Настройка программного обеспечения

# Выбор STM

Для разработки нужен удобный инструмент. Инструмент должен включать в себя среду разработки и отладки программного обеспечения (IDE), совместимую с микропроцессорами STM, и среду разработки графического интерфейса (GUI).

IDE должна поддерживать RTOS и иметь средства отладки под управлением RTOS.

Среда разработки GUI должна позволять рисовать экранную графику разрабатываемого устройства и отлаживать её в симуляторе, автоматически создавать программный код для отлаженного GUI и загружать его в IDE.

Инструмент разработки ПО должен легко устанавливаться и настраиваться для работы.

Поставленным условиям отвечает среда разработки от STMicroelectronics – STM32CubeIDE со встроенной графической библиотекой TouchGFX и графическим редактором TouchGFX Designer.

# Установка IDE

Установка ПО начинается с установки пакета STM32CubeIDE. Для установки выбрана версия 1.12.0, поскольку этот пакет имеется в доступе для скачивания.

После установки STM32CubeIDE должен быть обновлён до версии 1.13.0.

Подробное описание пакета STM32CubeIDE содержится в stm32cubeide-user-guide.pdf.

После установки CubeIDE требует подключения к сайту ST с авторизацией, однако в силу обстоятельств доступ из России блокирован. Нужно установить VPN сервис.

Авторизация в st.com:

* + 1. Почта: alsh100464@gmail.com
    2. Пароль: Sassas100464

# Установка TouchGFX

Установка графической библиотеки выполняется из установленного STM32CubeIDE путём добавления middleware. Подробное описание процесса установки здесь - <https://support.touchgfx.com/docs/introduction/installation>.

Вместе с установкой графической библиотеки TouchGFX скачивается и дистрибутив графического редактора TouchGFX Designer.

Установка TouchGFX Designer выполняется путём запуска дистрибутива из папки C:\Users\<user>\STM32Cube\Repository\Packs\STMicroelectronics\X-CUBE-TOUCHGFX\4.13.0

В июне 2023 года загружается версия 4.21.4.

В июле 2023 года появилась в обновлении версия 4.22.0. В этой версии добавилась возможность сжатия изображений для компактного хранения программы.

В июне 2024 года появилась в обновлении версия 4.24.0. Генерация кода в этой версии заканчивается ошибками при компиляции в STM32CubeIDE с любыми версиями GNU Tools for STM32. Рабочей версией остаётся версия 4.22.0.

Подробное описание TouchGFX содержится на сайте <https://support.touchgfx.com/docs/category/introduction>

Описание можно скачать оттуда в виде файла.

TouchGFX генерирует код для компилятора GNU Tools for STM32 (10.3-2021.10). После обновления STM32CubeIDE до версии 1.13.0 рабочим компилятором в ней установится GNU Tools for STM32 (11.3.rel1), который не совместим с кодом предыдущей версии компилятора. Требуется установка старой версии компилятора.

# Установка другой версии компилятора в STM32CubeIDE

**Внимание!!!** Проект создан в TouchGFX Designer **ver.4.22.0** и работает только с ним! Попытка перекодировать проект в ver. 4.24.0 закончилась ошибками.

На 28.07.2024 с проектом работает STM32CubeIDE ver.1.16.0, но только с **GNU Tools for STM32 версии 10.3-2021.10**!

Открываем свойства проекта Project/Properties.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описаниеВ С/С++ Build выбираем Settings.

На вкладке Tool Settings выбираем MCU Toolchain. Здесь смотрим, какой toolchain (компилятор, сборщик, отладчик и т.д.) установлен для workspace, а какой для проекта (Fixed).

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описаниеИзменить установки можно в менеджере, нажать на кнопку Toolchain Manager. Придётся подождать, пока Toolchain Manager скачает список доступных toolchains.

Если нужный пакет toolchain версии 10.3-2021.10 не установлен, то его нужно выбрать и нажать кнопку Install, после чего ответить на вопрос по выбору пакета, во втором окне уточняющая информация и – Finish. Окно закроется и начнётся процесс скачивания пакета и установки. Процесс идёт втихую, только в статусной строке будет информация о скачивании и установке. По окончании IDE попросит перезагрузку.

# Начало проектирования

Начинать лучше с создания проекта в графическом редакторе TouchGFX Designer. Для выбранного процессора или платы будет создан проект и выполнены необходимые настройки в нём для запуска RTOS и дисплея с тачскрином.

Далее можно перейти для разработки кода в IDE, при этом сохранится связь проекта с TouchGFX Designer. В любое время можно откорректировать имеющийся GUI или расширить его, не ломая разработанный пользователями код.

При совместной разработке тот, кто программирует GUI, создаёт основную ветвь проекта в репозитории, а потом подключает к проекту других участников проекта.

# Совместное проектирование

Совместное проектирование ведётся в среде GitHub, ведётся отслеживание и учёт различных версий программы.

Для работы нужно установить Git и GitCraken из [\\DC\Files\distrib\Git](file:///\\DC\Files\distrib\Git).

Зарегистрироваться в Git и получить приглашение на присоединение к проекту.

Так как GitCraken самостоятельно выгружается из компьютера, надёжнее использовать GitDesktop, после установки которого нужно выполнить клонирование проекта в локальный компьютер.

Для корректной работы проекта на локальном компьютере разработчика с помощью файла .gitignore, расположенного в корне проекта, должны быть исключены из синхронизации файлов папки:

* + 1. /TouchGFX/generated/\*
    2. /TouchGFX/build/\*
    3. /STM32CubeIDE/Debug/\*
    4. /Middlewares/\*

После получения в локальный репозиторий проекта из облака GitHub нужно запустить TouchGFX designer запуском файла «…проект\TouchGFX\имя\_проекта.touchgfx».

Проверить работу графической оболочки запуском симулятора.

Построить исполняемый код проекта в TouchGFX.

Далее переходим в CubeIDE путем запуска файла «…проект\ \STM32CubeIDE\.cproject». При запуске файла проект импортируется в IDE и сообщит об успешном выполнении импорта.

В импортированный в IDE проект не попадает файл \*.ioc, его нужно скопировать из корня проекта в проект STM32CubeIDE, предварительно переименовав. Имя файла должно соответствовать имени проекта в файле «…проект\ \STM32CubeIDE\.project». Обычно в первых строках этого файла: <projectDescription> <name>STM32F429I-DISCO</name>. Соответственно файлу .ioc нужно присвоить имя STM32F429I-DISCO.ioc.

Имя проекту можно присвоить любое. Для этого перед запуском файла «…проект\ \STM32CubeIDE\.cproject» в файле «…проект\ \STM32CubeIDE\.project» нужно поменять имя проекта, а также это имя присвоить и файлу \*.ioc. CubeIDE сам создаст нужные ссылки с новым именем.

Файл \*.ioc устанавливается в проект перетаскиванием файла из файлового проводника операционки в папку проекта в IDE, нужно мышкой перетащить и положить пиктограмму файла именно на папку проекта в IDE.

При перетаскивании выбрать опцию Link to files, галочку оставить.

После чего можно .ioc запустить и выполнить миграцию на обновлённую версию.

Всё, теперь можно редактировать, компилировать проект и загружать его в отладочную плату.

Разработку своего куска программы нужно вести в GitHub в отдельной ветке проекта, периодически объединяя свою разработку с разработками других участников проекта.

# Показать весь проект в Project Explorer

Загрузить в окно Project Explorer весь проект, а не только STM32F429I-DISCO, можно в File/Import.

Выбираем Existing Projects into Workspace.

Выбираем корневой директорий с полным проектом. В окне Projects появится проект, размещённый в выбранной папке. Если он ещё не был загружен в Workspace, он будет доступен для выбора. Нужно этот проект отметить галочкой.

Установить галочку в пункте Copy projects into workspace. В этом случае проект будет скопирован в рабочую область и появится в окне Project Explorer.

# Отладка во FreeRTOS

Настраиваем FreeRTOS в ioc. В Advanced settings нужно включить new library reentrant, чтобы компилятор заново подключал библиотеки без спроса.

Для отладки программ во FreeRTOS требуется выполнить настройки проекта по описанию п.6.2. в stm32cubeide-user-guide.pdf. Посмотреть, какие кнопки нажимать, можно здесь <https://www.youtube.com/watch?v=KTsrP28Gh6E>. Видео скачано в виде файла FreeRTOS aware debug.mp4 в папку /Настройка. Ниже привожу текстовое описание процесса.

FREERTOS/Config parameters/Run time and task stats gathering related definitions – все три параметра должны быть в состоянии Enabled.

FREERTOS/Config parameters/Kernel settings/RECORD\_STACK\_HIGH\_ADDRESS – установить в Enabled.

Теперь нужно сгенерировать код (code generation).

Процессор STM32F4 содержит DWT (Data Watchpoint and Trace unit). Я настроил DWT в качестве системного счётчика тиков вместо TIM7 для отладки в RTOS, так же как в видео выше. Настройку TIM7 нужно пропустить.

Находим в дереве проекта файл Application/User/freertos.c. Находим функцию \_\_weak void configureTimerForRunTimeStats(void). Между скобками вставляем текст:

CoreDebug->DEMCR |= CoreDebug\_DEMCR\_TRCENA\_Msk;

DWT->CTRL |= 1;

DWT->CYCCNT = 0;

В следующей функции \_\_weak unsigned long getRunTimeCounterValue(void) заменяем «return 0;» на « return DWT->CYCCNT;».

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описаниеСтроим (build) проект и конфигурируем отладочный сеанс из меню кнопки Debug:

Во вкладке Debugger конфигурирования отладки проекта устанавливаем флажок Enable RTOS Proxy. Выбираем драйвер FreeRTOS и порт ARM\_CM4F.

Чуть ниже в этой же вкладке нужно снять флажок Enable live expressions.

Всё. Нажать Apply и Debug.

После запуска отладчика в дереве проекта появится список выполняющихся задач с адресом размещения каждой задачи.

Чтобы посмотреть задачи, очереди, семафоры и таймеры RTOS нужно включить соответствующие окна из меню Window/Show View/FreeRTOS.

Если после всех манипуляций по настройке отладки во FreeRTOS в окне FreeRTOS Task List view отображается N/A вместо таймбара с %, надо изменить уровень оптимизации задач Middleware/FreeRTOS/tasks.c (п.6.2.1.3 в stm32cubeide-user-guide.pdf):

1. Right-clicking for tasks.c in *Project Explorer* view and open *Properties*

2. Select [**Properties**]>[**C/C++ Build**]>[**Settings**]>[**Tool Settings**]>[**Optimization**]

3. Set [**Optimization Level**] to None (-O0)

Так как счётчик тиков длительности выполнения задач имеет ограниченную длину, он довольно быстро переполняется и вместо таймбара с % появляется CNT OVL. Нужно или увеличить разрядность счётчика (а он уже 32-х разрядный), или сделать автосброс и начать считать с 0, либо ограничиться коротким временем оценки работы программы.