCD.

В случае необходимости изменить или пересобрать проект, Вы можете воспользоваться прилагаемым документом «Руководство по установке, настройке и использованию GNU Tools/Eclipse IDE в среде Windows для разработки программного обеспечения микроконтроллеров ARM», в котором детально описана как это сделать прилагаемыми к проекту средствами разработки.

После подачи напряжения питания или системного сброса плата выполняет следующие действия:

выполняется инициализация и запуск uIP стека

присваивается MAC адрес 12:07:aa:99:12:55

запускается DHCP клиент, который после выполнения запроса, ожидает ответ от DHCP сервера с информацией о назначенной сетевой конфигурации (назначенный IP адрес)

при получении ответа присваивается IP адрес, маска, IP адрес шлюза и SNTP сервера.

в случае таймаута назначается IP адрес 192.168.0.8, маска 255.255.255.0, IP адрес шлюза 192.168.0.1.

далее запускаются сервера HTTP(80 порт) и TELNET(23 порт)

в случае соединения удаленных клиентов с серверами HTTP и TELNET, выполняется обслуживание запросов.

Для выполнения тестирования необходимо:

- проверить связь с платой, используя утилиту PING;

```
C:\WINXP\system32\cmd.exe
Microsoft Windows XP [Версия 5.1.2600]
(C) Корпорация Майкрософт, 1985-2001.

C:\Documents and Settings\klen.DEVELOPER>ping 192.168.0.194

Обмен пакетами с 192.168.0.194 по 32 байт:

Ответ от 192.168.0.194: число байт=32 время=1мс TTL=128
Ответ от 192.168.0.194: число байт=32 время=3мс TTL=128
Ответ от 192.168.0.194: число байт=32 время<1мс TTL=128
Ответ от 192.168.0.194: число байт=32 время<1мс TTL=128

Статистика Ping для 192.168.0.194:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0 (0% потерь),
Приблизительное время приема-передачи в мс:
    Минимальное = 0мсек, Максимальное = 3 мсек, Среднее = 1 мсек

C:\Documents and Settings\klen.DEVELOPER>
```

- проверить работоспособность HTTP сервера;

Вас приветствует TE-STM32F107/uIP WEB сервер! - Opera

Файл Правка Вид Закладки Виджеты Инструменты Справка

Открыть Сохранить Печать Найти Домой Мозаика Каскад Голос

Вас приветствует TE-ST... x +

http://192.168.0.194/ Google

Найти на странице Найти следующее Голос Режим автора Показывать все изображения Подогнать по ширине


100%

[корень](#) [статистика WEB сервера](#) [статистика IP стека](#) [TCP/IP соединения](#) [Тест платы](#)


Этот WEB сайт размещена на плате TE-STM32F107 и обслуживается HTTP сервером, выполняющимся поверх **uIP embedded TCP/IP** стека.

Кликните по ссылке в заголовке для просмотра статистики.

Сайт является частью комплекта программного обеспечения поддержки платы TE-STM32F107.



**АППАРАТНАЯ ЧАСТЬ TE-STM32F107**

 Модуль TE-STM32F107 Барракуда с ядром Cortex-M3 и портом Ethernet производства [Терраэлектроника](#)

- проверить работоспособность CAN интерфейсов. Для этого необходимо с помощью утилиты TELNET создать соединение с платой. Командная оболочка TELNET сервера платы поддерживает коанду **can**, по которой выполняется тестовый обмен между модулями CAN и возвращается результат тестирования.

```

C:\ Telnet 192.168.0.194
uIP command shell
Type '?' and return for help
TE-STIM32F107> can
test packet transfer CAN1->CAN2 and CAN2->CAN1 is OK?
TE-STIM32F107>_

```

## Описание демонстрационной программы

Программное обеспечение состоит из

- файлов документации;
- исходных кодов для тестового firmware — программное обеспечение микроконтроллера;
- архивов средств разработки.

## Дерево директорий файлов проекта

<b>SDK</b>	директория поддержки разработки проектов(SDK)
<i>doc</i>	документация не относящаяся непосредственно к
проекту	
<i>libs</i>	внешние библиотеки
<i>scripts</i>	скрипты утилит используемых при разработке
<i>make</i>	скрипты GNU MAKE
<i>openocd</i>	скрипты OpenOCD
<i>gdb</i>	скрипты GNU GDB
<b>TE-STIM32F107_uIP_demo</b>	корневая директория демо-приекта
<i>doc</i>	документация по микросхемам
<i>lib</i>	собранные при компиляции библиотеки
<i>out</i>	выходные файлы (ELF,прошивка, дамп, листинг...)
<i>pc</i>	исходный программ для PC (не реализовано)
<i>scripts</i>	скрипты проекта (скрипт линкера)
<i>src</i>	исходные коды и файлы данных проекта
<i>application</i>	исходный код демо-приложения
<i>include</i>	общие заголовки
<i>libcmsis</i>	исходеый код библиотеки STM CMSIS
<i>libethernet</i>	исходеый код библиотеки STM Ethernet V1.1.0
<i>libfreertos-6.0.2</i>	исходеый код FreeRTOS V6.0.2
<i>libhardware</i>	исходеый код библиотеки STM Firmware V3.1.2
<i>libuip-1.0</i>	исходеый код библиотеки TCP/IP uIP V1.0

## Настройка проекта

Для демонстрации возможности использования различных библиотек демо- проект возможно собрать в различных конфигурациях. Конфигурация и настройки проекта содержится в файле *TE-STM32F107\_uIP\_demo/ src/ options.mk*. Рекомендуется начать изучение проекта именно с него. Некоторые параметры проекта:

### USE\_FREERTOS

Параметр задающий сборку с использованием FreeRTOS или без нее. Самый простой случай, с которого рекомендуется изучать проект – без использования FreeRTOS. В данной конфигурации программа представляет собой основной бесконечный цикл расположенный в функции *main*. В этом цикле периодически опрашиваются флаги запроса на отложенную обработку прерываний периферийного модуля ethernet и “периодическую активность” uIP. При установке этих флагов выполняется обработка принятых данных по Ethernet и возможно ответ (например ответ на HTTP запрос). В этом случае установите параметр **USE\_FREERTOS=0**. При использовании конфигурации с применением FreeRTOS логическая структура программы строится в терминах **задача**. Для обслуживания прерывания по приему данных по Ethernet и для реализации “периодической активности” uIP организуются отдельные независимые **задачи**. **Задачи** выполняются независимо, управляются по прерываниям от модуля Ethernet и системного таймера. Планировщик задач FreeRTOS выполняет разделение процессорного времени между задачами. Для соблюдения логики работы библиотеки uIP выполняется синхронизация между задачами с помощью **семафоров**. В такой конфигурации установите **USE\_FREERTOS=1**.

### USE\_DHCP

Параметр, задающий сборку с использованием динамической сетевой конфигурацией периферийного Ethernet модуля. В простейшем случае **USE\_DHCP=0** динамическая конфигурация не используется – IP адрес, маска подсети, адрес шлюза устанавливается в коде вызовами библиотеки uIP. В случае установки **USE\_DHCP=1**, реализуется DHCP клиент, который выполняет запрос на конфигурацию. В случае наличия в сети доступного DHCP сервера, клиент получает сетевую конфигурацию (адрес, маска подсети, адрес шлюза) и устанавливает ее автоматически. В случае если ответ по запросу не получен, то по таймауту выполняется конфигурация по умолчанию как в первом случае.

## Краткое описание применения FreeRTOS в демо-проекте

FreeRTOS — многозадачное ядро реального времени для встраиваемых систем. В демо-примере предпринята попытка продемонстрировать использования FreeRTOS. Применение средств FreeRTOS позволяет выполнять проектирование проекта на более абстрактном уровне – не на уровне основного цикла приложения и вызовов реализующих функциональность устройства, а в терминах отдельных независимых **задач**, которые могут взаимодействовать. Это более естественно и близко к терминам предметной области. С точки зрения программиста **задача** – это просто функция с бесконечным циклом, в котором выполняется определенное специфическое действие. Можно провести аналогию с функцией *main* в типовом проекте – FreeRTOS посредством **задач** реализует необходимое разработчику число независимых “*main*”, которые используют процессор, память и периферию независимо друг от друга и при необходимости могут взаимодействовать (выполнять синхронизацию, передавать данные друг другу). Вторым важным моментом является то, что планировщик задач, выполняющий планирование задач по шкале времени является своеобразным «менеджером времени». Теперь Вам во многих случаях не потребуется использовать отдельные таймеры для организации сложного асинхронного по времени выполнения, необходимого набора действий. Планировщик **задач** позволяет планировать, с точностью до системного “тика”, временной профиль выполнения для каждой **задачи** в отдельности согласно логике предметной области. Все что нужно сделать – это знать это временные

интервалы в системных тиках и вызвать соответствующие функции FreeRTOS. Третьим важным моментом является то, что разработчик может управлять приоритетами **задач**.

Таким образом FreeRTOS позволяет облегчить проектирование и написание кода, сделать разработку более абстрактной, избавить от повторяющегося из проекта в проект необходимости реализовывать одно и то же.

В демонстрационном примере организовано две пользовательские **задачи**. Одна из них выполняется только при необходимости обработать принятые данные. Запуск инициируется в коде обработчика прерываний Ethernet модуля. Вторая реализует циклическую функциональность uIP. Продемонстрировано применение средств синхронизации **задач** на примере **семафоров**.