



Базовый набор требований и рекомендаций по построению аппаратных платформ на базе СМК ВЕ-М1000

История изменений

Версия	Дата	Описание
1.0	26.09.2020	Начальная версия
1.1	14.04.2021	Изменено название, проведена корректировка всего текста
1.2	21.02.2022	В таблице 1 “Структура образа boot_flash” раздела 3.2 “boot_flash” размер DTB увеличен с 64КБ до 256КБ

Содержание

1	ВВЕДЕНИЕ	1
2	ПРОЦЕСС НАЧАЛЬНОЙ ЗАГРУЗКИ ВЕ-М1000	2
3	ТРЕБОВАНИЯ ДЛЯ ПОСТРОЕНИЯ СИСТЕМЫ НА ВЕ-М1000	2
3.1	SYSTEM_FLASH	2
3.2	BOOT_FLASH	2
3.3	SPD DDR4 SDRAM	3
3.4	RTC	3
3.5	BMC	4
3.6	UART	4
3.7	1Gb ETHERNET PHY RESET	4
3.8	PCIe RESET	4

1 Введение

В документе описаны основные подходы к обеспечению совместимости разрабатываемых аппаратных решений с системным Программным Обеспечением (ПО) из состава *Software Development Kit (SDK) ARM64*.

Следование указанным в документе рекомендациям позволит использовать существующее системное ПО без доработки и сократить сроки и стоимость разработки программно-аппаратных решений на базе микропроцессора ВЕ-М1000.

В состав SDK ARM64 входит следующий набор системных программных модулей:

- SCP firmware – в бинарном формате
- ARM Trusted Firmware-A (TF-A) – в исходных кодах
- TianoCore UEFI (EDK II) - в исходных кодах
- Linux kernel - в исходных кодах



2 Процесс начальной загрузки ВЕ-M1000

В процессе начальной загрузки микропроцессора участвуют следующие аппаратные модули:

- *System Control Processor (SCP)* – управляющий процессор Модуля Управления Системой
- *system_flash* - Микросхема энергонезависимой памяти типа NOR на интерфейсе Boot SPI контроллера загрузки (SPI0, имена выводов - SPI0_*)
- *boot_flash* - Микросхема энергонезависимой памяти типа NOR на интерфейсе SPI1 (имена выводов – SPI1_*)
- *mailbox* – встроенная SRAM-память, которая доступна для программного кода, исполняемого как на SCP, так и на любом ядре Cortex-A57

После подачи питания запускается процесс начальной загрузки микропроцессора, который производится в следующей последовательности:

1. SCP начинает исполнять код встроенного постоянного запоминающего устройства в модуле управления системой. Этот код вычитывает код из *system_flash*.
2. SCP производит основные настройки ВЕ-M1000.
3. SCP вычитывает код BL1 (часть ARM Trusted Firmware-A) из *boot_flash*.
4. Полученный код BL1 копируется в *mailbox*.
5. SCP даёт команду на исполнение кода BL1 из *mailbox* на ядре #0 кластера Cortex-A57 #0.

3 Требования для построения системы на ВЕ-M1000

В данном разделе перечислен минимальный набор компонентов и основные требования для построения системы на СнК ВЕ-M1000.

Названия сигналов указываются согласно документу ВЕ_M1000-xxx-Datasheet-xxx.pdf, доступному для скачивания на странице [Процессор ВЕ-M1000](#) официального сайта компании «БАЙКАЛ ЭЛЕКТРОНИКС».

3.1 system_flash

Обязательное требование: Должна быть реализована на микросхеме энергонезависимой памяти типа NOR (flash), подключенной к интерфейсу SPI0 с сигналом выбора Slave Select через вывод SPI0_SS_N[0], см. вывод AR19 в *Таблице 5-1 Полный список выводов* документа *ВЕ-M1000 Datasheet*.

Рекомендуемый объем *system_flash* - 8МБ, минимальный - 1МБ.

Образ для *system_flash* - файл `#{BRD_NAME}.flash0.img` (ближайший аналог `dbm.flash0.img`) – должен быть взят из папки `<...>/baikal/prebuilts` SDK ARM64.

3.2 boot_flash

Обязательное требование: Должна быть реализована на микросхеме энергонезависимой памяти типа NOR (flash), подключенной к интерфейсу SPI1 с комбинированным сигналом выбора Slave Select через вывод SPI1_SS_N[0] (вывод E37) и любой из выводов GPIO32[x]. Рекомендуемый минимальный объем 8МБ.



Если на `boot_flash` требуется поместить урезанную версию ОС ВЕ Linux, объем памяти `boot_flash` необходимо увеличить до 64МБ.

В этой памяти по нулевому смещению размещается код `boot_flash`.

Следующая таблица описывает структуру образа `boot_flash`.

Таблица 1 Структура образа `boot_flash`

Адресный регион	Описание
0x0000000 - 0x003EFFF	BL1
0x003F000 - 0x003FFFF	Board IDs
0x0040000 - 0x007FFFF	DTB (FDT blob)
0x0080000 - 0x013FFFF	UEFI vars
0x0140000 - 0x0AFFFFFF	FIP (blob (bl2,bl3.1,bl3.3))
0x0B00000 - 0x22FFFFFF	FAT file system (linux.image, board.dtb, initrd.gz, r.nsh)

Образ для `boot_flash` - файл `${BRD_NAME}.flash.img` (ближайший аналог `dbm.flash.img`) - может быть взят из папки `<...>/baikal/prebuilts SDK ARM64` или быть сгенерированным при помощи содержимого папки `<...>/baikal/img SDK ARM64`.

Для доступа к перепрограммированию `boot_flash` из операционной системы необходимо использовать комбинированный способ формирования сигнала «Slave Select» через вывод `SPI1_SS_N[0]` (E37) и один из сигналов `GPIO32[x]` для выбора именно данной микросхемы на шине SPI.

3.3 SPD DDR4 SDRAM

Обязательное требование: Для доступа к микросхемам **SPD** (*Serial Presence Detect*) на модулях памяти DDR4, они должны быть подключены к интерфейсу SMB1/I2C2 с соответствующим согласованием уровней.

Данный интерфейс используется компанией «БАЙКАЛ ЭЛЕКТРОНИКС» и будет поддерживаться в коде SDK.

3.4 RTC

RTC (*Real Time Clock*) - часы реального времени – это компонент системы, к наличию которого чувствительны все современные ОС. Необходимо предусмотреть интерфейс коммуникации с ВЕ-М1000 и разработать драйвер для UEFI.

Предпочтительней использовать I2C1, поскольку данный интерфейс используется компанией «БАЙКАЛ ЭЛЕКТРОНИКС» и будет поддерживаться в коде SDK.



3.5 BMC

BMC (*Board Management Controller*) - не является обязательным, но рекомендуется для использования в системе. Он предназначен для сервисного управления платформой. Необходимо предусмотреть интерфейс коммуникации с ВЕ-М1000 и разработать драйвер для ядра Linux и UEFI.

Предпочтительней использовать I2C1, поскольку данный интерфейс используется компанией «БАЙКАЛ ЭЛЕКТРОНИКС» и будет поддерживаться в коде SDK.

3.6 UART

Если требуется информация о процессе загрузки и вывод отладочной информации, рекомендуется использовать интерфейс UART1 (J38, J39).

Данный интерфейс используется компанией «БАЙКАЛ ЭЛЕКТРОНИКС» именно для этих целей и будет поддерживаться в коде SDK.

3.7 1Gb Ethernet PHY Reset

Для работы с 1 Gb Ethernet в микропроцессоре ВЕ-М1000 имеются два идентичных контроллера: **GMAC** (*1 Gb Ethernet Media Access Controller*) #0 и GMAC#1. Каждый из этих контроллеров поддерживает RGMII интерфейс с интерфейсом физического уровня PHY, установленным на плате. Для сброса PHY в GMAC рекомендуется использовать выходной сигнал G0_GP_OUT для GMAC#0 и G1_GP_OUT для GMAC#1.

Данный механизм сброса 1 Gb Ethernet PHY используется компанией «БАЙКАЛ ЭЛЕКТРОНИКС» и будет поддерживаться в коде SDK.

3.8 PCIe Reset

Для сброса внешних устройств, подключенных к PCIe интерфейсам, рекомендуется использовать любой из сигналов GPIO32[x].

Данный механизм сброса используется компанией «БАЙКАЛ ЭЛЕКТРОНИКС» и будет поддерживаться в коде SDK.