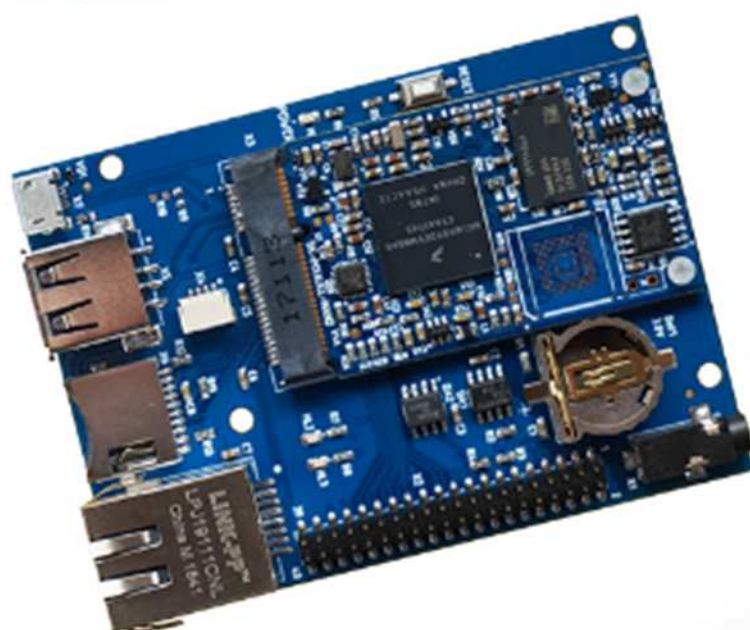


SK-IMX6ULL-NANO

ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ ПРИ СОВМЕСТНОМ
ИСПОЛЬЗОВАНИИ С ПЛАТОЙ SK-IMX6ULL-NANO-MB



SK-iMX6ULL-NANO

ПРОЦЕССОРНЫЙ МОДУЛЬ SK-IMX6ULL-NANO

- NXP(Freescale) ARM Cortex-A7 800МГц
- DDR3 256МБайт, DDR-800
- QSPI Flash 16МБайт
- 100/10М Ethernet PHY
- Интерфейсы: USB, Ethernet, 7xUART, SPI, I2C, PWM, GPIO, CAN ... общее количество возможных GPIO – 32
- Габариты: mini PCI-e форм-фактор 30x51x3мм
- Температурный диапазон -40 ... +85С

МАТЕРИНСКАЯ ПЛАТА SK-IMX6ULL-MB

Материнская плата SK-iMX6ULL-NANO-MB предназначена для совместного использования с процессорным модулем SK-iMX6ULL-NANO, содержит:

- Ethernet 100/10М, USB-Host, USB-OTG, MicroSD держатель, держатель батареи для RTC
- Разъем для подключения SK-FT230, EV-FT230 – USB-UART консоль
- RS485 PHY, CAN PHY
- Аудио выход, реализован на основе MQS выхода процессора

ВОЗМОЖНОСТЬ ПОДКЛЮЧЕНИЯ МОДУЛЕЙ РАСШИРЕНИЯ

- SK-SIM800C-Plug – модуль расширения GSM\Bluetooth

ПИТАНИЕ

Напряжение питания 5В. **Внимание!!!** Обязательно от стабилизированного источника питания!

Потребляемый ток в зависит от подключаемой периферии, сам модуль потребляет не более 0,3А.

С модуля можно получить напряжение питание 3,3В нагрузкой не более 0,4А.

ПЕРВОЕ ВКЛЮЧЕНИЕ

Из-за соображений технологичности производства, в составе системы отгружаемых изделий содержится минимальный набор утилит и сервисов, перед началом работы необходимо записать полную версию, см. раздел “Программирование QSPI Flash”.

В минимальном варианте, достаточно подключить USB кабель к разъему X7, после загрузки системы USB OTG порт платы перейдет в режим виртуального COM порта. При первом запуске потребуется установить драйвер USB-COM порта (из папки Linux/USB-COM_driver). Неудобство заключается в необходимости закрывать-открывать терминальную программу при каждом перезапуске платы. Определите в диспетчере устройств номер COM порта, который присваивается и откройте его терминальной программой (например Putty):

```
Welcome to SK-iMX6ULL
```

```
buildroot login:
```

Для получения доступа введите, логин: root, пароль: root

SK-iMX6ULL-NANO

В системе настроены сервисы FTP и SSH, доступ к которым можно получить, подключив плату Ethernet кабелем, IP адрес платы 192.168.0.136.

SK-IMX6ULL-NANO, СПОСОБЫ ЗАГРУЗКИ

Источник загрузки модуля – встроенная QSPI flash, выбор источника загрузки осуществлен программированием eFuse на этапе производства.

Штатная версия корневой файловой системы интегрирована в ядро и размещается в динамической памяти после загрузки, соответственно, все изменения КФС и сохраненные файлы будут потеряны после перезагрузки.

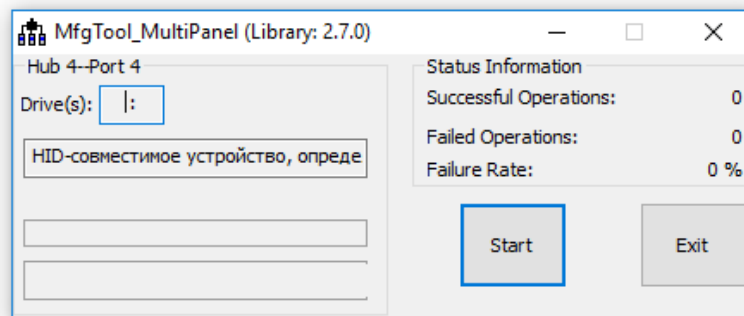
Запись во встроенную QSPI flash может осуществляться через USB-OTG порт.

ПРОГРАММИРОВАНИЕ QSPI FLASH

Для программирования используется утилита MFGTools, предварительно необходимо распаковать архив.

Замкните между собой контакты посадочного места J1 на самом модуле (это необходимо только на этапе сброса или включения питания), подключите USB кабель к разъему X7. Windows должна обнаружить новое HID устройство (установка дополнительных драйверов не требуется).

Запустите MfgTool2.exe, должно получиться:



Нажмите кнопку «Start», в консоли будет отображаться рабочий процесс, после завершения отключите-включите питание или нажмите кнопку “RESET” (J1 должен быть разомкнут).

SK-iMX6ULL-NANO

ВИРТУАЛЬНАЯ МАШИНА, ОСНОВНЫЕ ПОЛОЖЕНИЯ

Виртуальная машина VMware предназначена для сборки ядра Linux и корневой файловой системы без необходимости выделять для этого отдельный PC.

Виртуальная машина основана на Libuntu 14.10.

Сборка ядра Linux и КФС интегрирована и осуществляется в одном пакете **Buildroot**. **Buildroot** (www.buildroot.org) это инструмент для сборки КФС, содержит более тысячи приложений и библиотек, состав выбирает сам пользователь.

Для удобства, в виртуальной машине установлены и настроены сервисы для взаимодействия с внешним окружением

- **FTP сервер**
- **TFTP сервер**
- **SSH сервер**
- **Samba** – взаимодействие с сетями Microsoft
- **QtCreator**

QtCreator – среда разработки и отладки Qt приложений, имея Ethernet соединение (TCP/IP канал связи можно организовать практически через любой интерфейс процессора) с платой можно одним щелчком мыши собирать и запускать приложение на плате или вести пошаговую отладку.

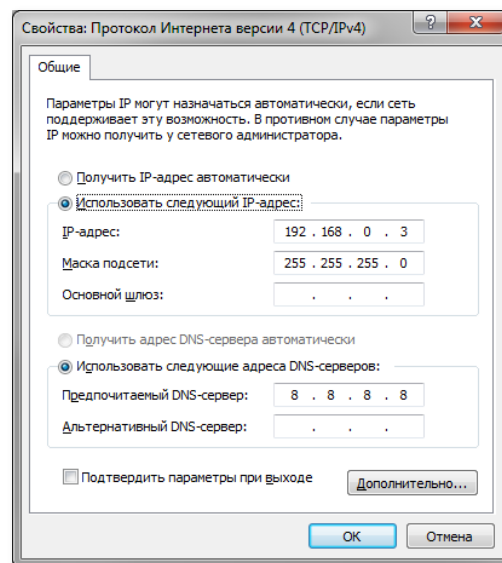
Qt наиболее предпочтительный пакет для написания приложений, т.к. в нем существует огромное количество библиотек и примеров, имеет широкую распространенность, графические приложения Qt работают значительно быстрее и потребляют гораздо меньше ресурсов (особенно памяти) системы в сравнении с приложениями X11.

НАСТРОЙКА ВИРТУАЛЬНОЙ МАШИНЫ

Перед началом работы необходимо скачать плеер виртуальной машины VMware, бесплатно распространяемый на сайте www.vmware.com.

Виртуальная машина имеет 2 сетевых интерфейса:

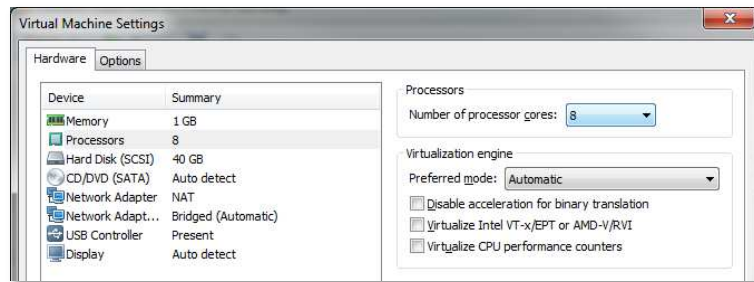
- 1) NAT – для доступа в Internet
- 2) Bridget – для взаимодействия по локальной сети, необходимо настроить VMware network adapter



SK-iMX6ULL-NANO

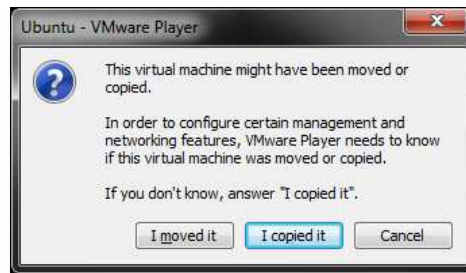
Так же необходимо настроить сетевой адаптер PC (или DHCP роутера), так, чтобы присваивался сетевой адрес в группе 192.168.0.XXX (любой кроме 1-3 и 136).

Перед запуском виртуальной машины рекомендуем зайти в ее свойства и выделить количество используемых процессорных ядер (по умолчанию, указан 1), это позволит ощутимо сократить время сборки.



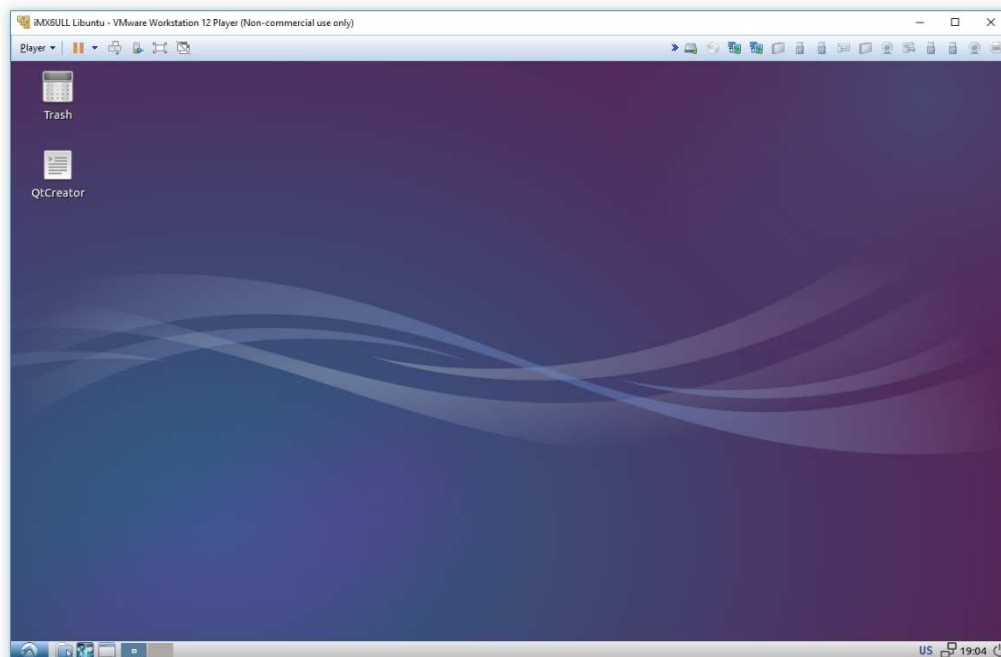
ПЕРВЫЙ ЗАПУСК ВИРТУАЛЬНОЙ МАШИНЫ

При первом запуске виртуальной машины (или после копирования-перемещения), VMware Player спросит:



Нужно ответить «I moved it», это позволит сохранить сетевые настройки.

После загрузки перед вами появится рабочий стол.

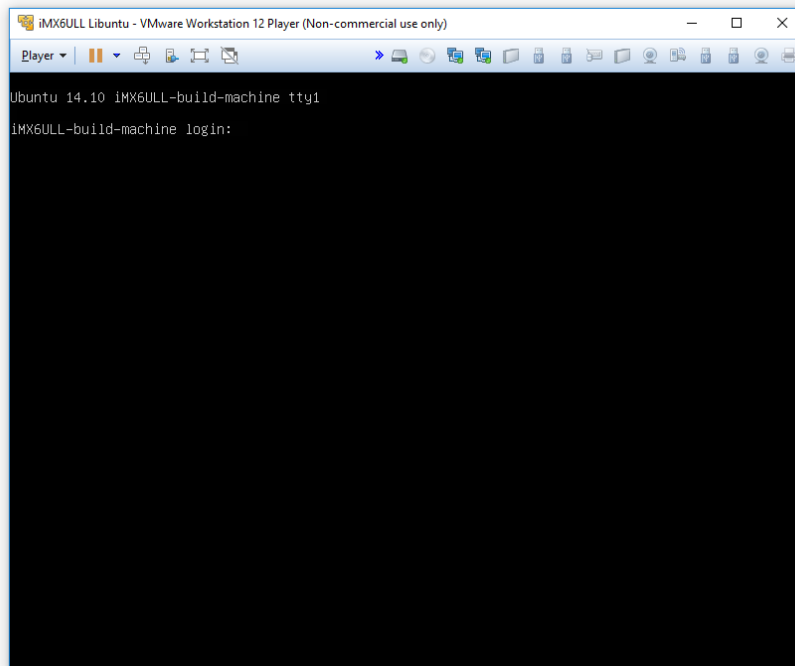


Разрешение экрана можно изменить в разделе **Perfences->Monitor Settings**

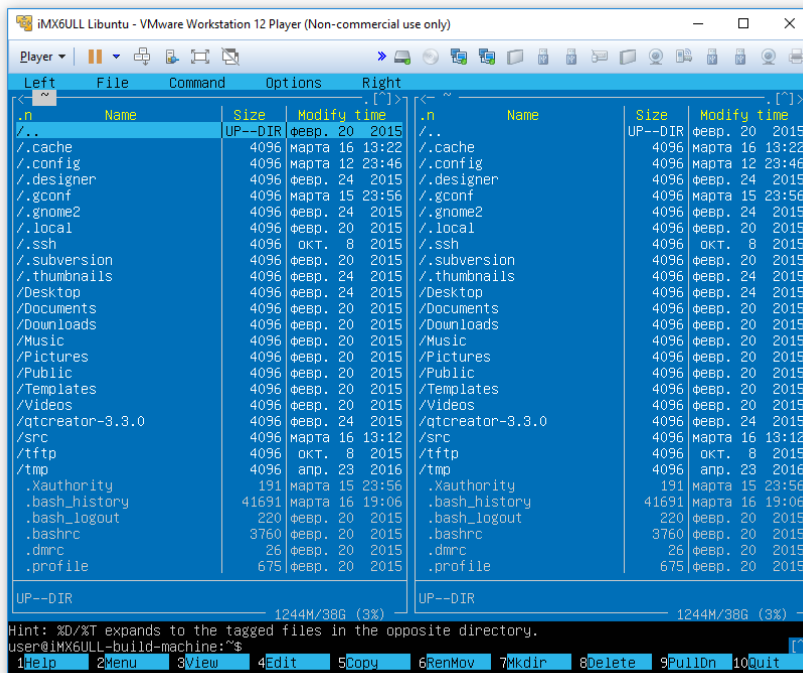
В системе присутствует один пользователь, **логин: user, пароль: 123456**
Суперпользователя в системе нет, для запуска приложений с его привилегиями необходимо использовать **sudo** (пароль 123456).

SK-iMX6ULL-NANO

Работать с файлами и текстами не всегда удобно через графический рабочий стол, для переключения в консольный режим необходимо нажать Ctrl+Alt+F(1-6) (Ctrl+Alt+F7 – переключение на графический рабочий стол).

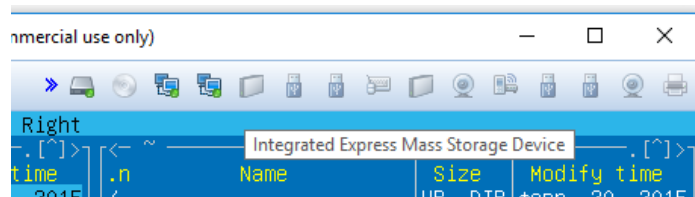


mc – Midnight Commander, файловый менеджер



SK-iMX6ULL-NANO

Через раздел **Player>Removable devices** можно подключать-отключать к виртуальной машине различные системные устройства, например, USB устройства, картридеры и т.п. Эта же функция дублируется через графическую панель:



Функция очень полезна, потому как позволяет подключить картридер непосредственно в виртуальную машину и напрямую оперировать с картами памяти.

Нет необходимости выключать виртуальную машину после завершения работы, можно перевести машину в режим паузы, а в следующий раз продолжить работу с момента паузы. **Внимание!** В режиме паузы останавливается системное время, что может негативно сказаться на сборке вновь скачиваемых архивов, во избежание этого следует подстраивать системное время, либо проводить перезагрузку виртуальной машины.

При правильной настройке сетевых интерфейсов, виртуальная машина должна иметь доступ в Internet, PC должен иметь успешный ping по адресу 192.168.0.2 (адрес Bridget сетевого адаптера в виртуальной машине) и при подключенной плате должен быть успешным ping адреса 192.168.0.136.

BUILDROOT

Buildroot располагается в папке /home/user/src/buildroot-2017.08-sk-ull

Перед началом сборки необходимо сконфигурировать Buildroot, имеются следующие варианты сборки:

1. `imx6ullsk_nano_defconfig` – основной вариант сборки

```
> cd /home/user/src/buildroot-2017.08-sk-ull
> make clean
> make imx6ullsk_nano_defconfig
> make menuconfig
```

Выбор изделия (модуль/плата/тип материнской платы) iMX6ULL под которое будет осуществляться сборка осуществляется в меню «Bootloaders» и в селекторе «Starterkit i.MX6ULL board».

Основные команды:

- **make** – сборка системы
- **make menuconfig** – запуск меню настроек и состава требуемых пакетов
- **clean** – очистка системы, **ВНИМАНИЕ!!!** Полностью удаляется содержимое папки `output`, что удалит все изменения в исходных кодах и настройки, перед чисткой нужно позаботиться об этом.
- **make linux-menuconfig** – запуск конфигуратора ядра Linux
- **make linux-rebuild** – принудительная сборка ядра Linux
- **make busybox-menuconfig** – запуск конфигуратора Busybox
- **make busybox-rebuild** – принудительная сборка Busybox
- **make uboot-rebuild** – принудительная сборка загрузчика U-boot

Длительность процесса сборки зависит от производительности вашего PC, может занять несколько часов. При повторных запусках будут собираться только вновь добавленные

SK-iMX6ULL-NANO

пакеты, что не требует много времени. **Важно!** При исключении пакета из сборки он не удаляется, остается в сборке КФС до чистки.

В результате сборки в папке **output** появится несколько новых папок:

- **build** – содержит рабочие папки собираемых пакетов, а также ядро и загрузчик
- **target** – результат сборки, скопировав сюда файл, он появится в образах КФС после выполнения `make`
- **images** – ядро, загрузчик, архив КФС ...

Для обновления КФС или ядра Linux на модуле SK-iMX6ULL-NANO, необходимо скопировать файл `output/images/zImage.imx6ull-sk-nano` в `mfgtools\Profiles\Linux\OS Firmware\files\` и провести процедуру описанную в разделе «Программирование QSPI flash».

BUSYBOX

Большинство системных утилит реализованы не отдельными программами, а специальным многофункциональным средством Busybox, в папке `/bin` находятся не программы, а ссылки на Busybox с указанием требуемого вызова.

SK-iMX6ULL-NANO

ТРЕБОВАНИЯ И РЕКОМЕНДАЦИИ ПО ИНТЕГРАЦИИ МОДУЛЯ SK-IMX6ULL-NANO

- 1) Хочу обратить особое внимание на особенность включения питающих напряжений процессора iMX6ULL (впрочем, это условие есть практически у всех семейств процессоров iMX):

4.2.3 Power Supplies Usage

All I/O pins should not be externally driven while the I/O power supply for the pin (NVCC_xxx) is OFF. This can cause internal latch-up and malfunctions due to reverse current flows.

Иными словами – если при включении питания процессора на любом из GPIO будет присутствовать какое-либо напряжение, внутри процессора может что то «защелкнуться» и привести к непредсказуемым последствиям (в лучшем случае он зависнет).

Из опыта эксплуатации iMX6S/D известно, что такое «защелкивание» может приводить к короткому замыканию внутри процессора и если источник питания обладает достаточной мощностью – к выведению микросхемы процессора из строя.

Эта особенность носит статистический характер - не факт, что его последствия проявятся на 10 экземплярах, но на партиях в сотни изделий скорее всего начнут появляться модули выходящие из строя без видимых причин при включении питания.

Так же хочу обратить внимание на зависимость данного «эффекта» от технологических аспектов производства самих процессоров, т.е. бывали случаи: в одном году интегрировали без проблем 100 модулей, а через год следующие 100 модулей через один переходили в короткое замыкание.

Во избежание последствий описываемого эффекта, требуется тщательно проанализировать все GPIO модуля на предмет возможного возникновения потенциала перед включением питания, на этапе прототипирования перепроверить отсутствие каких-либо потенциалов на всех подключаемых GPIO (для выявления случаев «паразитной» запитки). Если гарантировано невозможно обеспечить нулевой потенциал на линии, обязательно поставить на этих линиях буфер, а на этапе прототипирования проверить отсутствие «паразитной» запитки.

SK-iMX6ULL-NANO

ТАБЛИЦА НАЗНАЧЕНИЯ КОНТАКТОВ SK-IMX6ULL-NANO

Большинство портов ввода-вывода процессора iMX6ULL имеет до семи периферийных функций, таблица соответствия выводов процессора и разъемов модуля SK-iMX6ULL-NANO:

SK-iMX6ULL-NANO	CPU pin - NAME	SK-iMX6ULL-NANO	CPU pin - NAME
X1.1 - GND	GND	X1.2 - 5V	5В, питание модуля
X1.3 - USB_OTG1_DN	T15 - USB_OTG1_DN	X1.4 - 5V	5В, питание модуля
X1.5 - USB_OTG1_DP	U15 - USB_OTG1_DP	X1.6 - 5V	5В, питание модуля
X1.7 - GPIO/USBPWR	H14 - UART2_RTS	X1.8 - 5V	5В, питание модуля
X1.9 - USB_OTG2_DN	T13 - USB_OTG2_DN	X1.10 - Battery	P12 - VDD_SNVS_IN
X1.11 - USB_OTG2_DP	U13 - USB_OTG2_DP	X1.12 - RESET	P8 - POR
X1.13 - GPIO/USBPWR	J15 - UART2_CTS	X1.14 - GND	GND
X1.15 - GND	GND	X1.16 - SPI_CLK	A16 - ENET2_TX_D1
X1.17 - CAN_RX	G14 - UART3_RTS	X1.18 - SPI_MOSI	B15 - ENET2_TX_EN
X1.19 - CAN_TX	H15 - UART3_CTS	X1.20 - SPI_MISO	D17 - ENET2_TX_CLK
X1.21 - UART_TX_CON	K14 - UART1_TX	X1.22 - SPI_CS0	D16 - ENET2_RX_ER
X1.23 - UART_RX_CON	K16 - UART1_RX	X1.24 - SCI_CS1	B17 - ENET2_RX_EN
X1.25 - I2C1_SDA	L17 - GPIO_3	X1.26 - I2C2_SCL	C17 - ENET2_RX_D0
X1.27 - I2C1_SCL	L14 - GPIO_2	X1.28 - I2C2_SDA	C16 - ENET2_RX_D1
X1.29 - 3.3V	3.3V, выход	X1.30 - UART1_TX	J17 - UART2_TX
X1.31 - SD_D1	B2 - SD1_D1	X1.32 - UART1_RX	J16 - UART2_RX
X1.33 - SD_D0	B3 - SD1_D0	X1.34 - UART3_TX	H17 - UART3_TX
X1.35 - SD_CLK	C1 - SD1_CLK	X1.36 - UART3_RX	H16 - UART3_RX
X1.37 - SD_CMD	C2 - SD1_CMD	X1.38 - GPIO	G17 - UART4_TX
X1.39 - SD_D3	A2 - SD1_D3	X1.40 - GPIO	G16 - UART4_RX
X1.41 - SD_D2	B1 - SD1_D2	X1.42 - GPIO	F17 - UART5_TX
X1.43 - 3.3V	3.3V, выход	X1.44 - GPIO	G13 - UART5_RX
X1.45 - ETH_TXP	Ethernet PHY TX "+"	X1.46 - GPIO/ADC	L15 - GPIO1_1
X1.47 - ETH_TXN	Ethernet PHY TX "-"	X1.48 - DAC	N16 - JTAG_TDI
X1.49 - ETH_RXP	Ethernet PHY RX "+"	X1.50 - ADC	K13 - GPIO1_0
X1.51 - ETH_RXN	Ethernet PHY RX "-"	X1.52 - GND	GND

SK-iMX6ULL-NANO

СВЕДЕНИЯ ОБ ОРГАНИЗАЦИИ

www.starterkit.ru

info@starterkit.ru

Skype: starterkit.ru

Россия, г.Ижевск, ул.Телегина д.30

Тел.: (+7 3412) 478-448, +79226802173, +79226802174

www.otladka.com.ua

evodbg@gmail.com

Украина, г. Киев, ул. Желябова 8/4 офис 20

тел: 38-044-362-25-02

тел: 38-091-910-68-18

тел: 38-098-661-97-97

тел: 38-063-750-57-75