

ТЕХНИЧЕСКОЕ ОПИСАНИЕ
И РУКОВОДСТВО ПО ПОДКЛЮЧЕНИЮ
"Вычислитель-ВМ14Я COMExpress"
ООО СКТЬ «СКИТ»

Саратов, 2018г.

Наименование и назначение изделия

1.1 Полное наименование изделия и обозначение для заказа: модуль мезонинный «Вычислитель-ВМ14Я COMExpress РПАШ.467444.001». Для ознакомительных целей возможна поставка отладочной несущей платы, совместимой с изделием («плата регулировочная РПАШ.441461.001»).

1.2 Изделие предназначено для применения в качестве вычислительного и интерфейсного модуля, устройства приёма и обработки информации во встраиваемых системах и системах специального назначения.

Техническое описание

1.3 Архитектура изделия

Структурная схема Изделия приведена на рис. 1. Изделие выполнено в виде отдельного функционально законченного мезонинного модуля, который сопрягается с Носителем с помощью мезонинного разъёма.

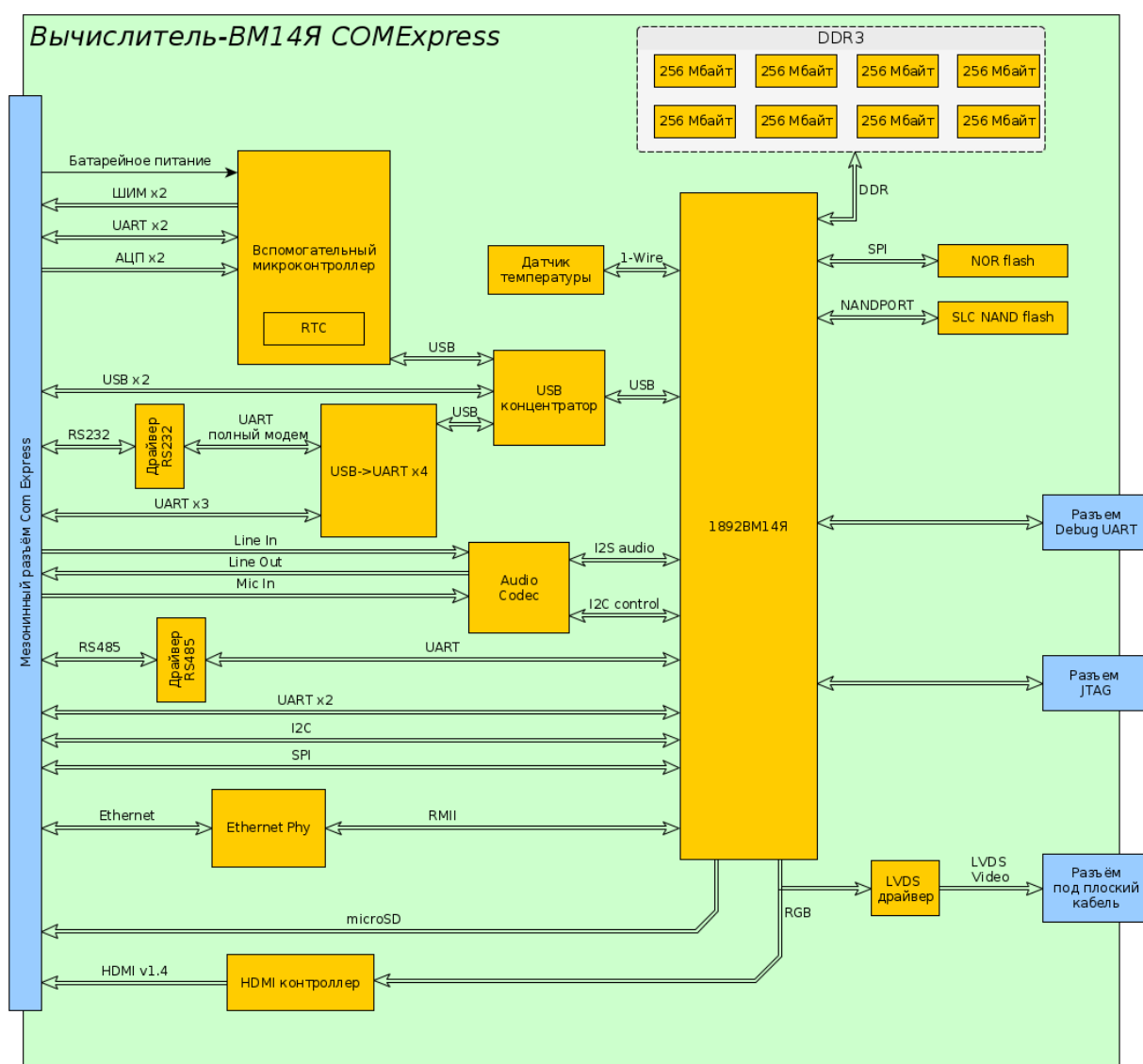


Рис.1 Структурная схема модуля «Вычислитель-ВМ14Я COMExpress»

Вычислительная часть Изделия основана на отечественном процессоре 1892BM14Я производства АО НПП «ЭЛВИС». Данный процессор имеет два ядра ARM Cortex-A9 с тактовой частотой не менее 744 МГц, а также двухядерный DSP-кластер ELcore-30M. К центральному процессору подключена оперативная память типа DDR3 объемом 2 Гбайта, ПЗУ типа NAND Flash объемом 4 Гбайта.

1.4 Описание внешних интерфейсов

Внешние интерфейсы Изделия перечислены в таблице 1.

Таблица 1

Наименование интерфейса	Кол-во	Примечание
Ethernet	1	Соответствует спецификации IEEE 802.3u 10/100BaseT/TX. Требуется гальванической развязки на плате-носителе. Расположение – мезонинный разъем.
USB	2	Соответствует спецификации USB 2.0. Расположение – мезонинный разъем.
UART с полным набором сигналов управления модемом	3	Логические уровни сигналов – LVTTTL. Расположение – мезонинный разъем.
UART с аппаратным управлением потоком	1	Логические уровни сигналов – LVTTTL. Расположение – мезонинный разъем.
UART, только данные	3	Логические уровни сигналов – LVTTTL. Расположение – мезонинный разъем.
RS232 с полным набором сигналов управления модемом	1	Уровни сигналов соответствуют стандарту RS-232. Расположение – мезонинный разъем.
RS485	1	Уровни сигналов соответствуют стандарту RS-485. Расположение – мезонинный разъем.
интерфейс карт памяти uSD	1	Соответствует спецификации SD Specification Version 3.00. Расположение – разъем на печатной плате Изделия.
HDMI	1	Соответствует спецификации HDMI v.1.4. Расположение – мезонинный разъем.
LVDS	1	Интерфейс LVDS видео, 3 линии данных и 1 тактовая линия. Расположение – разъем на печатной плате Изделия.
SPI	1	Логические уровни сигналов – LVTTTL. Подключение до 4 устройств. Расположение – мезонинный разъем.
I2C	1	Логические уровни сигналов – LVTTTL. Подтяжка линий интерфейса на плате Изделия. Расположение – мезонинный разъем.
ШИМ	2	Логические уровни сигналов – LVTTTL. Расположение – мезонинный разъем.
АЦП	2	Измеряемое напряжение от 0 до 3,3 В. Расположение – мезонинный разъем.
Комплект аудиосигналов: линейный выход, линейный вход, микрофонный вход	1	Расположение – мезонинный разъем.
Цифровые линии GPIO	10	Расположение – мезонинный разъем. Логические уровни сигналов – LVTTTL.

1.5 Требования к системе электропитания

Номинальное входное напряжение питания Изделия – 12 В, род тока – постоянный. Допустимый диапазон входного напряжения питания составляет – 7..18В, род тока – постоянный. Система электропитания Изделия реализована без гальванической изоляции.

1.6 Условия применения

Рабочий температурный диапазон Изделия – от минус 40 °С до плюс 70 °С.

1.7 Показатели надёжности

Изделие рассчитано на круглосуточную работу без обслуживания. Количественные показатели надёжности Изделия составляют:

- средняя наработка на отказ не менее 10000 ч;
- средний срок службы изделия не менее 10 лет.

1.8 Конструкция изделия

Тип мезонинного соединительного разъема, его расположение на печатной плате соответствует варианту COM Express Compact, назначение выводов частично соответствует типу COM Express Type 6 и приведено в приложении А. Конструкция Изделия предусматривает кондуктивный метод охлаждения посредством отвода тепла от теплонагруженных элементов Изделия через крышку на носитель без применения принудительного охлаждения.

1.9 Техничко-экономические особенности

С целью снижения стоимости Изделия и улучшения массогабаритных и функциональных показателей в Изделии применена электронная компонентная база импортного производства. Номенклатура ЭКБ ИП выбрана преимущественно без использования ЭРИ стран НАТО и их союзников.

1.10 Состав программного обеспечения

Изделие поставляется с предустановленным программным обеспечением в следующем составе:

- начальный загрузчик ОС UBoot;
- ОС Linux 4.1.27.4 или выше с комплектом драйверов для поддержки интерфейсов в соответствии с п.1.4.

Описание работы с предустановленным программным обеспечением приведено в приложении Б.

Приложение А. Описание внешних соединителей Изделия

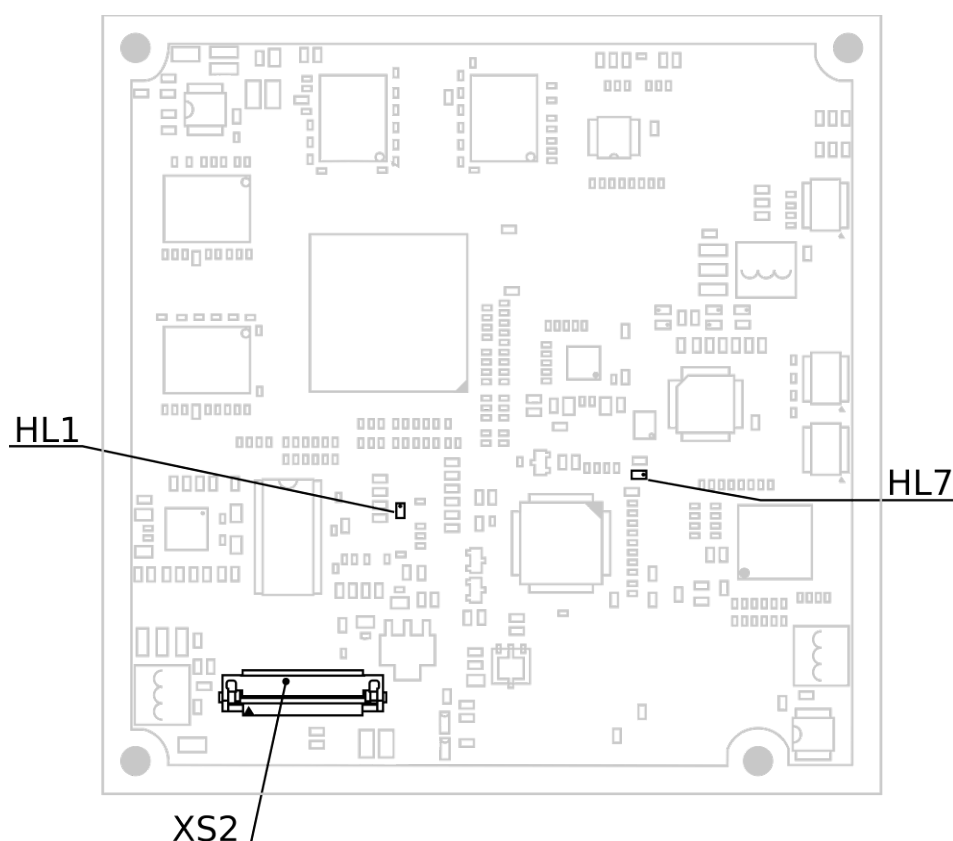


Рис.2 Внешний вид модуля «Вычислитель-ВМ14Я COMExpress» – вид сверху

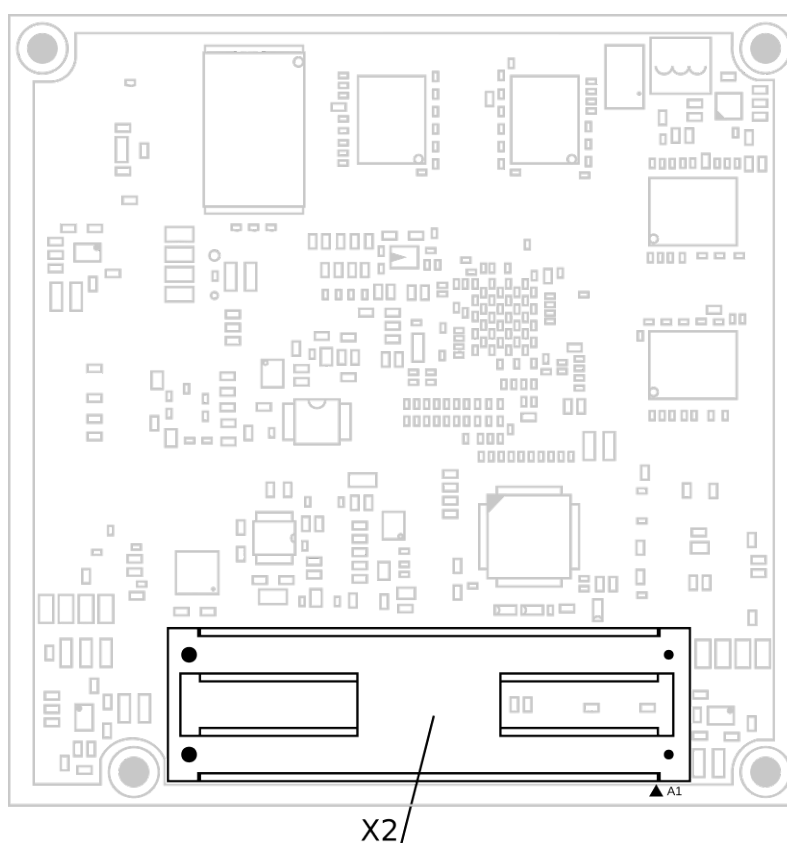


Рис.2 Внешний вид модуля «Вычислитель-ВМ14Я COMExpress» – вид снизу

Таблица А.1

Назначение	Обозначение	Тип	Ответная часть
Основной мезонинный соединитель	X2	Тусо 3-1827231-6	Foxconn QT002206-2141-3H Тусо 3-1827233-6
Соединитель для подключения дисплея по интерфейсу LVDS	XS2	Molex 52892-3033	Плоский кабель (FFC/FPC) с шагом 0,5 мм
Индикатор работы основного процессора	HL1	Светодиод	-
Индикатор работы вспомогательного контроллера	HL7	Светодиод	-

В таблице А.1 перечислены внешние соединители модуля и светодиоды индикации состояния. Назначение выводов основного мезонинного соединителя приведено в таблице А.2. В таблице А.3 приведено описание групп сигналов основного мезонинного соединителя. Назначение выводов соединителя для подключения дисплея по интерфейсу LVDS приведено в таблице А.4. Примечание – корпус мезонинного модуля электрически изолирован от всех внутренних цепей и может быть подключён к шине «корпус» несущей платы с помощью крепёжных элементов.

Таблица А.2

	A	B	C	D
1	GND	GND	GND	GND
2		ACT	GND	
3			+5V	AudioOut_L
4		+5V	+5V	AudioOut_R
5		+5V	GND	AudioGND
6		+5V	+5V	AudioIn_L
7	FXSD	+5V	+5V	AudioIn_R
8	LINK		GND	AudioMic_L
9	Eth_Rx-		+3.3V	AudioMic_R
10	Eth_Rx+		+3.3V	
11	GND	GND	GND	GND
12	Eth_Tx-	PWRBTN	+3.3V	
13	Eth_Tx+	+3.3V	+3.3V	
14	Eth_CTref	+3.3V	GND	GND
15		+3.3V		HDMI_DDCSCL
16	+1.5V	+3.3V		HDMI_DDCSDA
17	+1.1V			+5V_HDMI
18				HDMI_CEC
19				
20	BOOT0			
21	GND	GND	GND	GND
22	BOOT1			
23	BOOT2			
24			HDMI_HPD	
25	VideoSel			
26				HDMI_Tx2P
27				HDMI_Tx2M

	A	B	C	D
28				
29				HDMI_Tx1P
30				HDMI_Tx1M
31	GND	GND	GND	GND
32				HDMI_Tx0P
33		I2C_SCL		HDMI_Tx0M
34		I2C_SDA		
35			+1.8V	
36				HDMI_TxCP
37				HDMI_TxCM
38				
39				
40				
41	GND	GND	GND	GND
42				
43	USBD3_VBUS	USBD4_VBUS		
44	USB3_OC	USB4_OC	AGND	AGND
45	USBD3_DM	USBD4_DM	AI1	AI2
46	USBD3_DP	USBD4_DP	AGND	AGND
47	VCC_RTC		GND	GND
48			PWM1	PWM2
49		SYS_RESET		
50				
51	GND	GND	GND	GND
52				UART1_RTS
53				UART1_CTS
54	GPIO8	GPIO2	GND	UART1_RXD
55				UART1_TXD
56				
57	GND	GPIO6		GND
58				
59				
60	GND	GND	GND	GND
61				
62				
63	GPIO9	GPIO0		
64				
65				SDIO_CMD
66	GND			SDIO_DET
67	GPIO5		GPIO1	GND
68				SDIO_DA[0]
69				SDIO_DA[3]
70	GND	GND	GND	GND
71				SDIO_DA[2]
72				SDIO_DA[1]
73			GND	GND
74			UART3_CTS	
75			UART3_DTR	SDIO_CLK
76			GND	GND
77			UART3_DSR	UART3_RTS
78			UART3_DCD	UART3_RXD
79			UART3_RI	UART3_TXD
80	GND	GND	GND	GND

	A	B	C	D
81			UART5_RXD	
82			UART5_TXD	UART7_TXD
83			UART5_CTS	UART7_RXD
84			GND	GND
85	GPIO7		UART5_RTS	
86	GPIO4		UART5_RI	
87			GND	GND
88			UART5_DCD	RS485B
89	SPI_CS3		UART5_DSR	RS485A
90	GND	GND	GND	GND
91			UART5_DTR	
92	SPI_MISO		UART4_RI	RS232_TXD
93	GPIO3		GND	GND
94	SPI_SCK		UART4_DCD	RS232_RTS
95	SPI_MOSI		UART4_CTS	RS232_DTR
96			GND	GND
97		SPI_CS0	UART4_RTS	RS232_DCD
98	UART0_TXD	SPI_CS1	UART4_RXD	RS232_RI
99	UART0_RXD	SPI_CS2	UART4_TXD	RS232_RXD
100	GND	GND	GND	GND
101	UART6_TXD		UART4_DSR	RS232_CTS
102	UART6_RXD		UART4_DTR	RS232_DSR
103			GND	GND
104	+12V	+12V	+12V	+12V
105	+12V	+12V	+12V	+12V
106	+12V	+12V	+12V	+12V
107	+12V	+12V	+12V	+12V
108	+12V	+12V	+12V	+12V
109	+12V	+12V	+12V	+12V
110	GND	GND	GND	GND

Таблица А.3

Группа сигналов	Описание
GND	цифровая шина «общий»
+12V	входное питание 12В
+5V	внутреннее питание модуля +5В, только для питания элементов защиты для внешних интерфейсов и согласования уровней сигналов
+3.3V	внутреннее питание модуля +3,3В, только для питания элементов защиты для внешних интерфейсов и согласования уровней сигналов
Eth_Rx+, Eth_Rx-, Eth_Tx+, Eth_Tx-, Eth_CTref, LINK, ACT	интерфейс Ethernet, требует гальванической развязки; сигнал Eth_CTref – питание средней точки развязывающего трансформатора; сигналы LINK и ACT имеют низкий активный уровень
GPIO0...GPIO9	дискретные сигналы общего назначения
AudioOut_*, AudioIn_*, AudioMic_*	входной и выходной аудио интерфейсы, а также вход для подключения электретного микрофона
AudioGND	шина «общий» аудиоинтерфейсов
HDMI_*, +5V_HDMI	видеоинтерфейс HDMI
USB*_VBUS, USB*_OC, USB*_DM, USB*_DP	интерфейсы USB; сигналы *_VBUS – 5В для питания устройств; сигналы *_OC – сигналы о перегрузке по питанию интерфейса USB (при наличии на несущей плате схем контроля тока), активный низкий уровень
I2C_SCL, I2C_SDA	интерфейс I2C

Группа сигналов	Описание
SPI_SCK, SPI_MOSI, SPI_MISO, SPI_CS0...3	интерфейс SPI, до 4-х устройств
SDIO_*	интерфейс для подключения карты памяти типа uSD ¹⁾ .
PWRBTN	входной сигнал включения питания, активный низкий уровень
SYS_RESET	входной сигнал сброса, активный низкий уровень
VCC_RTC	батарейное питание часов реального времени
AI1, AI2	аналоговые входы, диапазон напряжений от 0В до 3,3В
AGND	аналоговая шина «общий»
PWM1, PWM2	выходные ШИМ-сигналы
UART0_*, UART6_*, UART7_*	интерфейсы UART, только данные
UART1_*	интерфейс UART с аппаратным управлением потоком
UART3_*, UART4_*, UART5_*	интерфейсы UART с сигналами управления модемом
RS485A, RS485B	интерфейс RS-485
RS232_*	интерфейс RS-232

¹⁾ – для корректной работы «горячего» подключения карты памяти необходимо, чтобы сигнал SDIO_DET при вставленной карте был в низком уровне, без карты – в высоком уровне.

Таблица А.4

№ контакта	Название цепи	Назначение
1	+3.3V	питание +3,3 В
2	+3.3V	
3	+3.3V	
4		нет подключения
5	GND	цифровая шина «общий»
6	TxO0-	видеовыход. Дифференциальная пара данных 0.
7	TxO0+	
8	GND	цифровая шина «общий»
9	TxO1-	видеовыход. Дифференциальная пара данных 1.
10	TxO1+	
11	GND	цифровая шина «общий»
12	TxO2-	видеовыход. Дифференциальная пара данных 2.
13	TxO2+	
14	GND	цифровая шина «общий»
15	TxCLK-	видеовыход. Дифференциальная пара CLK.
16	TxCLK+	
17	GND	цифровая шина «общий»
18		нет подключения
19		
20	GND	цифровая шина «общий»
21		нет подключения
22		
23	GND	цифровая шина «общий»
24	LVDS_BKLT_CTRL	включение и регулировка яркости подсветки
25		нет подключения
26		
27	+5V	питание +5 В
28	+5V	
29	+5V	
30	+5V	

Приложение Б. Описание работы с программным обеспечением Изделия

Б.1 Требования к ПК, используемому для разработки

Для корректной работы инструментов разработчика (далее SDK), используемых для разработки и отладки прикладного программного обеспечения (далее ПО) требуется операционная система Ubuntu для архитектуры x86_64(amd64). Для корректной работы SDK в системе должен быть установлен пакет build-essential:

```
$ sudo apt-get install build-essential
```

Б.2 Установка SDK

Для начала работы с Изделием необходимо установить инструменты разработчика, которые доступны по ссылке ftp://ftp.skitlab.ru/tools/vizit/sdk/vm14_sdk.tar.bz2 и включают в себя:

- кросс-компилятор GCC, используемый для сборки разрабатываемого ПО;
- отладчик GDB для удалённой отладки;
- система сборки CMake;
- ключи для доступа к плате по протоколу SSH;

Для установки SDK необходимо распаковать архив vm14_sdk.tar.bz2 в каталог /opt:

```
$ tar -xvjf /path/to/vm14_sdk.tar.bz2 -C /opt
```

Далее необходимо добавить путь к исполняемым файлам компилятора в список путей переменной окружения PATH:

```
$ echo "export PATH=$PATH:/opt/vm14_sdk/usr/bin" >>  
~/.bash_profile
```

После изменения переменной PATH, необходимо повторно войти в систему.

Б.3 Настройка доступа к Изделию по протоколу SSH

Для доступа к командной консоли изделия, а также для передачи файлов между ПК разработчика и изделием используется подключение по протоколу SSH через интерфейс Ethernet.

Для доступа к устройству по протоколу SSH необходимо настроить аутентификацию по ключам. На изделии предустановлен публичный ключ, секретная пара к которому поставляется в составе SDK в файле id_rsa. Для доступа к изделию по предустановленному сетевому адресу изделия 192.168.200.25, файл с секретной частью ключа необходимо указать в команде следующим образом:

```
$ ssh -i id_rsa root@192.168.200.25
```

После успешного выполнения данной команды выдаётся приглашение консоли изделия:

```
# _
```

Передача файлов между ПК разработчика и домашним каталогом пользователя root изделия выполняется следующим образом(команды выполняются на ПК):

```
$ scp -i id_rsa /path/to/file root@192.168.200.25:/root/
```

Соединение с изделием может быть установлено после окончания загрузки, о чем свидетельствует мигающий светодиод HL1.

Б.4 Сборка и запуск прикладного ПО на изделии

Поставляемым SDK поддерживается сборка ПО, написанного на языках C/C++ с использованием следующих библиотек:

- libc, стандартная библиотека языка C (реализация glibc);
- libstdc++, стандартная библиотека языка C++;
- libasound, библиотека доступа к звуковой подсистеме ALSA;
- оконная система Xorg;
- Qt, многофункциональная библиотека высокого уровня для C++. Включены модули Core, Widgets, Network, Sql, Xml;
- libiiio, библиотека доступа к АЦП.

Сборка примеров программ, поставляемых в составе SDK в каталоге examples, выполняется при помощи программы CMake.

Для использования CMake совместно с кросс-компилятором предназначен файл настройки инструментария (arm-toolchain.cmake), расположенный в каталоге SDK. В этом файле необходимо исправить пути в переменных CMAKE_FIND_ROOT_PATH и CMAKE_C_COMPILER в соответствии с реальным расположением SDK.

Далее необходимо сконфигурировать проект командой, предварительно перейдя в каталог examples:

```
$ cmake -DCMAKE_TOOLCHAIN_FILE=/path/to/arm-toolchain.cmake
```

В результате будет создан скрипт сборки Makefile, и можно выполнить сборку проекта командой:

```
$ make
```

В результате выполнения данной команды в каталоге examples будут созданы исполняемые файлы примеров программ, которые необходимо скопировать на изделие как указано в разделе Б.3 и выполнить запуск примера командой из консоли изделия:

```
# cd /root/
# ./gpio
```

Прервать выполнение примера можно комбинацией клавиш Ctrl-C.

Б.5 Создание новых проектов с использованием CMake

Для создания проекта программы для изделия с использованием системы сборки проектов CMake необходимо создать файл описания проекта CMakeLists.txt. Данный файл содержит описание проекта на командном языке CMake, подробнее о котором можно прочитать на официальном сайте CMake: <https://cmake.org/cmake/help/v3.5/manual/cmake-buildsystem.7.html>.

Б.6 Управление линиями ввода-вывода GPIO из пользовательских программ

Управление линиями GPIO из пользовательских программ Linux производится посредством специализированной файловой системы SysFS, которая смонтирована в каталоге `/sys` корневой файловой системы изделия.

Для получения,прекращения доступа к конкретной линии GPIO служат файлы SysFS `/sys/class/gpio/export` и `/sys/class/gpio/unexport`. При записи в них номера линии в текстовом формате (в виде последовательности цифровых символов) производится соответственно захват либо освобождение линии GPIO.

Процессор 1892BM14Я имеет четыре порта GPIO, под каждый из которых в Linux зарезервировано 32 линии. Соответствие базовых номеров линий портам процессора приведено в таблице:

Базовый номер линии	Порт процессора
480	GPIOA0
448	GPIOB0
416	GPIOC0
384	GPIOD0

Вычисление номеров остальных выводов GPIO процессора производится добавлением номера вывода к соответствующему базовому адресу. Например при выполнении команды:

```
# echo 486 > /sys/class/gpio/export
```

Будет произведён захват управления для линии GPIOA6 процессора 1892BM14Я. При этом будет создан каталог `/sys/class/gpio/gpio486/`, содержащий управляющие файлы для линии 486.

Направление передачи для линии задается записью строк «out» для выхода или «in» для входа в файл `/sys/class/gpio/gpio486/direction`. Состояние линии, переключённой в режим выхода, устанавливается записью «0» или «1» в файл `/sys/class/gpio/gpio486/value`. Следующие команды установят линию процессора GPIOA6 в состояние выхода высокого уровня.

```
# echo out > /sys/class/gpio/gpio486/direction
# echo 1 > /sys/class/gpio/gpio486/value
```

Соответствие внешних линий GPIO изделия линиям процессора представлено в таблице:

Внешний вывод	Линия процессора	Номер линии в Linux
GPIO0	GPIOA6	486
GPIO1	GPIOA7	487
GPIO2	GPIOA8	488
GPIO3	GPIOA9	489
GPIO4	GPIOA10	490
GPIO5	GPIOA11	491
GPIO6	GPIOA12	492
GPIO7	GPIOA13	493
GPIO8	GPIOA14	494
GPIO9	GPIOA15	495

Б.7 Использование последовательных интерфейсов UART, RS-232 и RS-485

Изделие имеет набор последовательных интерфейсов, основанных на UART, управление которыми производится через подсистему tty Linux. Управление интерфейсами производится посредством файлов устройств, расположенных в каталоге */dev/* корневой файловой системы изделия. Соответствие портов и файлов устройств приведено в таблице:

Интерфейс	Файл устройства	Примечание
UART1	<i>/dev/ttyUSB2</i>	
UART2	<i>/dev/ttyUSB3</i>	
UART3	<i>/dev/ttyUSB1</i>	
UART4	<i>/dev/ttyS1</i>	
UART5	<i>/dev/ttyM0</i>	
UART6	<i>/dev/ttyS2</i>	
UART7	<i>/dev/ttyM1</i>	
RS-232	<i>/dev/ttyUSB0</i>	
RS-485	<i>/dev/ttyS3</i>	

Для начала обмена через любой из этих интерфейсов сначала необходимо его сконфигурировать. Это может быть сделано командой stty:

```
# stty -F /dev/ttyS2 speed 115200 clocal
```

Команда stty имеет дополнительные опции, описание которых можно посмотреть по ссылке: <https://www.opennet.ru/man.shtml?topic=stty&category=1>.

Настройка последовательного интерфейса из программы на языках C/C++ производится посредством функции tcsetattr стандартной библиотеки языка C. После настройки порта обмен через него в программе на языке C может производиться стандартными системными вызовами read и write.

Из консоли запись в порт может быть выполнена посредством перенаправления:

```
# cat /etc/fstab > /dev/ttyS2
```

Чтение из порта в консоли также может быть выполнено командой cat:

```
# cat /dev/ttyS2
```

Интерфейс RS-485 работает в режиме полудуплекса, причем управление направлением передачи производится автоматически драйвером порта при вызове функций read или write.

Б.8 Работа с устройствами, подключёнными по шине I2C

Работа с шиной I2C производится через файлы устройств в каталоге */dev/* корневой файловой системы изделия. На внешний разъем изделия выведена шина I2C, которой соответствует файл устройства */dev/i2c-0*.

Для работы с устройствами на шине I2C из консоли изделия существуют команды i2cget, i2cset, i2cdetect и i2cdump. Первые две команды предназначены для чтения и записи значений регистров в устройствах I2C. Считывание регистра с адресом 0x12 устройства с адресом 0x35 производится командой:

```
# i2cget -y 0 0x35 0x12
```

Запись значения 0x33 в тот же регистр:

```
# i2cset -y 0 0x35 0x12 0x33
```

При указании адреса устройства необходимо иметь ввиду, что в команде указывается поле адреса устройства, а не весь адресный байт I2C целиком, в котором младший бит отвечает за направление передачи.

Команда `i2cdetect` предназначена для определения устройств, подключенных к шине I2C, а команда `i2cdump` выводит значения всех регистров какого-либо устройства.

Программный интерфейс для обмена по шине I2C с использованием языков C/C++ описан по ссылке: <https://www.kernel.org/doc/Documentation/i2c/dev-interface>.

Б.9 Работа с устройствами, подключёнными по шине SPI

Работа с шиной SPI производится через файлы устройств в каталоге `/dev/` корневой файловой системы изделия. На внешний разъем изделия выведена шина SPI с возможностью подключения до четырех устройств. Соответственно, в каталоге `/dev/` имеются четыре устройства, каждое из которых соответствует одному сигналу выбора устройства шины SPI: `/dev/spidev32766.0`, `/dev/spidev32766.1`, `/dev/spidev32766.2` и `/dev/spidev32766.3`.

Командная консоль изделия не содержит специальных команд для взаимодействия с шиной SPI, однако для проверки можно использовать перенаправления и конвейеры оболочки:

```
# cat /dev/urandom > /dev/spidev32766.0
# cat /dev/spidev32766.0
```

При этом передача будет вестись с параметрами шины по-умолчанию. Если же необходимо изменить параметры шины (число бит в слове, активный уровень тактового сигнала для чтения и записи), то необходимо воспользоваться программным интерфейсом для языка C, описание которого доступно по ссылке: <https://www.kernel.org/doc/Documentation/spi/spidev>.

Б.10 Работа с каналами АЦП

Изделие содержит два канала АЦП, предназначенных для контроля медленно меняющихся процессов. Управление каналами АЦП производится посредством интерфейса ПО ядра Linux. Взаимодействие с данным интерфейсом возможно как через SysFS, так и через библиотеку `libiio`.

Считывание значение АЦП по входу AI1 через SysFS производится командой:

```
# cat /sys/bus/iio/devices/iio\:device0/in_voltage0_raw
```

Считывание значение АЦП по входу AI2 через SysFS производится командой:

```
# cat /sys/bus/iio/devices/iio\:device1/in_voltage0_raw
```

Значение выводится в квантах АЦП. Один квант АЦП равен 0,000802001953125 В.

Работа с библиотекой `libiio` описана по ссылке: <http://analogdevicesinc.github.io/libiio/>.

Б.11 Управление каналами ШИМ

Изделие может формировать до двух каналов широтно-импульсной модуляции (ШИМ).

Для управления параметрами формируемого сигнала необходимо, как и в случае с GPIO осуществить захват управления каналом ШИМ. Для этого нужно выполнить следующую команду для первого канала ШИМ:

```
# echo 0 > /sys/class/pwm/pwmchip0/export
```

Для второго канала следует выполнить команду:

```
# echo 0 > /sys/class/pwm/pwmchip1/export
```

Далее можно произвести установку параметров формируемого сигнала при помощи записи в файлы в каталогах `/sys/class/pwm/pwmchip0/pwm0/` и `/sys/class/pwm/pwmchip1/pwm0/` для первого и второго каналов соответственно. Эти файлы:

- `enable` — запись «1» в него разрешает формирование соответствующего сигнала, запись «0» — запрещает;
- `period` — записанное в него в текстовом формате (последовательность цифровых символов) число задает период генерации импульсов в наносекундах;
- `duty_cycle` — записанное в него в текстовом формате число задает длительность импульса в наносекундах.

Пример:

```
# echo 1 > /sys/class/pwm/pwmchip0/pwm0/enable
# echo 10000 > /sys/class/pwm/pwmchip0/pwm0/period # 10 мкс
# echo 2000 > /sys/class/pwm/pwmchip0/pwm0/duty_cycle # 2 мкс
```

Освободить канал ШИМ можно записав «0» в файлы `/sys/class/pwm/pwmchip0/unexport` и `/sys/class/pwm/pwmchip1/unexport` для первого и второго канала соответственно.

Б.12 Считывание значения с датчика температуры

На устройстве установлен датчик температуры, подключаемый по каналу 1-Wire. Для считывания значения температуры необходимо считать содержимое файла `/sys/bus/w1/devices/28-xxxxxxxxxx/w1_slave`, где `xxxxxxxxxxxx` — серийный номер микросхемы термодатчика, уникальный для каждого изделия. При считывании выводится следующая информация:

```
# cat /sys/bus/w1/devices/28-000000d7970b/w1_slave
7c 01 4b 46 7f ff 04 10 09 : crc=09 YES
7c 01 4b 46 7f ff 04 10 09 t=23750
```

Здесь значение, указанное после «t=», является температурой в тысячных долях градуса.

Б.13 Использование портов подключения дисплеев

Изделие имеет два порта подключения графических дисплеев:

- порт LVDS с глубиной цвета 18 бит. Подключение дисплея LVDS осуществляется к разъему XS2, назначение выводов которого приведено в табл. А.4 приложения А;
- порт HDMI версии 1.4.

Так как процессор содержит один графический контроллер, то выбор активного видеовыхода осуществляется сигналом VideoSel (контакт A25 соединителя X2) при загрузке ОС. При подаче лог. «0» на сигнал VideoSel активным интерфейсом становится LVDS. При неподключенном сигнале VideoSel вывод видео производится через интерфейс HDMI.

Установка разрешения экрана осуществляется утилитой fbset:

```
# fbset 1280x720-60
```

Утилита позволяет установить следующие разрешения: 640x480-60, 800x600-60, 1024x768-60, 1366x768-60, 1280x720-60, 1280x800-60, 1920x1080-60.

Формирование изображения производится через интерфейс фреймбуфера, который доступен через файл устройства `/dev/fb0`.

Чтобы протестировать формирование изображения, необходимо загрузить на изделие в соответствии с разделом Б.3 файл изображения в одном из форматов PNG, BMP или JPEG и воспользоваться командой fbv:

```
# fbv image.jpg
```

После выполнения команды, изображение должно появиться на выбранном экране. Исходный код данной утилиты, доступный по ссылке <https://github.com/godspeed1989/fbv>, можно рассматривать в качестве примера программирования фреймбуфера на языке С.

Более рациональным подходом является использование для программирования вывода изображения, в том числе графического пользовательского интерфейса, высокоуровневой библиотеки Qt.

Б.14 Использование шины USB

Изделие позволяет подключить до двух внешних устройств с интерфейсом USB версии 2.0. Поддерживаются скорости подключения 1.5Мбит/с, 12 Мбит/с и 480Мбит/с. Операционная система изделия содержит драйвера для следующих типов USB-устройств:

- USB HID устройства (клавиатура, мышь)
- USB Mass storage class (Накопители FLASH)
- USB UART cp210x

Посмотреть список подключённых USB-устройств можно командой lsusb.

Б.15 Использование карт памяти microSD

Изделие поддерживает подключение карты формата microSD, в том числе во время работы. При подключении SD-карты в каталоге `/dev/` корневой файловой системы изделия создаётся несколько файлов с префиксом mmcblk0.

Файл `mmcblk0` – блочное устройство, которое ссылается на первый сектор SD-карты. Также могут быть созданы один или более устройств `mmcblk0pN` (где N – число 1,2,3...), каждое из которых ссылается на первый сектор соответствующего раздела карты.

Для использования файловой системы на разделе карты, он должен быть примонтирован в какой-либо каталог корневой файловой системы:

```
# mount -t vfat /dev/mmcblk0p1 /mnt/
```

После выполнения данной команды содержимое файловой системы на разделе SD-карты будет доступно в каталоге `/mnt/`.

Также возможна загрузка ОС изделия с карты microSD содержащей специально подготовленное программное обеспечение, которое поставляется вместе с инструкциями по его установке по отдельному запросу. Для загрузки изделия с microSD-карты необходимо подключить интерфейс карты памяти к линиям SDIO_* соединителя X2, соединить контакты A20 и A22 соединителя X2 с цепью GND, а A23 – с цепью +3.3V, после чего подать питание на изделие.

Б.16 Проигрывание и запись аудио

Изделие имеет линейный аналоговый вход, который позволяет записывать аудиосигналы с внешних источников. Также изделие имеет линейный аналоговый выход, который позволяет воспроизводить аудио. Для тестирования аудио на изделии предустановлены следующие программы: `mplayer`, `aplay` и `arecord`.

Программа `mplayer` может воспроизводить аудио и видео в различных форматах. Запуск программы:

```
# mplayer file.mp3
```

Программа `arecord` позволяет записать аудио сигнал в файл без сжатия:

```
# arecord -r 48000 -f S16_LE file.snd
```

Записанный файл не имеет заголовка и содержит только аудиоданные. Воспроизвести его можно командой:

```
# aplay -r 48000 -f S16_LE file.snd
```

Регулировка громкости записи и воспроизведения может быть произведена программой:

```
# alsamixer
```

Данная программа отображает ползунок регулировки громкости воспроизведения при нажатии на клавишу F3 и ползунок регулировки громкости записи при нажатии на клавишу F4. Перемещение ползунков осуществляется стрелками «Вверх» и «Вниз».

Запись и воспроизведение звука из программы на С производится с использованием библиотеки `libasound`, с документацией на которую можно ознакомиться по ссылке: <http://www.alsa-project.org/alsa-doc/alsa-lib/>.

Б.17 Обмен данными по интерфейсу Ethernet

Изделие поддерживает обмен данными по интерфейсу Ethernet с использованием стека протоколов TCP/IP. Для организации обмена по протоколу TCP/IP, необходимо использовать BSD Socket API, который является частью библиотеки `libc`.

Б.18 Обновление программного обеспечения

Изделие содержит средства удаленного обновления всех компонентов встроенного программного обеспечения. Обновление производится через интерфейс Ethernet по протоколу FTP.

Для начала процедуры обновления необходимо перевести изделие в сервисный режим. Для этого необходимо произвести загрузку системы, предварительно подключив контакт B49 соединителя X2 к цепи «GND» на время загрузки ОС.

При загрузке в сервисном режиме доступ через SSH не работает, и консоль изделия выводится на интерфейс UART6.

После успешной загрузки в сервисном режиме, после появления приглашения необходимо войти в систему, введя логин «**root**».

```
Welcome to Service Console
MOD login:
```

В зависимости от конфигурации сети, к которой подключено изделие, необходимо выполнить настройку сети. По-умолчанию программное обеспечение сервисного режима пытается настроить сеть автоматически, используя протокол DHCP. Однако, если в сети отсутствует DHCP-сервер, то необходимо выполнить настройку вручную:

- Установить IP-адрес интерфейса:

```
# ifconfig eth0 192.168.xxx.xxx up
```

- Установить шлюз по-умолчанию:

```
# route add default 192.168.xxx.yyy eth0
```

- Установить адрес DNS-сервера:

```
# echo "nameserver 192.168.xxx.zzz" > /etc/resolv.conf
```

Далее следует обновить ПО изделия. ПО изделия состоит из нескольких компонентов, каждый из которых обновляется при помощи своей команды:

- Обновление образа первичного загрузчика:

```
# umtd spl ftp://ftp.skitlab.ru/updates/vizit/with_x_4g/
```

- Обновление образа основного загрузчика:

```
# umtd ldr ftp://ftp.skitlab.ru/updates/vizit/with_x_4g/
```

- Обновление образа ядра Linux:

```
# umtd krn ftp://ftp.skitlab.ru/updates/vizit/with_x_4g/
```

- Обновление образа корневой файловой системы:

```
# umtd nfs ftp://ftp.skitlab.ru/updates/vizit/with_x_4g/
```

- Прошивка вспомогательного контроллера:

```
# umtd mcu ftp://ftp.skitlab.ru/updates/vizit/with_x_4g/
```

Б.19 Завершение работы системы и перезагрузка

Накопитель NAND flash, установленный в изделии, содержит файловую систему с поддержкой записи UBIFS. Данная файловая система обладает устойчивостью к однократным нештатным отключениям питания. Тем не менее, для уменьшения вероятности отказа вследствие нарушения целостности файловой системы, её необходимо корректно деинициализировать (размонтировать) перед снятием питания с изделия.

Для корректного завершения работы системы следует использовать команду:

```
# poweroff
```

Для корректного завершения работы системы и последующей перезагрузки следует использовать команду:

```
# reboot
```