

**Экосистема операционной системы FX-RTOS
Руководство по сборке и установке**

Оглавление

Сборка и установка библиотек связующего программного обеспечения.....	3
Библиотека, реализующая протокол USB-устройства	3
Библиотека, реализующая протокол CAN, CanOpen	3
Библиотека реализующая журналируемый FAT	4
Библиотека, реализующая трансляцию адресов и выравнивание износа Flash-памяти (NAND FTL)	4
Библиотека, реализующая протокол MQTT на базе сетевого стека.....	5
Сборка и установка версий FX-RTOS для различных аппаратных платформ.....	5
FX-RTOS для архитектуры x86_64.....	5
FX-RTOS для архитектуры Эльбрус 2000 (e2k)	6
Гипервизор на платформе ARM AArch64	6
Гипервизор на платформе ARM AArch64	6
Сборка и установка драйверов контроллеров периферийных интерфейсов.	7

Сборка и установка библиотек связующего программного обеспечения.

Библиотека, реализующая протокол USB-устройства

Сборка библиотеки USB должна выполняться в каталоге `.\USB device\lib` относительно корневой директории экосистемы.

Предварительные условия: компилятор и компоновщик должны быть доступны через переменную PATH.

1. Установить переменную окружения `GCC_PREFIX` для используемого компилятора. Пример: `set GCC_PREFIX=arm-none-eabi-`
2. Выполнить сборку проекта командой `build.bat`

Результат: Создается файл статической библиотеки с расширением `*.a` в директории сборки.

Библиотека USB-устройства может быть компонована с основным проектом встраиваемого приложения. Для этого нужно указать соответствующие опции для компилятора/линкера.

Запись программы в ПЗУ микроконтроллера осуществляется через JTAG-интерфейс посредством внешнего JTAG-адаптера либо через встроенный загрузчик по UART.

Библиотека, реализующая протокол CAN, CanOpen

Сборка библиотеки USB должна выполняться в каталоге `.\CAN\lib` относительно корневой директории экосистемы.

Предварительные условия: компилятор и компоновщик должны быть доступны через переменную PATH.

1. Установить переменную окружения `GCC_PREFIX` для используемого компилятора. Пример: `set GCC_PREFIX=arm-none-eabi-`
2. Выполнить сборку проекта командой `build.bat`

Результат: Создается файл статической библиотеки с расширением `*.a` в директории сборки.

Библиотека CAN может быть компонована с основным проектом встраиваемого приложения. Для этого нужно указать соответствующие опции для компилятора/линкера.

Запись программы в ПЗУ микроконтроллера осуществляется через JTAG-интерфейс посредством внешнего JTAG-адаптера либо через встроенный загрузчик по UART.

Библиотека реализующая журналируемый FAT

Сборка библиотеки отказустойчивой FAT-совместимой ФС должна выполняться в каталоге `.\Journaling FAT\lib` относительно корневой директории экосистемы.

Предварительные условия: компилятор и компоновщик должны быть доступны через переменную PATH.

1. Установить переменную окружения `GCC_PREFIX` для используемого компилятора. Пример: `set GCC_PREFIX=arm-none-eabi-`
2. Выполнить сборку проекта командой `build.bat`

Результат: Создается файл статической библиотеки с расширением `*.a` в директории сборки.

Библиотека ФС может быть компонована с основным проектом встраиваемого приложения. Для этого нужно указать соответствующие опции для компилятора/линкера.

Запись программы в ПЗУ микроконтроллера осуществляется через JTAG-интерфейс посредством внешнего JTAG-адаптера либо через встроенный загрузчик по UART.

Библиотека, реализующая трансляцию адресов и выравнивание износа Flash-памяти (NAND FTL)

Сборка библиотеки NAND FTL должна выполняться в каталоге `\NAND FTL\lib` относительно корневой директории экосистемы.

Предварительные условия: компилятор и компоновщик должны быть доступны через переменную PATH.

1. Установить переменную окружения `GCC_PREFIX` для используемого компилятора. Пример: `set GCC_PREFIX=arm-none-eabi-`
2. Выполнить сборку проекта командой `build.bat`

Результат: Создается файл статической библиотеки с расширением `libnandftl.a` в директории сборки.

Библиотека NAND FTL может быть компонована с основным проектом встраиваемого приложения. Для этого нужно указать соответствующие опции для компилятора/линкера.

Подготовка оборудования и загрузка на устройство

Запись программы в ПЗУ микроконтроллера осуществляется через JTAG-интерфейс посредством внешнего JTAG-адаптера либо через встроенный загрузчик по UART.

Библиотека, реализующая протокол MQTT на базе сетевого стека

Сборка библиотеки NAND FTL должна выполняться в каталоге \MQTT\lib относительно корневой директории экосистемы.

Предварительные условия: компилятор и компоновщик должны быть доступны через переменную PATH.

1. Установить переменную окружения GCC_PREFIX для используемого компилятора. Пример: set GCC_PREFIX=arm-none-eabi-
2. Выполнить сборку проекта командой build.bat

Результат: Создается файл статической библиотеки с расширением libfxnet.a в директории сборки.

Библиотека LIBFXNET может быть компонована с основным проектом встраиваемого приложения. Для этого нужно указать соответствующие опции для компилятора/линкера.

Подготовка оборудования и загрузка на устройство

Запись программы в ПЗУ микроконтроллера осуществляется через JTAG-интерфейс посредством внешнего JTAG-адаптера либо через встроенный загрузчик по UART.

Сборка и установка версий FX-RTOS для различных аппаратных платформ

FX-RTOS для архитектуры x86_64

Сборка ядра OCPB совместно со слоем портирования для x64, известную так же под названиями AMD64, IA64, x86_64, осуществляется компилятором GCC на рабочей станции под управлением ОС Linux:

Порядок проверки:

Сборка ядра:

- 1) Запустить «make src_os» внутри сборочной директории для целевой системы.
Исходники ядра для целевой системы должны появиться внутри папки src_os;
- 2) Запустить "make kernel":
- 3) Запустить «make iso», чтобы преобразовать kernel.elf в os.iso.

Образ можно запустить на виртуальной машине с помощью следующей командной строки:
`qemu-system-x86_64 -serial vc -cpu Skylake-Server -cdrom os.iso`

FX-RTOS для архитектуры Эльбрус 2000 (e2k)

Сборка FX-RTOS со слоем портирования для архитектуры набора команд Эльбрус 2000 версии 4, компилятором LCC рабочей станции с процессором Эльбрус 8С:

Сборка ядра:

- 1) Запустить «make srcosu» внутри сборочной директории для целевой системы. Исходники ядра для целевой системы должны появиться внутри папки src;
- 2) Скопировать директорию src в папку с демо приложением и запустить команду "make":

Результат: исполняемый двоичный файл `fxrtos_demo.bin` должен появиться после компиляции.

Гипервизор на платформе ARM AArch64

Сборка FX-RTOS со слоем портирования для 64-битной архитектуры набора команд armv8-a, компилятором GCC на рабочей станции под управлением ОС Linux:

Сборка ядра:

- 1) Установить переменную окружения `GCC_PREFIX` для используемого компилятора под AArch64. Пример: `set GCC_PREFIX=aarch64-elf-`
- 2) Запустить «make src_os» внутри сборочной директории для целевой системы. Исходники ядра для целевой системы должны появиться внутри папки `src_os`.
- 3) Запустить "make kernel":

Результат: исполняемый файл `fxrtos.elf` должен появиться после компиляции.

Гипервизор на платформе ARM AArch64

Сборка FX-RTOS с гипервизором для 64-битной архитектуры набора команд armv8-a, компилятором GCC на рабочей станции под управлением ОС Windows 10:

1. Установить переменную окружения `GCC_PREFIX` для используемого компилятора под AArch64. Пример: `set GCC_PREFIX=aarch64-elf-`
2. Выполнить сборку всех компонентов системы командой `build.bat`. Данная команда объединяет ядра хостовой (FX-RTOS Microkernel) и гостевой ОС (FX-RTOS Nanokernel) в один исполняемый ELF-файл `fxrtos.elf`.

3. Запустить эмулятор QEMU:

```
# qemu-system-aarch64.exe -sdl -smp 1 -machine sbsa-ref -kernel fxrtos.elf
```

Данная команда запускает эмулятор, передавая ему на выполнение исполняемый файл с ядром гипервизора и гостевой операционной системы.

Сборка и установка драйверов контроллеров периферийных интерфейсов.

Драйверы контроллеров периферийных интерфейсов в составе:

- Драйвер адаптера Ethernet для МК NXP LPC1768
- Драйвер адаптера Ethernet для МК ST STM32F4
- Драйвер адаптера Ethernet для МП Atmel SAMA5D
- Драйвер USB устройства МК Миландр серий 1986BE9x, 1986BEх
- Драйвер контроллера MMC для МП Atmel SAMA5D
- Драйвер контроллера шины CAN для МК 1986BE9x
- Драйвер контроллера шины CAN для МК ST STM32F4

Предварительные условия: компилятор и компоновщик должны быть доступны через переменную PATH.

1. Установить переменную окружения GCC_PREFIX для используемого компилятора. Пример: set GCC_PREFIX=arm-none-eabi-
2. Выполнить сборку проекта командой build.bat

Результат: Создается файл статической библиотеки с расширением *.a в директории сборки.