**Министерство науки и высшего образования Российской Федерации  
федеральное государственное автономное образовательное учреждение высшего образования  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»  
(Университет ИТМО)**

**Факультет программной инженерии и компьютерной техники**

**Лабораторная работа №3**

**«Запуск freeRTos на cLab»**

**Дисциплина «Встроенные системы»**

Выполнили: Калугина М.М.

Саржевский И.А.

Группа: P3402

Санкт-Петербург  
2020

# Описание cLab

Представленная система основана на sdk32.

Для работы предоставлен шаблон, в который можно писать пользовательский код. Исходный код шаблона находится по ссылке <https://github.com/lmtspbru/SDK_cLAB>.

Клиентская часть выполнена в виде веб-приложения, расположенного по адресу <https://sdk.ifmo.ru/remotesdk>

Для начала рассмотрим окружение:

В интерфейсе пользователя есть поле для загрузки bin-файла проекта и 2 окна: терминал, в котором отображается вывод программы по SDK\_TRACE\_Print и график прошедших событий.

Для работы необходимо написать исходный код на языке C, конфигурация портов ввода/вывода, а также различной перифирии производится в STM32CubeIDE (рекомендуется) или STM32CubeMX., собрать его и отправить бинарный файл на сервер. Для работы в системе был использован шаблон, предоставляемый создателями cLab для написания пользовательского кода.

Предоставленная нам система имеет 3 лампочки, возможность добавить 10 пользовательских событий и функции для вывода текста в консоль, при этом размер трассировочного буфера ограничен 256 байтами. Использование трассировочного буфера для вывода каких-либо данных в терминал и для отображения событий возможно между командами MX\_TRACE\_Init(); и SDK\_TRACE\_Stop();. При этом на выполнение пользовательского кода отведено 8 секунд.

Кроме того, в системе можно работать с прерываниями и таймером, хотя в шаблоне, предложенном для выполнения лабораторной настроенную в этом проекте периферию рекомендуется не менять

# Задание

Имея предоставленную систему сLab, попробовать установить на ней freeRTos, описать процесс установки и его результаты

# Ход работы

Для того, чтобы в интерфейсах cLab выводились какие-либо события в граф и можно было отслеживать работу через доступные нам интерфейсы, пользовательский код должен находится между SDK\_TRACE\_Start и SDK\_TRACE\_Stop.

После настройки FreeRTos по мануалу (https://www.digikey.com/en/maker/projects/getting-started-with-stm32-introduction-to-freertos/ad275395687e4d85935351e16ec575b1), при попытке запустить ОС при помощи osKernelStart между SDK\_TRACE\_Start и SDK\_TRACE\_Stop(рис 1), получаем от системы cLab ошибку, потому что в таком случае никогда не выполняется команда SDK\_TRACE\_Stop.

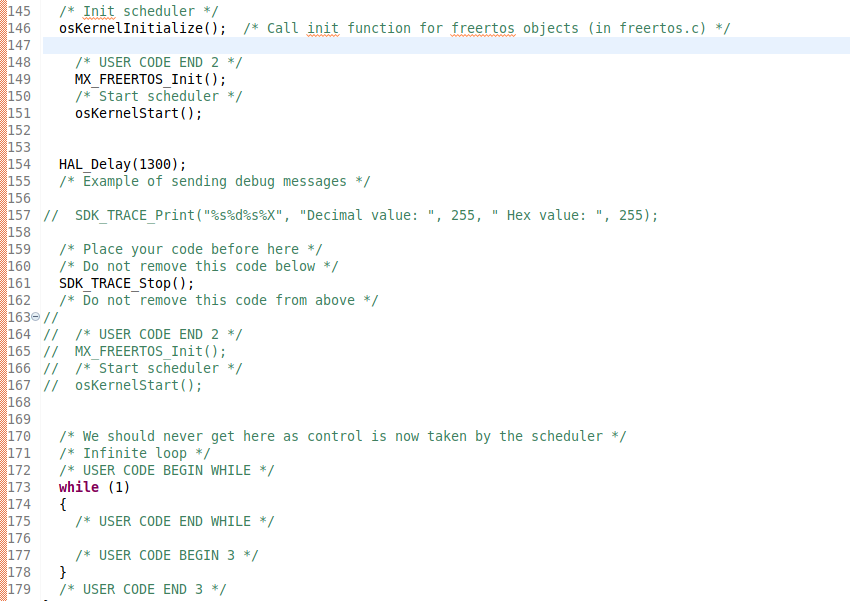


Рис. 1

Для завершения работы нужно, чтобы SDK\_TRACE\_Stop выполнился, поэтому была предпринята попытка вынести в отдельную таску для ОС команду, которая завершала бы запись в трассировочный буфер (рис 2). В этом случае получает от cLab ошибку. Но так как cLab не позволяет подробно увидеть содержание ошибки, то без понимания причин, дальнейшии попытки вынести ошибку в отдельную таску не привели к положительному результату.

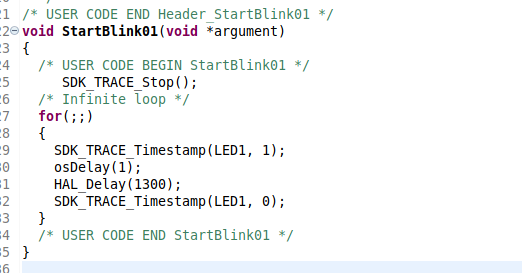


Рис 2.

Если пытаться запустить freeRTos после SDK\_TRACE\_Stop, то сразу с выполнением команды SDK\_TRACE\_Stop выполнение пользовательского кода в cLab завершается. (Рис 3, 4)

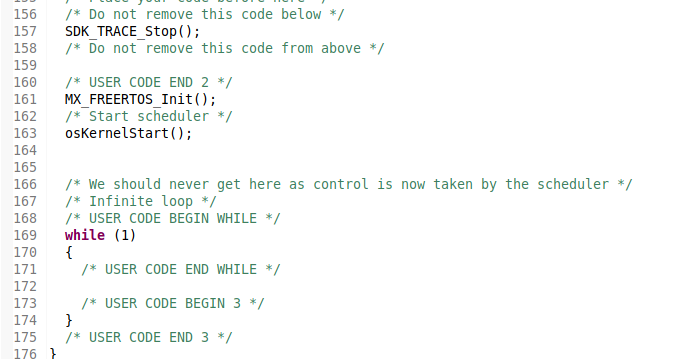


Рис 3

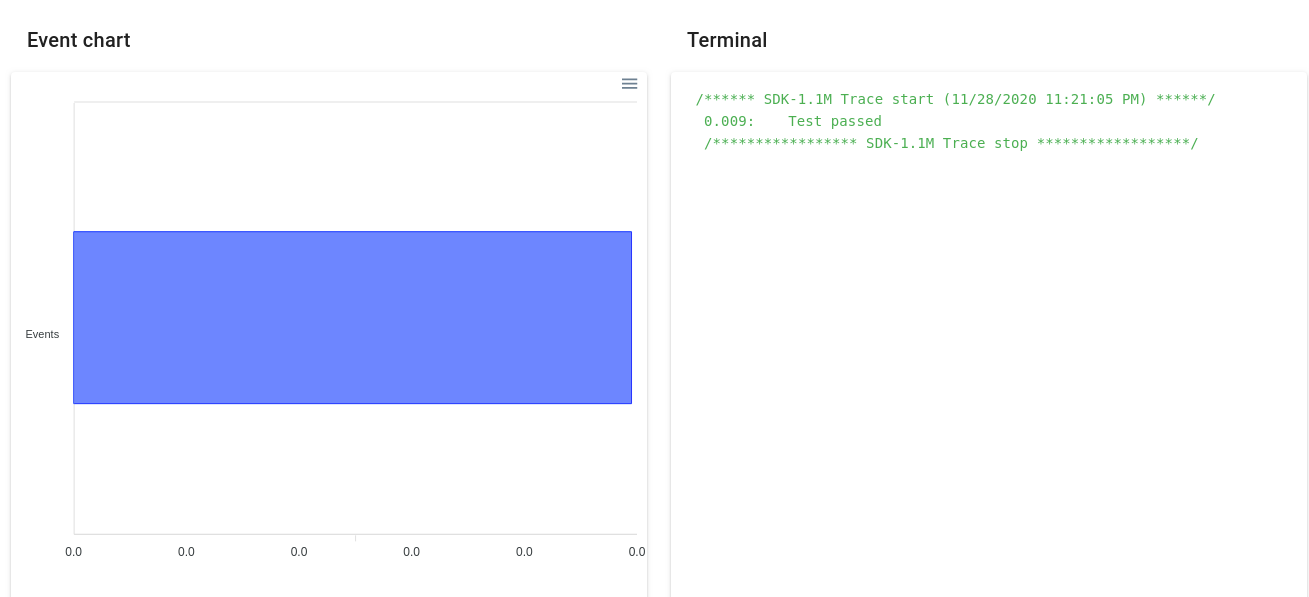


Рис 4

Альтернативно была предпринята попытка адаптировать существующий проект по запуску freeRTos на стенде SDK-1.1M (https://github.com/lmtspbru/SDK-1.1M/tree/master/SDK\_FreeRTOS) для запуска в cLab.

Для этого были добавлены необходимые сорс-файлы, а именно trace.c и crc32.c и их заголовки. Был модифицирован main, в него были добавлены вызовы необходимых для cLab функций (MX\_TRACE\_Init, SDK\_TRACE\_Start, SDK\_TRACE\_Stop). В сам freeRTos был добавлен новый поток, который раз в определенное время выводил в консоль cLab строку “It works!” (спойлер - не works).

|  |
| --- |
| void MX\_FREERTOS\_Init(void) {  osThreadDef(Task\_UART, StartTaskUART, osPriorityNormal, 0, 128);  Task\_LED2Handle = osThreadCreate(osThread(Task\_UART), NULL);  }  void StartTaskUART(void const \* argument)  {  /\* USER CODE BEGIN StartTaskUART \*/  /\* Infinite loop \*/  for(;;)  {  SDK\_TRACE\_Timestamp(PRINT, 1);  SDK\_TRACE\_Print("It works!");  SDK\_TRACE\_Timestamp(PRINT, 0);  osDelay(400);  }  /\* USER CODE END StartTaskUART \*/  } |

Проект с данными модификациями успешно собирается в STM32CubeIDE, однако в cLab запустить его не удалось. Так как cLab не пишет никакой информации об ошибке и не предоставляет никаких средств для отладки, адаптировать проект удалось.

# Вывод

В результате работы было предположено, что система freeRTos на текущем варианте cLab не может быть запущена для использовании в доступном нам окружении при помощи тех методов, что были рассмотрены.

Основная причина невозможности запустить FreeRTos в системе cLab заключается в том, что FreeRTos и cLab имеют разный жизненный цикл, который нельзя совместить между собой. Помимо того, система cLab не имеет на данный момент каких-либо инструментов отладки кода, что затрудняет проведение тестирования и настройки пользовательских программ.