

Wi-Fi

Contents

Режимы работы

Первое подключение по Wi-Fi

- Антенны
- Подключение к точке доступа

Настройка Wi-Fi на контроллере Wiren Board

- Настройка в режиме точки доступа
- Установка пароля на подключение к точке доступа
- Настройка в режиме точки доступа и клиента одновременно
- Отключение режима точки доступа
- Настройка в режиме клиента
- Подключение к Wi-Fi точке доступа вручную
- Универсальный файл настроек Wi-Fi
- Автоматическое переключение при проблемах с соединением

Режимы работы

Wi-Fi в Wiren Board можно настроить на работу в одном из двух или трёх режимов:

- Режим точки доступа (включён по умолчанию). Работает относительно медленно. Скорости вполне хватит для работы с веб-интерфейсом, но не стоит использовать как замену роутера.
- Режим клиента.
- Одновременная работа в режиме и точки доступа, и клиента.

В очень редких случаях возможна несовместимость адаптера Wi-Fi в Wiren Board с некоторыми другими устройствами Wi-Fi. Это общая проблема реализаций Wi-Fi на чипсетах разных производителей. Если вы столкнулись с необъяснимыми проблемами при работе, рекомендуем поменять настройки шифрования, ширины канала и т.п.

Первое подключение по Wi-Fi

Антенны

Прикрутите антенну к разъёму для антенны Wi-Fi.

Без антенны Wi-Fi в контроллерах Wiren Board работает на расстоянии не более одного метра. Чтобы получить стандартный для Wi-Fi радиус работы, нужно подключить к соответствующему разъёму контроллера антенну. Если контроллер находится в щитке (особенно в металлическом) или отдельной комнате, лучше использовать выносную антенну. Разъём для антенны — стандартный для Wi-Fi RP-SMA (https://en.wikipedia.org/wiki/SMA_connector#Reverse_polarity_SMA) ("гнездо", у GSM-антенн - наоборот).



Сравнение разъёмов для антенн Wi-Fi (RP-SMA) и GSM (SMA)

Подключение к точке доступа

Контроллер создает Wi-Fi точку доступа и мы можем подключиться к ней:

- Откройте на ноутбуке или телефоне список WiFi точек доступа.
- Выберите из списка точку доступа с именем WirenBoard-XXXXXXX. Где XXXXXXX - серийный номер контроллера.

При подключении по Wi-Fi контроллер будет доступен по IP-адресу **192.168.42.1**.

По умолчанию, для подключения к контроллеру по Wi-Fi не требуется пароль, но вы можете это изменить.

Настройка Wi-Fi на контроллере Wiren Board

Настройка производится стандартным для Linux Debian способом - через файл `/etc/network/interfaces`. Краткие инструкции для типовых задач даны ниже, на сайте Linux Debian есть подробная документация (<https://wiki.debian.org/ru/NetworkConfiguration>).

Настройка в режиме точки доступа

Режим точки доступа включён по умолчанию. Работа в режиме точки доступа обеспечивается демоном **hostapd** (<https://wireless.wiki.kernel.org/en/users/documentation/hostapd>).

Сперва настраиваем демон hostapd:

1. в файле `/etc/default/hostapd` раскомментируйте строку (то есть удалите знак `#` в начале строки)

```
DAEMON_CONF="/etc/hostapd.conf"
```

2. отредактируйте файл `/etc/hostapd.conf`, чтобы он выглядел так:

```
interface=wlan0
#driver=nl80211 # оставьте эту строку закомментированной
ssid=WirenBoard # вместо WirenBoard можете подставить другое имя для создаваемой точки доступа
channel=1
```

Теперь нужно настроить сам интерфейс. Настройка делается в файле `/etc/network/interfaces`:

1. раскомментируйте и отредактируйте (или добавьте, если их не было) строки, относящиеся к настройке в режиме точки доступа:

```
iface wlan0 inet static
    address 192.168.42.1 # здесь 192.168.42.1 - адрес, по которому в новой сети будет находиться Wiren Board; можете указать другой адрес
    netmask 255.255.255.0
```

2. закомментируйте строки, относящиеся к работе в режиме клиента:

```
#auto wlan0
#iface wlan0 inet dhcp
#    wpa-ssid {ssid}
#    wpa-psk {password}
```

Выполните команду

```
/etc/init.d/hostapd restart
```

В итоге у нас получилась открытая точка доступа, для подключения к которой не требуется пароль.

Установка пароля на подключение к точке доступа

Подключитесь к контроллеру по SSH и откройте файл настроек `/etc/hostapd.conf`, для этого введите команду:

```
nano /etc/hostapd.conf
```

Добавьте в конец файла строки:

```
wpa=2
wpa_passphrase=your_password
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP CCMP
rsn_pairwise=TKIP CCMP
```

Придумайте свой пароль и замените в файле `your_password` на него. Сохраните файл нажатием клавиш `Ctrl+O` и выйдите из редактора `Ctrl+X`.

После этого выполните команду:

```
/etc/init.d/hostapd restart
```

Контроллер применит новые настройки и связь с ним будет потеряна. Нужно будет заново подключиться к контроллеру по WiFi с указанным паролем. Если изменения настроек вы не можете подключиться к контроллеру по WiFi — подключитесь к нему по Ethernet и проверьте настройки в файле `/etc/hostapd.conf`.

Настройка в режиме точки доступа и клиента одновременно

Режим одновременной работы модуля Wi-Fi и в режиме точки доступа, и в режиме клиента, называется *Concurrent Mode* или *STA+SoftAP*, и поддерживается не всеми Wi-Fi модулями. Он работает на всех версиях Wiren Board 6 и на некоторых ревизиях WB5. Проверено, что он работает из коробки на Wiren Board с чипом Realtek 8723BU и ядром Linux 4.1.15. Чтобы проверить, выполняются ли эти условия, выполните команды:

```
uname -a
lsmod | grep 8723bu
```

Если условия не выполнены, возможно, на вашем Wiren Board, всё равно, можно настроить *Concurrent Mode*. В качестве отправной точки используйте инструкцию (<http://randomstuffedosometimes.blogspot.ru/2016/03/rtl8192cu-and-rtl8188cus-in-station-and.html>).

Если условия выполнены:

1. Выполните команду

```
iwconfig
```

В её выводе должны быть показаны два интерфейса Wi-Fi: *wlan0* и *wlan1*.

2. Настройте по двум предыдущим инструкциям подключение в режиме клиента и подключение в режиме точки доступа, но используйте для них разные интерфейсы. Например, оставьте *wlan0* для точки доступа, а клиента сделайте на *wlan1*. Соответствующая часть файла */etc/network/interfaces* должна выглядеть так:

```
# Wireless interfaces
auto wlan1
iface wlan1 inet dhcp
    wpa-ssid {ssid} # вместо {ssid} подставьте имя точки доступа
    wpa-psk {password} # вместо {password} подставьте пароль

auto wlan0
iface wlan0 inet static
    address 192.168.42.1
    netmask 255.255.255.0
```

Отключение режима точки доступа

Если вы хотите перевести адаптер в режим клиента, подключиться к Wi-Fi точке доступа в ручном режиме или совсем отключить Wi-Fi на контроллере — отключите режим точки доступа:

1. Отключите автоматический запуск сервиса *hostapd*:

```
systemctl disable hostapd
```

2. Остановите демон *hostapd*

```
service hostapd stop
```

3. Теперь прокомментируйте настройки точки доступа и задайте настройки WiFi-клиента:

- откройте файл для редактирования

```
mcedit /etc/network/interfaces
```

- прокомментируйте строки, относящиеся к настройке в режиме точки доступа:

```
#allow-hotplug wlan0
#iface wlan0 inet static
# address 192.168.42.1
# netmask 255.255.255.0
```

4. Сохраните и закройте файл настроек.

5. Запретите раздачу IP-адресов, для этого остановите DHCP-сервер:

```
systemctl disable dnsmasq
service dnsmasq stop
```

Режим точки доступа отключен, чтобы его включить, выполните инструкции из раздела Настройка в режиме точки доступа.

Настройка в режиме клиента

После настройки точки доступа в режиме клиента, контроллер будет подключаться к точке доступа автоматически при каждой загрузке операционной системы.

Вы можете настроить автоматическое подключение контроллера к Wi-Fi точке доступа:

1. Отключите точку доступа по инструкции в разделе Отключение режима точки доступа
2. Откройте файл настроек:

```
mcedit /etc/network/interfaces
```

3. Раскомментируйте и отредактируйте строки (или добавьте, если их не было):

```
auto wlan0
iface wlan0 inet dhcp
    wpa-ssid ssid # вместо ssid подставьте имя точки доступа
    wpa-psk password # вместо password подставьте пароль
```

4. Если точка доступа скрыта, то добавьте параметр:

```
wpa-scan-ssid 1
```

5. Сохраните и закройте файл настроек.

6. Завершите настройку, для этого перезапустите беспроводной интерфейс командами:

```
ifdown wlan0 && ifup wlan0
```

Подключение к Wi-Fi точке доступа вручную

Подключение в ручном режиме будет разорвано после перезагрузки контроллера.

Если у вас возникла проблема с настройкой автоматического подключения, то вы можете попробовать подключиться к Wi-Fi точке доступа вручную:

1. Отключите точку доступа по инструкции в разделе Отключение режима точки доступа
2. Запустите поиск доступных точек доступа с помощью команды `iwlist wlan0 scanning`:

```
~# iwlist wlan0 scanning | grep -i essid
ESSID:"DIR-615"
ESSID:"MTSRouter_2.4GHz_072433"
ESSID:"Smart_box-40B598"
ESSID:"TP-Link_0E5AW"
ESSID:"TP-LINK_78DC"
```

в примере контроллер «видит» пять точек доступа.

3. Этот шаг зависит от типа сетевой аутентификации, выбранной в настройках точки доступа, к которой вы хотите подключиться:

- WPA-PSK:

1. Задайте параметры подключения:

```
iwconfig wlan0 essid ИмяТочкиДоступа key ПарольОтТочкиДоступа
```

2. Запустите сетевой интерфейс:

```
ifconfig wlan0 up
```

- WPA2-PSK:

1. Сгенерируйте файл с учётной записью для подключения к точке доступа:

```
wpa_passphrase ИмяТочкиДоступа ПарольОтТочкиДоступа > /root/wpa.conf
```

2. Установите подключение с использованием сгенерированного файла:

```
wpa_supplicant -Dwext -iwlan0 -c/root/wpa.conf &
```

4. Подождите 15 секунд и проверьте подключение командой `iwconfig wlan0`:

```
~# iwconfig wlan0 | grep -i essid
wlan0 IEEE 802.11bgn ESSID:"DIR-615" Nickname:"<WIFI@REALTEK>"
```

в примере контроллер подключён к точке доступа с именем DIR-615. Если в строке будет `unassociated`, то контроллер не смог подключиться.

5. Если контроллер успешно подключился к точке доступа и на ней запущен DHCP-сервер, то запустите `dhcpcd`:

```
dhcpcd wlan0
```

6. Проверьте, получил ли контроллер IP адрес, для этого используйте команду `ip a`:

```
~# ip a | grep wlan0
5: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    inet 192.168.2.83/24 brd 192.168.2.255 scope global wlan0
```

в примере контроллер получил ip-адрес 192.168.2.83.

Настройка подключения контроллера к точке доступа завершена.

Универсальный файл настроек Wi-Fi

Ниже приведен текст файла с настройками для подключения к сетям с разными параметрами шифрования. Оригинал файла можно посмотреть на сайте www.raspberrypi.org (<https://www.raspberrypi.org/forums/viewtopic.php?t=7592>).

```
#####
#; start of wireless bits
#; this command stays for all configs
auto wlan0
#####
#; comments indicated by #;
#; commands indicated by #
#; remove the # to enable the command
#####
#; if using static IP then...#
#iface wlan0 inet static
# address UR_IP
#gateway UR_ROUTER_IP
#netmask 255.255.255.0
#####
#; otherwise use dhcp #
#iface wlan0 inet dhcp
#####
#; OPEN wireless config #
#wireless-essid UR_ESSID
#wireless-mode managed

#####
#; WEP wireless config #
#wireless-essid UR_ESSID
#wireless-key UR_KEY
#; end of WEP config

#####
#; WPA and WPA2 wireless config #
# all command config lines above HERE to be #'ed except the entry auto wlan0
#####
wpa-driver wext
wpa-ssid UR_ESSID
#; wpa-ap-scan is 1 for visible and 2 for hidden hubs
wpa-ap-scan 1
#; wpa-proto is WPA for WPA1 (aka WPA) or RSN for WPA2
wpa-proto WPA
#; wpa-pairwise and wpa-group is TKIP for WPA1 or CCMP for WPA2
wpa-pairwise TKIP
wpa-group TKIP
wpa-key-mgmt WPA-PSK
#; use "wpa_passphrase UR_ESSID UR_KEY" to generate UR_HEX_KEY
#; enter the result below
wpa-psk UR_HEX_KEY
#####
# end of wireless bits
```

Автоматическое переподключение при проблемах с соединением

Способ заимствован на сайте alexba.in (<http://alexba.in/blog/2015/01/14/automatically-reconnecting-wifi-on-a-raspberrypi/>).

Допустим, контроллер подключён к роутеру с адресом 192.168.0.1 через интерфейс wlan1:

1. Создайте в папке /root скрипт `wifi_autoconnect.sh`:

```
ncedit /root/wifi_autoconnect.sh
```

с содержанием:

```
#!/bin/bash

# Подставьте имя интерфейса
WLANINTERFACE=wlan1
# Подставьте адрес роутера или сервера в интернете, доступ к которому будет проверяться
SERVER=192.168.0.1

PATH="/bin:/sbin:/usr/local/sbin:/usr/sbin:$PATH"
# Only send two pings, sending output to /dev/null
ping -I ${WLANINTERFACE} -c2 ${SERVER} > /dev/null

# If the return code from ping ($?) is not 0 (meaning there was an error)
if [ $? != 0 ]
then
# Restart the wireless interface
ifdown -f ${WLANINTERFACE}
ifup ${WLANINTERFACE}
fi
```

2. Сделайте файл исполняемым, выполнив команду

```
chmod +x /root/wifi_autoconnect.sh
```

3. Запланируйте выполнение скрипта каждую минуту:

Добавьте в конец файла /etc/crontab строку

```
* * * * * root /root/wifi_autoconnect.sh
# Обязательно добавьте пустую строку в конец файла
```

SIM68V

- English
- русский

Contents

Краткие характеристики

В Wiren Board

Управление

- Отключение модуля
- AlwaysLocate

AGPS

Документация

Краткие характеристики

- Чипсет MT3333, MediaTek
 - Количество каналов: 33 для слежения, 99 для захвата
 - Навигационные системы: ГЛОНАСС/GPS/GALILEO/GZSS
 - Точность : 2.5м (СЕР 50% 24часа)
 - Чувствительность:
 - Tracking: - 167dBm
 - Navigation: -157dBm
 - Cold Start: -148dBm
 - TTFF:
 - Холодный старт: 28 сек (типичное значение)
 - Горячий старт: 1 сек (типичное значение)
 - Работа с переотраженными сигналами
 - SBAS (WAAS, EGNOS, GAGAN, MSAS)
 - Функции:
 - Принудительный выбор режима работы GPS Only, GLONASS Only или GPS/GLONASS
 - AlwaysLocate (адаптивная работа приемника в зависимости от движения для экономии энергопотребления)
 - EPO/Hot Still (применение скачанных с сервера аппроксимированных эфемерид на 7/14/30 дней)
 - EASY (самостоятельная аппроксимация эфемерид на 3 дня вперед)
 - LOCUS (хранение треков в памяти)
 - AIC (подавление шумовых гармоник с информированием об их наличии - Jamming)
 - Протокол: NMEA 0183/MTK
 - Частота вывода сообщений до 10Гц (1Гц по умолчанию)
 - Скорость обмена UART: 4800...921600 бит/сек (115200 или 9600 бит/сек по умолчанию)
 - Тип антенн: активная/пассивная
 - Питание: 2.8...4.3В, 27мА в режиме ГЛОНАСС+GPS при непрерывном слежении
 - Рабочий диапазон температур -40°C...+85°C

В Wiren Board

Подключён к чипу SC16IS752. Порт в Linux `/dev/ttyNSC1`. Скорость по-умолчанию 9600. (Некоторые модули могут иметь скорость 115200). Проктокол вывода - NMEA.

Конфигурация порта:

```
stty -F /dev/ttyNSC1 line 0 ignbrk -brkint -icrnl -imaxbel -opost -onlcr -isig -icanon -iexten -echo -echoe -echok -echoctl -echoke 9600
```

Терминал (проверка работы):

```
minicom /dev/ttyNSC1
```

Управление

Модулем GPS можно управлять с помощью отправки в порт проприетарных команд чипсета MT3333. Список команд см. в разделе документация.

Примеры полезных команд приведены ниже.

Отключение модуля

Команда **161 PMTK_CMD_STANDBY_MODE**. Переводит модуль в неактивный режим для снижения энергопотребления. Для пробуждения модуля надо послать любой байт в порт.

```
echo -e "\$PMTK161,0*28\r" > /dev/ttyNSC1
```

В Wiren Board rev. 2.8 питание внешней активной антенны не отключается при переводе модуля в режим энергосбережения. Для отключения питания внешней активной антенны необходимо отключить всю шину питания 3.3В .

AlwaysLocate

Режим AlwaysLocate - адаптивная работа приемника в зависимости от движения для экономии энергопотребления. В этом режиме приёмник периодически засыпает и просыпается, при этом периодичность и продолжительность режимов определяется автоматически.

225 PMTK_CMD_PERIODIC_MODE

Вернуться в обычный режим работы:

```
echo -e "\$PMTK225,0*2B\r" > /dev/ttyNSC1
```

Режим AlwaysLocate (TM) standby mode:

```
echo -e "\$PMTK225,8*23\r" > /dev/ttyNSC1
```

Режим AlwaysLocate (TM) backup mode:

```
echo -e "\$PMTK225,9*22\r" > /dev/ttyNSC1
```

AGPS

Чипсет поддерживает загрузку скачанных с сервера эфемерид для ускорения старта. Загруженные в модуль эфемериды сохраняются во внутренней флеш-памяти модуля.

Подробнее: <http://electronix.ru/forum/index.php?s=&showtopic=114152&view=findpost&p=1177669>

Загрузка происходит с помощью простого бинарного протокола. Готовой реализации для Linux пока нет.

Документация

- Описание
- Документация
- Проприетарные команды MT3333 от SIMCOM: Файл:SIM28 SIM68R SIM68V NMEA Messages Specification V1.01.pdf
- Описания других модулей на MT333x: http://ultran.ru/sites/default/files/it530m_datasheet_0.pdf, http://www.wless.ru/files/GPS/Locosys/MC-1010-G_datasheet_v1.0.pdf
- ftp://ftp.macrogroupp.ru/Support/Seminar/webinar_new%20modules_%2001.01.2013%20.pdf

wirenboard

Wiren Board 2.8

wirenboard

Wiren Board 2.8

Руководство по эксплуатации

Самая актуальная документация всегда доступна на нашем сайте по ссылке: https://wirenboard.com/wiki/Wiren_Board_rev_2.8

Этот документ составлен автоматически из основной страницы документации
и ссылок первого уровня.

Содержание

[Wiren Board 2.8](#)

[Первое включение Wiren Board](#)

[Debug UART](#)

[Создание microSD-карты с образом](#)

[Стандартный образ ФС](#)

[Работа с GPIO](#)

[RS-485](#)

[SC16IS752](#)

[Wi-Fi](#)

[SIM68V](#)

[1-Wire в Wiren Board 2.8](#)

[Device Tree](#)

[Сборка ядра Linux](#)

[Hardware Overview rev. 2.8-rus](#)

[Подключение периферийных устройств к контроллеру Wiren Board](#)

[UEXT-разъемы](#)

[Питание USB-портов](#)

[Wiren Board 2.8. Исходные файлы](#)

[Периферийные устройства](#)

Wiren Board 2.8

[Первое включение Wiren Board](#)



Wiren Board rev. 2.8

Software

[Debug UART](#)

[Создание microSD-карты с образом](#)

[Стандартный образ ФС](#)

[Работа с GPIO](#)

Работа с GPIO

Подсистемы:

- [ADC#Software](#)
- [Управление низковольтной нагрузкой](#), см. [Работа с GPIO](#), [Управление низковольтной нагрузкой#Пример работы в Linux](#)
- [GSM/GPRS](#)
- [RS-485 через SC16IS752](#)
- [Wi-Fi](#)
- [NFC#Software](#)
- [Работа с GPS: SIM68V и SC16IS752](#)
- [1-Wire в Wiren Board 2.8](#)

Установка ПО

Кросс-компиляция

Пересборка Device Tree

Сборка ядра

Примеры от Olimex: <https://github.com/OLIMEX/OLINUXINO/tree/master/SOFTWARE/iMX233>

Hardware

Wiren Board 2.8 Описание платы

Управление низковольтной нагрузкой

UEXT-разъемы

Питание USB-портов

Wiren Board 2.8. Исходные файлы

Работа с внешними устройствами

Периферийные устройства

Первое включение Wiren Board

- [English](#)
- русский

Вы смотрите статью по старой версии контроллера Wiren Board. [Перейти к статье о Wiren Board 6.7](#)

Contents

[Подключение антенн](#)

[Карта памяти](#)

[Питание](#)

[Debug UART](#)

Подключение антенн

Разъем u.fl для вайфая находится внутри Wiren Board.

Для подключения антенны необходимо снять красную плату, и легким нажатием до щелчка одеть ответный разъем (удобно это сделать карандашом).

Антенны для GPS и GSM подключаются с внешней стороны платы.

Не располагайте антенну GSM рядом с процессором. Наводки на кварц процессора вызывают его перезагрузку. Поверните антенну в сторону от разъемов USB и Ethernet (как на фото).

Карта памяти

[Создайте microSD-карту с образом.](#)

Вставьте в разъем (на боковой стороне платы OLinuXino-MICRO).

Питание

Подключите внешний блок питания в power jack. Или подайте питание на клеммники 17 (Vin) и 18 (GND). Допустимый диапазон напряжений - 5-22 В.

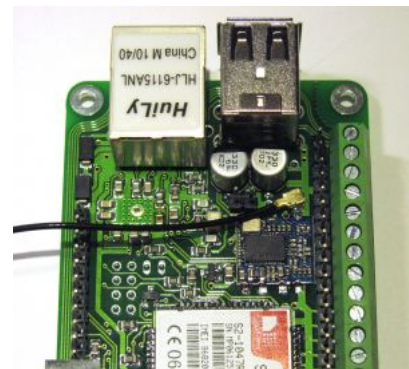
Debug UART

Про Debug UART прочитайте [здесь](#).

Если у вас Windows, скачайте [Putty](#) и запустите ее.

В настройках укажите тип соединения - serial, скорость - 115200, линия - COM3 (обычно это COM3, иначе смотрите в диспетчере устройств список портов).

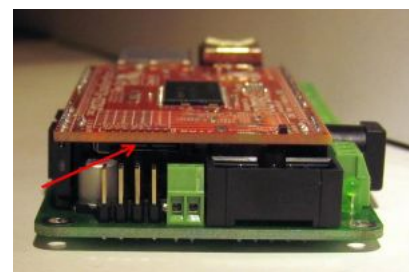
Подключите переходник USB-UART. Откройте сессию.



Подключение антенны Wi-Fi



Подключение антенн GPS и GSM



Разъем для карты памяти

Создание microSD-карты с образом

- [English](#)
- [русский](#)

Contents

Запись готового образа на карточку

[Выбор нужного образа](#)

[Для Windows](#)

[Для Linux](#)

Создание образа по частям

[Сборка вместе](#)

[Таблица разделов](#)

[Загрузчик](#)

[Создание фс](#)

[Копирование образа на раздел](#)

Пример

Запись готового образа на карточку

Выбор нужного образа

- Зайдите на [страницу готовых образов в Github](#) и выберите нужный образ:
 - для Wiren Board 4 - название оканчивается на `_wb4`
 - для Wiren Board Smart Home 3.5 - название оканчивается на `_wb3`
 - если серийный номер вашего Wiren Board Smart Home 3.5 больше 300 - используйте образ с `newwifi` в названии
 - для WB rev. 2.8 - название оканчивается на `_wb28`

У образа будет расширение `.dd`, либо `.dd.gz`, либо `img.zip`

- Распакуйте архив
- Следуйте инструкции для вашей операционной системы

Для Windows

- скачайте программу для записи образов(например, [Win32DiskImager](#))
- вставьте microSD-карту в ридер
- узнайте букву, под которой она появилась (например "F:")
- проигнорируйте сообщения о необходимости отформатировать диск перед использованием, если такое появится
- убедитесь, что другие программы не используют флеш-карту
- в Win32DiskImager выберите распакованный образ карты, выберите букву диска и нажмите кнопку *Write*

Для Linux

- вставьте microSD-карту в 리더
- узнайте название устройства, соответствующего карте. Обычно это /dev/mmcblk0 или /dev/sdX (где X - буква). В этом может помочь команда

```
dmesg | tail
```

Не перепутайте название устройства! Неправильно указав название устройства, вы навсегда потеряете все данные на вашем компьютере!

- отмонтируйте разделы карты, которые Linux примонтировал автоматически:
 - если устройство называется /dev/mmcblk0, то разделы называются /dev/mmcblk0p1, /dev/mmcblk0p2 и т.д.
 - если устройство называется /dev/sdb, то разделы называются /dev/sdb1, /dev/sdb2 и т.д.

Пример команды:

```
umount /dev/mmcblk0p1
```

- запишите образ на карту:

```
sudo dd if=sdcard.dd of=/dev/mmcblk0 bs=4M
```

, где "sdcard.dd" - путь к ранее скачанному распакованному файлу с образом.

Пример процесса целиком:

```
wget https://github.com/contactless/wireboard/releases/download/0.6-20140614/sdcard_20140614.img.zip
unzip sdcard_20140614.img.zip
umount /dev/mmcblk0p2
umount /dev/mmcblk0p1
sudo dd if=sdcard_20140614.img of=/dev/mmcblk0 bs=4M conv=fdatasync
sync
```

Создание образа по частям

Внимание! Это сложный вариант самостоятельной подготовки образа карточки. Лучше воспользуйтесь вариантом, описанным выше.

Внимание! На 6 ноября 2015 инструкция ниже ещё и безнадежно устарела.

Сборка вместе

Согласно [1]

- разбить флешку на два раздела
- записать u-boot на первый раздел
- создать фс на втором разделе
- скопировать gootfs на второй раздел

Сначала надо выяснить имя устройства с флеш-картой. Воспользуйтесь поиском. Можно, к примеру, попробовать запустить parted и посмотреть в нём. Устройство может быть /dev/sdb, а может выглядеть и как /dev/mmcblk0

Найдя устройство создаем переменную чтобы облегчить себе использование нижеследующих команд

```
MYDISK="/dev/sdb"
```

Между кавычками пишем название своего устройства с флеш-картой.

Таблица разделов

Перед запуском убедитесь, что разделы на карточке не примонтированы.

Теперь с помощью скрипта создадим разделы на карте. [2]

Скачиваем скрипт:

```
wget [https://raw.githubusercontent.com/contactless/wireboard/master/image/create_partitions.sh]
```

Запускаем скрипт и указываем ему имя устройства с флеш-картой:

```
sudo bash create_partitions.sh $MYDISK
```

Загрузчик

См. Сборка U-Boot

См. [Сборка образа](#).

Готовый образ u-boot для записи в раздел: [u-boot](#)

Скачиваем образ

```
wget "https://github.com/contactless/wirenboard/blob/master/contrib/u-boot/u-boot.sb.cl25?raw=true" -O u-boot.sb
```

Теперь выясняем имена разделов на карте. Скрипт выше отработал и разделил карту на разделы, нам надо узнать название первого из этих разделов. Если название устройства microSD-карты имело вид **/dev/sdX**, то первый раздел будет иметь название **/dev/sdX1**. Если устройство называлось **/dev/mmcblkX**, то первый раздел - **/dev/mmcblkXp1** (обратите внимание на **p** перед номером раздела).

Найдя имя первого раздела укажите его тут:

```
MYDISK1="/dev/sdb1"
```

Теперь заливаем образ раздела на карту

```
sudo dd if=u-boot.sb of=$MYDISK1 bs=512 seek=4
```

Создание фс

Находим название второго раздела на флешке и прописываем его в переменную

```
MYDISK2="/dev/sdb2"
```

rootfs станет названием этого раздела.

```
sudo mkfs.ext4 $MYDISK2 -L rootfs
```

Копирование образа на раздел

Готовый образ (включая ядро, dtbs, модули и прошивки): [releases](#)

См. также [Сборка образа](#)

```
wget https://github.com/contactless/wirenboard/releases/download/0.1/rootfs.tar.gz
```

Теперь надо примонтировать созданную файловую систему созданную нами ранее. *Как это сделать в терминале?*

По-умолчанию в Ubuntu она монтируется в **/media/\$USER/rootfs/**. Найдите куда система смонтировалась на вашем компьютере.

Распаковываем образ на раздел:

```
sudo tar xfpz rootfs.tar.gz -C /media/$USER/rootfs/
```

Отмонтируем файловую систему:

```
umount /media/user/rootfs
```

Пример

ОС Ubuntu, свежая SD-карта подключенная к встроенному считывателю и определяющаяся как **/dev/mmcblk0**. [Репозиторий](#) скачан, мы находимся в его корне.

Образ **rootfs.tar.gz** находится внутри папки **rootfs**.

```
cd image
sudo umount /dev/mmcblk0p1
sudo ./create_partitions.sh /dev/mmcblk0
sudo dd if=./contrib/u-boot/u-boot.sb of=/dev/mmcblk0p1 bs=512 seek=4
sudo ./create_fs.sh /dev/mmcblk0p2

# Ubuntu automount:
udisksctl mount -b /dev/mmcblk0p2

#extract rootfs
sudo tar xfpz ../rootfs/rootfs.tar.gz -C /media/$USER/rootfs/

umount /dev/mmcblk0p2
```

См. также [Стандартный образ ФС](#).

Стандартный образ ФС

- [English](#)
- [русский](#)

Contents

[Настройки по-умолчанию](#)

[Ethernet MAC](#)

[Wi-Fi](#)

[Скачать](#)

Настройки по-умолчанию

Логин: **root**, Пароль: **wirenboard**

Ethernet MAC

MAC-адрес Ethernet устанавливается программно. Адрес указан в файле **/etc/network/interfaces**

Wi-Fi

По-умолчанию создаётся точка доступа **Wiren Board**. Чтобы отключить это поведение, необходимо

- Изменить настройки интерфейса wlan0 беспроводной сети в **/etc/network/interfaces**
- Отключить hostapd - создание программной точки доступа с помощью файла **/etc/default/hostapd**

Скачать

<https://github.com/contactless/wirenboard/releases> См. [Создание microSD-карты с образом](#)

Работа с GPIO

ВНИМАНИЕ: статья рассчитана на разработчиков или опытных пользователей и даёт общие рекомендации того, как использовать gpio в обход официального ПО WirenBoard.

Если вам нужно работать напрямую с gpio, то мы рекомендуем делать это через драйвер `wb-mqtt-gpio` (<https://github.com/wirenboard/wb-homa-gpio>).

Описание доступных ножек gpio для конкретной ревизии контроллера можете посмотреть в статье GPIO.

Contents

Меры предосторожности

Именованние gpio

Вычисление номера gpio

Работа из userspace

Bash

Интерфейс sysfs

Чтение и запись

Работа с прерываниями

Работа через chardev

Python

Прямое обращение через память процессора

Работа в ядре Linux

Рекомендации по Device Tree

Пример device-tree node

Меры предосторожности

- Убедитесь, что вашу задачу нельзя решить стандартными средствами программного обеспечения Wiren Board.
- Все порты Wiren Board, в том числе и GPIO, работают с напряжением 3.3V.
- Подключение сигнала с напряжением большим 3.3V к ножке GPIO грозит выходом из строя процессорного модуля.

В случае необходимости подключения устройств, работающих с более высоким напряжением, необходимо использовать схемы согласования или подключать (для 5V) через резистор в 20 кОм и более.

Именованние gpio

К сожалению, четкого стандарта по именованию gpio не существует, но при работе с контроллерами WirenBoard стоит придерживаться следующих правил:

- выводы gpio сгруппированы по банкам (*banks*; эквивалентно *gpiochips*)
- каждый банк содержит 32 gpio. Нумерация банков начинается с 0.

Вычисление номера gpio

Для управления ножкой gpio нужно знать её номер. В рассматриваемых примерах будем работать с gpio *A1_IN* контроллера WB6.7 (номер: 109; *gpiochip 3, offset 13*): Вычислим банк gpio и offset, зная номер (109):

```
# Поделим 109 на 32. Целая часть – номер банка, остаток - offset:
109.0 / 32.0 = 3, остаток 13
```

То же самое справедливо и наоборот. Зная банк и offset (3 и 13, соответственно), можно вычислить номер gpio:

```
# Умножим номер банка на 32 и прибавим offset:
3 * 32 + 13 = 109
```

Работа из userspace

Перед началом работы из userspace, необходимо убедиться, в том, что нужный gpio — свободен. Для этого можно посмотреть на вывод команды

```
cat /sys/kernel/debug/gpio
```

В выводе команды видим примерно следующее:

```
gpiochip0: GPIOs 0-31, parent: platform/209c000.gpio, 209c000.gpio:
gpio-0 (          ) in hi IRQ
gpio-10 (          ) in lo
gpio-11 (          ) in hi
gpio-13 (          ) out lo
gpio-26 (          ) out lo
gpio-27 (          ) out hi
```

Это значит, что gpio 0, 26 и 27 уже экспортированы в sysfs и доступны для управления. Gpio 11 и 13 заняты ядерным драйвером *openwige* и недоступны для использования. Остальные gpio банка 0 — свободны.

Если нужный gpio — занят, то можно остановить драйвер:

```
lsmod | grep wl # узнаем название драйвера
rmmod wl_gpio # выгружаем драйвер, название которого узнали
```

ВНИМАНИЕ: остановка драйверов может привести к неожиданному поведению контроллера. Теперь нужный gpio свободен до следующей перезагрузки.

Bash

В настоящий момент, для работы с gpio в userspace доступны 2 интерфейса: *sysfs* и *chardev* (начиная с версии ядра 4.8).

Различия между *chardev* и *sysfs* хорошо описаны в этой статье (<https://embeddedbits.org/new-linux-kernel-gpio-user-space-interface/>). Sysfs имеет статус deprecated, поэтому, по возможности, стоит работать через chardev.

Интерфейс sysfs

Для работы через sysfs с определённым GPIO его надо экспортировать:

Здесь и далее N — номер gpio

```
echo N > /sys/class/gpio/export
```

Экспортированные gpio появляются в каталоге */sys/class/gpio*:

```
~# ls -l /sys/class/gpio/
export
gpio32
gpiochip0
gpiochip120
gpiochip32
gpiochip64
unexport
```

В директории */sys/class/gpioN* теперь находятся файлы для работы с GPIO (где N — номер GPIO, как и было сказано ранее):

```
~# ls -l /sys/class/gpio/gpioN/
active_low
device
direction
edge
power
subsystem
uevent
value
```

Установка направления GPIO (ввод/вывод) производится с помощью записи в файл `direction`

```
echo in > /sys/class/gpio/gpioN/direction # установим GPIO номер N на ввод
echo out > /sys/class/gpio/gpioN/direction # установим GPIO номер N на вывод
```

Чтение и установка значения GPIO производится с помощью файла `value`.

Чтение и запись

Чтение:

```
echo in > /sys/class/gpio/gpioN/direction # установим GPIO номер N на ввод
cat /sys/class/gpio/gpioN/value # вернёт 1 или 0
```

Запись:

```
echo out > /sys/class/gpio/gpioN/direction # установим GPIO номер N на вывод
echo 0 > /sys/class/gpio/gpioN/value # установим логический 0 (низкое напряжение) на GPIO номер N
echo 1 > /sys/class/gpio/gpioN/value # установим логический 1 (высокое напряжение) на GPIO номер N
```

Пример:

1. Находим номер GPIO, соответствующий вашей версии контролера нужному клеммнику в таблице WB2.8. Для клеммника номер 2 в версии 2.8 это GPIO 32.
2. Экспортируем GPIO в `sysfs`

```
echo 32 > /sys/class/gpio/export
```

3. Устанавливаем GPIO в режим вывода для управления транзистором. Это обязательно, т.к. GPIO может находиться в режиме ввода и иметь высокий импеданс, оставляя транзистор в неопределённом состоянии.

```
echo out > /sys/class/gpio/gpio32/direction
```

4. Открываем транзистор, подавая логический высокий уровень на затвор:

```
echo 1 > /sys/class/gpio/gpio32/value
```

5. Закрываем транзистор, подавая логический ноль на затвор:

```
echo 0 > /sys/class/gpio/gpio32/value
```

Работа с прерываниями

Через интерфейс `sysfs` можно запросить прерывания по изменению состояния процессора.

Установка прерывания производится путём записи значения в файл `"edge"`. Значения могут быть:

- `none` — отключить прерывание
- `rising` — включить прерывание по нисходящему фронту
- `falling` — включить прерывание по восходящему фронту
- `both` — включить прерывание по обоим фронтам.

Пример работы с прерываниями:

```
-# echo 3 > /sys/class/gpio/export # экспортируем GPIO номер 3 (TB10 на WB3.3)
-# cat /sys/class/gpio/gpio3/edge # проверяем состояние прерывания
none
-# echo falling > /sys/class/gpio/gpio3/edge # устанавливаем прерывание по нисходящему фронту
-# cat /proc/interrupts | grep gpiolib # прерывание появилось в списке. 26 - внутренний номер прерывания, 0 - количество событий
26:      0      gpio-mxs    3      gpiolib
-# cat /proc/interrupts | grep gpiolib # после нескольких событий, 76 - количество событий
26:      76      gpio-mxs    3      gpiolib
```

Прерывания можно ловить из `userspace` с помощью системного вызова `epoll()` и `select()` на файл `value`. Пример работы см. [1] (<https://github.com/contactless/wiegand-linux-sysfs>)

См. также [elinux.org](http://elinux.org/GPIO) (<http://elinux.org/GPIO>)

Работа через chardev

Представленный в ядре 4.8 интерфейс `chardev` имеет C/Python библиотеку `libgpiod` и `userspace`-утилиты для работы с `gpio`. Исходный код библиотеки и документация доступны в репозитории `libgpiod` (<https://github.com/brgl/libgpiod>).

Утилиты распространяются в составе debian-пакетов *gpiod* и *libgpiod-dev* для debian buster и новее. К сожалению, для **stretch пакетов в официальных репозиториях нет**.

Если нужно установить *libgpiod* в debian stretch, можно воспользоваться сторонними репозиториями (например, этим (<https://github.com/rcn-ee/repos>)). **Используйте сторонние репозитории на свой страх и риск; компания WirenBoard не контролирует их содержимое.**

Для работы с *gpio* из *bash* в пакете *gpiod* поставляются следующие утилиты:

- *gpiodetect* — информация обо всех банках *gpio* в системе
- *gpioinfo* — подробная информация обо всех линиях *gpio* определённого банка
- *gpioget* <чип> <линия> — возвращает значение определённого *gpio*
- *gioset* <чип> <линия1>=<значение1> <линия2>=<значение2> — устанавливает состояние на определенные линии *gpio*
- *gpiofind* <название> — возвращает номер *gpio*
- *gpiomon* — отслеживание событий *gpio*

Примеры использования *gpiod* можно посмотреть в [2] (<https://www.acmesystems.it/gpiod>) и [3] (<https://github.com/brgl/libgpiod>)

Python

Для управления *gpio* из *python* был написан модуль

```
wb_common.gpio
```

Модуль представляет собой обёртку вокруг *sysfs*. Исходный код доступен на нашем github. (https://github.com/wirenboard/wb-common/blob/master/wb_common/gpio.py)

Модуль позволяет работать с *gpio* в синхронном и асинхронном (с регистрацией коллбэков) режимах.

Прямое обращение через память процессора

Этот метод настоятельно НЕ РЕКОМЕНДУЕТСЯ для использования без достаточных оснований. Для работы из C/C++ стоит использовать работу через файлы в *sysfs* или *chardev*, как описано в предыдущих разделах.

Управлять GPIO можно с помощью прямого доступа к регистрам процессора, в обход Linux, через интерфейс */dev/mem*. При этом, по сравнению с работой через *sysfs* минимизируются накладные расходы. Этот метод можно использовать, если вам необходим очень быстрый доступ к GPIO, например *bitbang* протоколов или ШИМ. Стоит иметь в виду, что планировщик процессов всё ещё может вносить в работу программы значительные задержки. Рекомендуется выносить критичные ко времени задачи в ядро.

См. [4] (<http://olinux.wordpress.com/2012/09/11/imx233-olinuxino-gpios-faster-and-faster/>) , [5] (<https://github.com/OLIMEX/OLINUXINO/blob/master/SOFTWARE/IMX233/gpio-mmap.h>)

Работа в ядре Linux

Ознакомиться с ядром Linux, используемым в контроллерах WirenBoard можно в нашем репозитории ядра (<https://github.com/wirenboard/linux>).

Рекомендации по Device Tree

Device-tree, используемые на контроллерах WirenBoard, доступны в репозитории ядра. Разные аппаратные ревизии контроллера используют разные *dts* (например, *dts* для WB6.X можно найти здесь (<https://github.com/wirenboard/linux/blob/dev/v4.9.x/arch/arm/boot/dts/imx6ul-wirenboard61.dts>))

Указывать GPIO в Device Tree необходимо для настройки работы GPIO в режиме программного SPI, I2C, для использования GPIO в качестве источника прерываний и т.д. Так, например, на пин 10@UEXT1 (CS) и пины 5@UEXT2 (SCL), 6@UEXT2 (SDA), 10@UEXT2 (CS) выведены линии GPIO процессора. Их можно сконфигурировать для использования, например, в качестве *chip-select* для SPI или в качестве I2C.

GPIO процессора и периферийных устройств разбиты на банки (*gpiochip*). GPIO процессора разбиты на 3 банка по 32 GPIO: *gpio0*, *gpio1*, *gpio2*. Адресация GPIO в Device Tree происходит по номеру банка и номеру GPIO *внутри* банка.

Пример device-tree node

Определим сигнал 6@UEXT2 (SDA) в качестве источника прерываний для драйвера *mrf24j40*. Согласно таблице Список GPIO, сигнал соответствует GPIO 53 процессора. 53 принадлежит второму банку *gpio* (от 32 до 63). Номер GPIO внутри банка 53-32=21 :

```
6lowpan@0 {
    compatible = "microchip,mrf24j40";
    spi-max-frequency = <100000>;
    reg = <6>;
    interrupt-parent = <&gpio1>;
    interrupts = <21 0>;
};
```

RS-485

Contents

Описание

Как правильно проложить шину

Добавление устройства в веб-интерфейс

Как ускорить опрос устройств

Работа с портом RS-485 контроллера из собственного ПО

Описание

RS-485 — стандарт коммуникации по двухпроводной шине.

Теоретически на шину можно подключать до 256 устройств. Длина линии может быть до 1200 метров, но она сильно влияет на скорость передачи данных.

Энциклопедия АСУ ТП. Интерфейс RS-485 (https://www.bookasutp.ru/Chapter2_3.aspx) — подробно про работу интерфейса.

В устройствах Wiren Board используется Протокол Modbus поверх RS-485. Пожалуйста, ознакомьтесь с ним для лучшего понимания работы устройств.

Максимальная скорость передачи данных в периферийных устройствах Wiren Board — до 115 200 бит/с.

Как правильно проложить шину

В статье RS-485:Физическое подключение описано как правильно проложить шину.

Добавление устройства в веб-интерфейс

RS-485:Настройка через веб-интерфейс — что сделать для появления устройства в веб-интерфейсе контроллера.

Как ускорить опрос устройств

Для ускорения опроса устройств по шине RS-485 рекомендуем:

1. Увеличить скорость обмена до 115200 бит/с. На разумных длинах и топологии сети все должно нормально работать. Если на шине есть устройства, не поддерживающие эту скорость, см. пункт 3.
2. Отключить через веб-интерфейс в настройках устройства ненужные каналы.
3. Разделить устройства по типам и портам, контроллере 2 порта RS-485 и еще 3 можно добавить модулями расширения:
 - Устройства, не поддерживающие скорость 115200, подключите отдельно.
 - Счетчики MAP так же подключите отдельно или с оборудованием, не требующим быстрой реакции. В счетчиках очень много параметров, опрос идет медленно.
 - При большом количестве устройств разделите их на несколько портов. При прочих равных скорость вырастеткратно количеству портов.

Работа с портом RS-485 контроллера из собственного ПО

- Стандартно в Wiren Board с подключёнными по RS-485 устройствами работает Драйвер `wb-mqtt-serial` (ранее `wb-home-modbus`). Он позволяет работать с подключёнными устройствами RS-485 через систему MQTT-сообщений.
- Если вы хотите работать с портом RS-485 напрямую, не используя этот драйвер — отключите его, иначе он будет писать в порт RS-485.
- Работа с последовательным портом из Linux
- Доступ к порту RS-485 контроллера Wiren Board с компьютера
- Настройка параметров обмена данными по RS-485 для modbus-устройств Wiren Board

SC16IS752

- English
- русский

SC16IS752 - микросхема, управляющая последовательными портами в некоторых версиях Wiren Board.

Contents

Описание

GPIO

В Wiren Board Smart Home 3.5

UART0

UART1

В Wiren Board 2.8

UART0

UART1

GPIO

Описание

SPI-to-UART bridge IC - NXP SC16IS752

Два полных порта UART, До 8 линий GPIO.

Подключен к SPI. Используется GPIO в качестве chip-select.

Для работы в Linux используется драйвер `sc16is7x2`, который пока не портирован в upstream. Его можно найти в нашем репозитории Linux [1]. Драйвер включён в стандартный образ.

GPIO

Драйвер `sc16is7x2` экспортирует 8 GPIO в стандартный sysfs-интерфейс Linux под номерами 120-127.

В Wiren Board Smart Home 3.5

UART0

Устройство в Linux: `/dev/ttyNSC0`. Полудуплексный RS-485-трансивер. Порт RS-485-1 (зелёная пара клеммников).

Направление работы трансивера задаётся чипом с помощью сигнала RTS автоматически, аппаратный контроль потока должен быть отключен. В следующих версиях драйвера, режим RS-485 возможно будет необходимо активировать из userspace.

UART1

Устройство в Linux: `/dev/ttyNSC1`. Полудуплексный RS-485-трансивер. Порт RS-485-2 (синяя пара клеммников).

Управление устройством/GPIO с помощью Linux: RX/TX управление по портам WB-SH-3.5-HEX2. Для этого необходимо...

управляющие сигналы/GPIO, а также линии кх/лх выведены на разъем wB 3P 3.5: UEXT. Разъем предназначен для использования GPIO, а также для подключения модуля расширения RS232.

В Wiren Board 2.8

UART0

Устройство в Linux: **/dev/ttyNSC0**. TX/RX линии порта выведены на разъём UEXT UEXT2. Параллельно к ним подключен полудуплексный RS-485-трансивер (если есть). Направление работы трансивера задаётся чипом с помощью сигнала RTS автоматически, аппаратный контроль потока должен быть отключен. В следующих версиях драйвера, режим RS-485 возможно будет необходимо активировать из userspace.

UART1

Порт **/dev/ttyNSC1** в Linux. К TX/RX-линиям порта подключен GPS-модуль SIM68V.

GPIO

7 линий GPIO выведены на отверстия с шагом 2.54 на плате *Wiren Board*. К одной из линий GPIO подключен сигнал PPS с GPS-модема.

Драйвер sc16is7x2 экспортирует 8 GPIO в стандартный sysfs-интерфейс Linux под номерами 120-127.

1-Wire в Wiren Board 2.8

Внимание! Эта страница относится только к очень старой версии устройства. Для всех новых смотрите статью [1-Wire](#).

Contents

Подключение

Настройка Device Tree

Работа с 1-Wire в Linux

Работа с температурными датчиками

Подключение

1-Wire датчики можно подключать к линиям GPIO Wiren Board.

Существует два типа подключения датчиков: обычный и с использованием паразитного питания:

- обычное подключение датчиков: подключите линию GND к земле, линию DATA к GPIO, линию VCC к питанию +3.3V или +5V
- подключение с использованием паразитного питания: подключите линию GND к земле, линию DATA к GPIO. Линия VCC должна быть соединена с линией GND.

В любом случае рекомендуется использование внешнего подтягивающего резистора номиналом 2-10 кОм между линией DATA и напряжением питания +3.3V или +5V.

Настройка Device Tree

После подключения 1-Wire сети к какому-либо GPIO необходимо внести соответствующие изменения в файл Device Tree. См. Пересборка Device Tree

```
onewire@0 {
    compatible = "w1-gpio";
    gpios = <gpio0 2 0>;
};
```

Соответствующую запись необходимо добавить внутри корневой секции. Параметр **gpios** указывает на используемый GPIO. В данном примере это GPIO номер 2 т.е. клеммник номер 9 (см. Список GPIO). Подробнее про формат поля **gpios**: Работа с GPIO

Работа с 1-Wire в Linux

Для работы используется ядерный интерфейс в `/sys/bus/w1/`:

```
root@wirenboard:~# ls /sys/bus/w1/
devices drivers drivers_autoprobe drivers_probe uevent
```

```
root@wirenboard:~# ls /sys/bus/w1/devices/
28-000002dc22b0 28-000004a7d3f9 28-000004d01255 28-000004d11d72 w1_bus_master1 w1_bus_master2 w1_bus_master3
```

Здесь **28-000002dc22b0** - идентификаторы обнаруженных устройств на шине, а **w1_bus_master1** - ведущие устройства 1-Wire, каждый из которых соответствует одному GPIO.

Список обнаруженных устройств можно также посмотреть командой:

```
root@wirenboard:~# cat /sys/bus/w1/devices/w1_bus_master1/w1_master_slaves
28-000004a7d3f9
28-000002dc22b0
28-000004d11d72
28-000004d01255
```

По-умолчанию сканирование шины происходит раз в 10 секунд.

Работа с температурными датчиками

За работу с датчиками температуры отвечает модуль ядра `w1_therm`.

Загрузим модуль (автозагрузка отключена намеренно):

```
root@wirenboard:~# modprobe w1_therm
```

Теперь в директориях устройств-термометров появляется файл `w1_slave`:


```
root@wirenboard:~# ls /sys/bus/w1/devices/28-000004a7d3f9/  
driver id name power subsystem uevent w1_slave
```

С его помощью можно считывать показания термометра:

```
root@wirenboard:~# cat /sys/bus/w1/devices/28-000004a7d3f9/w1_slave  
93 01 4b 46 7f ff 0d 10 32 : crc=32 YES  
93 01 4b 46 7f ff 0d 10 32 t=25187
```

Здесь **t=25187** температура в тысячных долях градусов Цельсия, т.е. **25.187C**.

Последовательно прочитаем показания всех термометров на шине:

```
root@wirenboard:~# for i in /sys/bus/w1/devices/28-*; do cat $i/w1_slave; done  
c4 01 4b 46 7f ff 0c 10 3b : crc=3b YES  
c4 01 4b 46 7f ff 0c 10 3b t=28250  
97 01 4b 46 7f ff 09 10 1c : crc=1c YES  
97 01 4b 46 7f ff 09 10 1c t=25437  
9a 01 4b 46 7f ff 06 10 78 : crc=78 YES  
9a 01 4b 46 7f ff 06 10 78 t=25625  
99 01 4b 46 7f ff 07 10 79 : crc=79 YES  
99 01 4b 46 7f ff 07 10 79 t=25562
```

Device Tree

- English
- русский

Device Tree - файлы описания аппаратной конфигурации. Они используются в Linux и, следовательно в Wiren Board. Модификация Device Tree-файлов может понадобиться при переназначении портов, подключении некоторых UEXT-устройств, при подключении 1-wire датчиков и т.д.

Подробнее про Device Tree: https://en.wikipedia.org/wiki/Device_tree , http://elinux.org/Device_Tree

Device Tree для Wiren Board находится в файле imx23-wirenboard28.dts. В нём также используется файл описания процессора, imx23.dtsi

Файлы Device Tree бывают в текстовом формате dts, который компилируется в бинарный формат dtb

Contents

Загрузка

Пересборка

Загрузка

Компилятор

Компиляция

Установка

Загрузка

В стандартном образе Wiren Board загрузчик U-Boot считывает dtb-файл описания Device Tree и передаёт его ядру. Имя dtb-файла задаётся в файле /boot/uEnv.txt :

```
#These are the default settings for some useful u-boot variables:  
fdt_file=/boot/dtbs/imx23-wirenboard28.dtb
```

Файл dtb находится в /boot/dtbs/imx23-wirenboard28.dtb (для Wiren Board rev. 2.8), в /boot/dtbs/imx23-wirenboard32.dtb (для Wiren Board Smart Home rev. 3.5)

Пересборка

Для внесения изменений в Device Tree надо скачать Device Tree в текстовом формате, скомпилировать файл imx23-wirenboard28.dts и записать результат (imx23-wirenboard28.dtb) в /boot/dtbs/

Другой способ - Сборка ядра

Загрузка

Необходимо скачать из [1] файл imx23-wirenboardXX.dts, соответствующий версии устройства и зависимости. Зависимости на настоящий момент это файлы imx23.dtsi, skeleton.dtsi.

Основной DTS-файл:

imx23-wirenboard32.dts для Wiren Board Smart Home rev. 3.5

imx23-wirenboard28.dts для Wiren Board rev. 2.8

```
$ mkdir dts  
$ cd dts  
  
$ #export WB_BRANCH=v3.12-rc3-imxv5-x0.3 # для ядра 3.12  
$ export WB_BRANCH=v3.13-imxv5-x0.1  
  
$ wget https://raw.githubusercontent.com/contactless/linux/$WB_BRANCH/arch/arm/boot/dts/mxs-pinfunc.h  
  
$ wget https://raw.githubusercontent.com/contactless/linux/$WB_BRANCH/arch/arm/boot/dts/imx23-wirenboard28.dts  
2013-11-17 04:24:28 (37.9 MB/s) - «imx23-wirenboard28.dts» сохранён [5255/5255]  
  
$ wget https://raw.githubusercontent.com/contactless/linux/$WB_BRANCH/arch/arm/boot/dts/imx23.dtsi  
2013-11-17 04:24:33 (272 KB/s) - «imx23.dtsi» сохранён [13052/13052]  
  
$ wget https://raw.githubusercontent.com/contactless/linux/$WB_BRANCH/arch/arm/boot/dts/skeleton.dtsi  
  
$ wget https://raw.githubusercontent.com/contactless/linux/$WB_BRANCH/arch/arm/boot/dts/imx23-pinfunc.h  
  
$ ls  
imx23.dtsi      imx23-wirenboard28.dts  skeleton.dtsi  
imx23-pinfunc.h  mxs-pinfunc.h
```

Компилятор

Компилятор Device Tree в Ubuntu и Debian находится в пакете `device-tree-compiler`. Установим его:

```
$ sudo apt-get install device-tree-compiler
```

Также требуется `gcc`

```
$ sudo apt-get install gcc
```

Компиляция

```
$ gcc -E -Wp,-MD,imx23-wireboard28.dtb.d.pre.tmp -nostdinc -I. -undef -D__DTS__ -x assembler-with-cpp -o .imx23-wireboard28.dtb.dts.tmp imx23-wireboard28.dts  
$ cat .imx23-wireboard28.dtb.dts.tmp | grep -v "^#" | dtc -I dts -O dtb -o imx23-wireboard28.dtb  
DTC: dts->dtb on file "-"
```

Установка

Скомпилированный файл необходимо записать в `/boot/dtsb`. **Обязательно сделайте резервную копию существующего в `/boot/dtsb` файла!**

Сборка ядра Linux

Сборка ядра Linux вручную может понадобиться, например, если нужно включить в ядро модули, отсутствующие в стандартной поставке Wiren Board. Если вы не знаете, для чего вам это нужно, то, скорее всего, вам не нужно собирать ядро вручную.

Начиная с 8 апреля 2021 года, скрипты для сборки deb-пакетов ядра Linux для Wiren Board добавлены в репозиторий с кодом ядра: <http://github.com/wirenboard/linux>.

Contents

Подготовка сборочной машины

Получение исходного кода

Сборка

Сборка deb-пакета

Разные модели контроллеров

Настройка ядра

Сборка вручную

Подготовка сборочной машины

Сборка ядра должна производиться на настольном компьютере, ноутбуке или сервере под управлением Linux. Собирать ядро на самом Wiren Board не стоит - у контроллера не хватит дискового пространства для получения репозитория, а также вычислительной мощности. Даже на настольном компьютере сборка ядра может занять десятки минут.

Сборочные скрипты писались с расчётом на дистрибутивы Debian и Ubuntu, инструкции в этой статье приводятся также из расчёта использования этих дистрибутивов. Инструкция была проверена в Ubuntu 18.04.

Для сборки ядра понадобится установить пакеты с необходимым для сборки ПО:

```
$ sudo apt update && sudo apt install build-essential libncurses5-dev fakeroot lzop bc git
```

Если вы собираете ядро для **Wiren Board 6** и новее, нужно установить компилятор для **arm-linux-gnueabi**:

```
$ sudo apt install gcc-arm-linux-gnueabi
```

Для **Wiren Board 5** и более старых понадобится другой компилятор:

```
$ sudo apt install gcc-arm-linux-gnueabi
```

Получение исходного кода

Исходный код ядра Linux с правками от команды Wiren Board хранится в репозитории на Github. Чтобы получить его на свой компьютер, выполните команды:

```
$ git clone https://github.com/wirenboard/linux
$ cd linux
$ git submodule update --init --recursive
```

Сборка

Все промежуточные и конечные артефакты сборки - объектные файлы, dtb, модули ядра .ko, zImage и т.д. - будут находиться в поддиректории **.build-wbX**, где X зависит от модели контроллера. В этой же директории находится конфигурация ядра (см. ниже).

Чаще всего для использования на контроллере удобней всего собрать deb-пакет с файлами ядра.

Если вам нужны только некоторые бинарные файлы (zImage, модули и dtbs), то после успешной сборки deb-пакета их можно будет найти в сборочной поддиректории **.build-wbX**.

Сборка deb-пакета

```
$ make mrproper
$ KERNEL_FLAVOUR=wb6 VERSION_SUFFIX="--my-test-kernel" ./scripts/package/wb/do_build_deb.sh # Wiren Board 6
```

Можно поменять значение VERSION_SUFFIX на свой вкус согласно правилам оформления версий пакетов в Debian. Это значение будет добавлено в конец номера версии пакета и поможет отличить собранное вручную ядро от ядра из репозитория Wiren Board.

После сборки в корне появятся файлы пакетов (пример для Wiren Board 6):

- linux-image-wb6_4.9.22-wb1~my~test~kernel_armhf.deb - образ ядра, модули и dtbs;
- linux-headers-wb6_4.9.22-wb1~my~test~kernel_armhf.deb - нужен для разработки
- linux-libc-dev_4.9.22-wb1~my~test~kernel_armhf.deb - нужен для разработки

На контроллер достаточно скопировать файл пакета linux-image-wb6, в нём уже содержится всё необходимое.

Разные модели контроллеров

Модель контроллера	KERNEL_FLAVOUR	файл defconfig	сборочная директория	переменная CROSS_COMPILE	файлы dts и dtb
Wiren Board 6	wb6	imx6_wirenboard_defconfig	.build-wb6	CROSS_COMPILE=arm-linux-gnueabi-	imx6ul-wirenboard6*
Wiren Board 5 и ниже	wb2	mxs_wirenboard_defconfig	.build-wb2	CROSS_COMPILE=arm-linux-gnueabi-	imx28-wirenboard5*
Wiren Board 7	wb7	wirenboard7_defconfig	.build-wb7	CROSS_COMPILE=arm-linux-gnueabi-	sun8i-r40-wirenboard7*

Настройка ядра

При запуске скрипт сборки deb-пакета спрашивает, использовать ли конфигурацию по-умолчанию:

```
$ KERNEL_FLAVOUR=wb7 scripts/package/wb/do_build_deb.sh
Building kernel packages for wb7 (Wiren Board 7)
Revision: -wb100
Architecture: armhf
Config: wirenboard7_defconfig
.config already present
Use wirenboard7_defconfig instead? (y/N)
```

Конфигурация по-умолчанию при этом берётся из файлов в **arch/arm/configs/**, например **imx6_wirenboard_defconfig**.

Чтобы поменять конфигурацию, запустите **make** с необходимыми параметрами:

```
$ make KBUILD_OUTPUT=.build-wb6 ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- menuconfig # для Wiren Board 6
```

вместо **menuconfig** можно использовать графический **xconfig**.

После сохранения, конфигурация записется в сборочную директорию: **.build-wbX.config**.

Теперь вы можете собрать ядро с новой конфигурацией, выполнив обычную команду

```
$ KERNEL_FLAVOUR=wb7 scripts/package/wb/do_build_deb.sh
```

и ответив N, чтобы использовать новую конфигурацию, вместо конфигурации по-умолчанию:

```
Use wirenboard7_defconfig instead? (y/N)
```

Когда вы полностью довольны результатом, можно посмотреть изменения относительно исходной версии:

```
# для Wiren Board 6
$ make KBUILD_OUTPUT=.build-wb6 ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- savedefconfig # приводит конфигурацию к стандартному виду и записывает в .build-wb6/defconfig
$ diff -u arch/arm/configs/imx6_wirenboard_defconfig .build-wb6/defconfig # посмотреть её отличия от исходной конфигурации
```

Чтобы заменить исходную версию настроек своей (например, при подготовке патча или pull request):

```
# для Wiren Board 6
$ make KBUILD_OUTPUT=.build-wb6 ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- savedefconfig
$ cp .build-wb6/defconfig arch/arm/configs/imx6_wirenboard_defconfig
```

После этого скрипт **do_build_deb.sh** будет использовать обновлённую конфигурацию при ответе "y".

Сборка вручную

Этот этап не требуется для стандартных задач. Если вы хотите выполнить вручную какой-либо этап сборки, то можно запустить **make** вручную.

```
# для Wiren Board 6
$ make KBUILD_OUTPUT=.build-wb6 ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- zImage dtbs modules -j4
```

(-j4 запускает сборку в 4 потока, можно это убрать или поменять значение на более подходящее вашему компьютеру, идеальное значение - количество ядер CPU)

В результате появятся нужные нам файлы:

- .build-wb6/arch/arm/boot/zImage - образ ядра, который нужно скопировать в /boot/zImage на контроллере;
- .build-wb6/arch/arm/boot/dtb/* - файлы device tree, используемые для настройки оборудования при запуске, копируются в директорию /boot/dtbs/.

Файлы модулей можно найти с помощью команды:

```
$ find . -name '*.ko'
```

Далее файлы модулей копируются на контроллер в директорию /lib/modules/<версия ядра>/kernel/ с сохранением исходного пути, как было при сборке.

Hardware Overview rev. 2.8-rus

- English
- русский

Contents

Описание

- Размеры
- Питание
- Power over Ethernet
- Аккумулятор
- АЦП
- Клеммники
- GPIO
- Debug UART
- Исходные файлы
- Чертеж платы



Wirren Board rev. 2.8

Описание

Ниже дано описание Wirren Board версии 2.8.

Wirren Board состоит из двух печатных плат:

- модифицированной iMX233-OLinuXino-MICRO
- основной платы Wirren Board

Основная плата содержит:

- Wi-Fi на базе чипа Realtek RTL8188
- GSM/GPRS модем - SIMCom SIM900R
- GPS/ГЛОНАСС приёмник SIMCom SIM68V
- NFC на чипе NXP PN532. Антенна для NFC нарисована по периметру платы. See also NFC
- USB to Ethernet конвертер и USB хаб, SMSC LAN9514
- RS-485 - ADM3483.
- 8-канальный аналоговый мультиплексор для АЦП (7 клеммников и напряжение питания). See also ADC.
- Выходы "открытый коллектор" для управления низковольтной нагрузкой
- SPI-to-UART bridge IC - NXP SC16IS752
- SPI GPIO-расширитель - Microchip MCP23S08
- Схема питания на чипе LTC4002

Размеры

Габариты платы - длина - 105 мм (+ 3 мм разъемы Ethernet и USB), ширина - 54 мм (+3 мм разъем питания), высота - 20 мм.

Расстояние между монтажными отверстиями - 96 и 46 мм.

Один из углов сделан с перфорацией для отлома. Это упрощает установку в корпуса, где по углам расположены стойки под винты крышки.

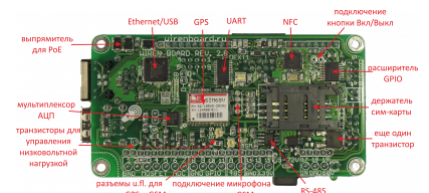
Питание

Напряжение питания 4,7-22В. Среднее потребление платы - 1,5-2 Вт. Но т.к. модуль GPRS потребляет импульсно до 8 Вт, рекомендуется использовать блоки питания с мощностью не менее 10 Вт.

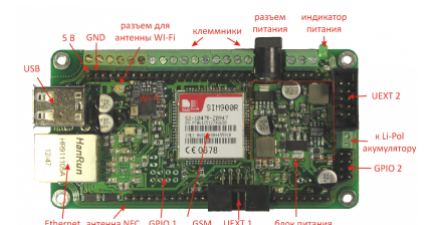
Разъем питания 5.5x2.1мм, также входное напряжение можно подключать к клемме 17.

Кнопка Вкл/Выкл.

Подключете кнопку с фиксацией или тумблер к отверстиям на плате. При замкнутых контактах плата отключается. Заряд аккумулятора при этом продолжается.



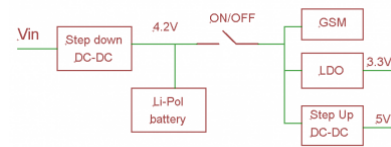
Wirren Board rev. 2.8



Wirren Board rev. 2.8

Power over Ethernet

Только в версии с поддержкой PoE. Версии с поддержкой PoE имеют надпись HLJ-6115ANL на Ethernet-разъёме. Подключение питания по Ethernet к Wiren Board без поддержки PoE (с надписью HanRun на Ethernet-разъёме) приведёт к повреждению устройства!



Блок-схема питания

Подключение PoE совместно с блоком питания приводит к повреждению блока питания.

Поддерживается питание по кабелю Ethernet - Passive PoE напряжением от 7 до 24 Вольт. Питание передаётся по неиспользуемым парам: "+" ("") по паре 4-5 (синий, бело-синий), "-" ("") по паре 7-8 (коричневый, бело-коричневый). Полярность не имеет значения.

См. также Power over Ethernet.

Аккумулятор

Wiren Board позволяет подключить Li-Ion (Li-Pol) аккумулятор с максимальным напряжением 4.2В. При подключении питания аккумулятор заряжается током до 0,5 А (зависит от нагрузки). Есть защита от переразряда аккумулятора, при напряжении ниже 2,9В аккумулятор отключается. Тем не менее, рекомендуется использовать защищённые аккумуляторы.

В текущей версии отсутствует защита от зарядки при отрицательных температурах. Будьте осторожны, аккумуляторы от этого портятся.

Соблюдайте полярность подключения аккумулятора!

АЦП

Пользователю доступно 7 каналов АЦП, выведенных на клеммники (см. таблицу). Максимальное измеряемое напряжение составляет 5В. См. ADC.

Клеммники

См. также Управление низковольтной нагрузкой.

клеммник	функция	Max. V, I	GPIO	вывод olinuxino	состояние по умолчанию*	комментарии
Клеммник 2	FET/ADC	30V, 2A	32	3 @ CON1	Открыт	Выход "открытый коллектор", ADC channel 3
Клеммник 3	FET/ADC	30V, 2A	33	4 @ CON1	Закрыт	Выход "открытый коллектор", ADC channel 0
Клеммник 4	FET/ADC	30V, 2A	34	5 @ CON1	Закрыт	Выход "открытый коллектор", ADC channel 1
Клеммник 5	FET/ADC	30V, 2A	35	6 @ CON1	Открыт	Выход "открытый коллектор", ADC channel 2
Клеммник 6	FET/ADC	30V, 2A	39	10 @ CON1	Неопределено	Выход "открытый коллектор", ADC channel 4
Клеммник 7	FET/ADC	30V, 2A	1	12 @ CON1	Неопределено	Выход "открытый коллектор", ADC channel 6
Клеммник 8	GND					
Клеммник 9	GPIO/ADC	3.3V, 10mA	2	13 @ CON1	47K pull-up	ADC channel 7
Клеммник 10	GPIO	3.3V, 10mA	3	14 @ CON1	47K pull-up	
Клеммник 11	GPIO	3.3V, 10mA	4	15 @ CON1	47K pull-up	
Клеммник 12	RS-485 A			UART1 от sc16is752		если есть RS-485
Клеммник 13	RS-485 B			UART1 от sc16is752		если есть RS-485
Клеммник 14	GND					
Клеммник 15	+3.3v	0.5A				
Клеммник 16	+5.0v	0.5A				
Клеммник 17	Vin	22				Входное напряжение; ADC channel 5
Клеммник	GND					

18						
Клеммник 19	Power FET	60V, 3A		28 @ CON2	Неопределено	с аппаратным ШИМом

/* При загрузке системы транзисторы будут в этих состояниях. Это особенность (баг) версии WB2.8.

GPIO

См. Список GPIO.

Debug UART

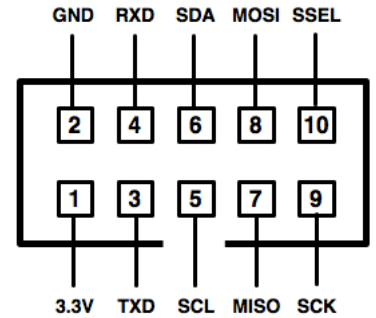
Выведен на разъём UEXT1 (чёрный разъём с десятью штырьками на длинной стороне платы).

Исходные файлы

См. Wiren Board 2.8. Исходные файлы

Чертеж платы

Файл:WB 2.8 chertej.pdf



UEXT pinout for IDC connector
(looking into connector on host board)

Подключение периферийных устройств к контроллеру Wiren Board

Contents

Управление низковольтной нагрузкой

Технические подробности
Подключение нагрузки

Датчики с аналоговым выходом по напряжению

Датчики с аналоговым токовым выходом

Датчики/счетчики с импульсными выходами/кнопки

Устройства с выходом открытый коллектор

Контакты с управляющим напряжением 220В

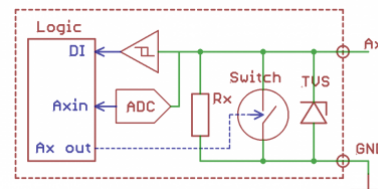


Схема входов/выходов A1-A4

Управление низковольтной нагрузкой

Технические подробности

Для управления низковольтной нагрузкой в Wiren Board предназначены так называемые «транзисторные выходы», они же FET или **открытый коллектор**. С их помощью можно управлять включением низковольтных ламп, светодиодных лент, внешних блоков реле и т.п.

Транзисторными выходами можно управлять из веб-интерфейса, там они называются соответственно клеммам: **A1_OUT** — **A4_OUT**.

Для сокращения общего числа клеммников каналы управления низковольтной нагрузкой совмещены с каналами АЦП. Поэтому выходы имеют большое, но конечное сопротивление — 100кОм. Это может вызывать, например, слабое свечение светодиодных лент, но проблему можно решить — подтянуть вывод резистором к питанию.

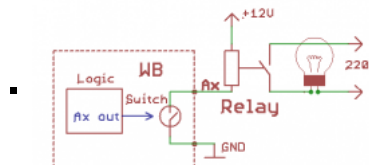
В контроллерах Wiren Board 6 выходы защищены от импульсных перенапряжений, короткого замыкания и перегрева.

Подключение нагрузки

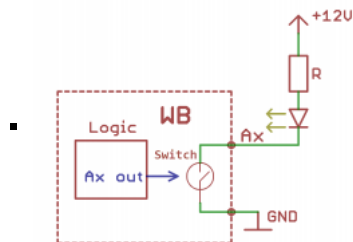
Чтобы подключить нагрузку, подключите «плюс» нагрузки к источнику питания, а «минус» к транзисторному входу. Нагрузка включается подачей высокого уровня на выход. Если суммарный ток на канале превышает 2 А — дополнительно подключите клемму **GND** к минусу источника питания.

При управлении индуктивной нагрузкой (реле), возникают всплески напряжения. Для защиты от перенапряжения в контроллер встроены защитные диоды — внешних защитных элементов не требуется.

Также для управления низковольтной нагрузкой можно использовать модуль дискретных выходов WBIO-DO-HS-8.



Пример подключения реле к выходам A1-A4



Датчики с аналоговым выходом по напряжению

Клеммы A1-A4 могут измерять напряжение, поэтому к ним можно подключить датчики с аналоговым выходом по напряжению, например, температурные сенсоры.

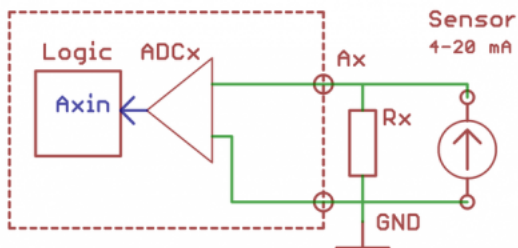
Подключите землю устройства к клемме GND, или соедините с общей земляной шиной. Выход датчика подключите к одной из клемм **Ax**.

Для точного измерения напряжения можно использовать модуль ввода-вывода WBIO-AI-DV-12_I/O_Module или модуль аналоговых входов WB-MAI11.

Датчики с аналоговым токовым выходом

Специальных токовых входов в WB нет, но можно, используя резистор $R_x = 100-300\text{ Ом}$, ток преобразовать в напряжение и подключить по аналогии с датчиком, имеющим аналоговый выход по напряжению.

Так же можно использовать модуль ввода-вывода WBIO-AI-DV-12_4-20mA или модуль аналоговых входов WB-MAI11.



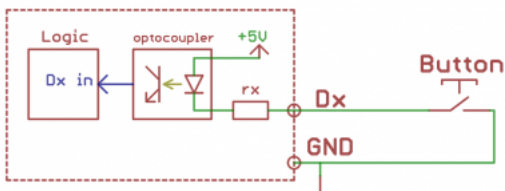
Датчики/счетчики с импульсными выходами/кнопки

Такие устройства формируют сигнал, замыкая подходящие к ним два провода.

Способы подключения к контроллеру:

1. С помощью клемм **Ax** контроллера. Подключите один провод к источнику питания 5-24 В, второй провод к клемме **Ax**. Подробнее смотрите на странице Подключение устройств с импульсными выходами.
2. С помощью модуля ввода-вывода WBIO-DI-WD-14 (14 каналов). Один из проводов подключите к GND, второй к клемме **Dx** модуля.
3. С помощью модуля расширения WBE2-DI-DR-3 (3 канала). Один из проводов подключите к GND, второй к клемме **Ox** соответствующей модулю расширения.

Некоторые счетчики имеют импульсный выход на оптроне, тогда два провода имеют полярность — «плюс» и «минус». В таком случае минус подключается к **GND**, а «плюс» ко входу. Либо для первого способа — «плюс» к питанию, а «минус» к **Ax**.



Устройства с выходом открытый коллектор

Есть три способа подключить такие устройства к контроллеру:

1. С помощью модуля ввода-вывода WBIO-DI-WD-14 (14 каналов). Выход «открытый коллектор» подключите к клемме **Dx** модуля. Землю устройства к iGND модуля.
2. С помощью модуля расширения WBE2-DI-DR-3 (3 канала). Выход «открытый коллектор» подключите к клемме **Ox** соответствующей модулю расширения, а землю устройства к GND контроллера.
3. Можно подключать к клеммам **A1-A4**, при этом нужно также подключить внешний подтягивающий резистор между **5Vout** и соответствующей клеммой **Ax** номиналом около 10 кОм. Соедините земли устройства и контроллера.

Контакты с управляющим напряжением 220В

Используйте модуль ввода-вывода с релейными выходами, например WBIO-DO-R1G-16.

Подключите управляющую катушку контактора через реле модуля расширения, схему подключения смотрите в разделе «Монтаж» на странице используемого модуля.

Модуль WBIO-DO-R1G-16 содержит TVS, защищающий контакты реле от искрения. Внешние защитные компоненты не требуются.

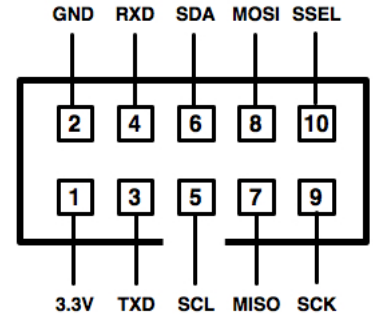
UEXT-разъемы

- English
- русский

Universal EXTension (UEXT) - схема разъема, включающего в себя линию питания и три последовательных интерфейса передачи данных: UART, I²C, SPI. Разработан компанией Olimex и является открытым и бесплатным. UEXT specification.

UEXT разъем состоит из 10 пинов, по 5 пинов в два ряда в пластиковой оправе. Расстояние между пинами 2.54 mm. Разъем соединяется с 2x5 (10-pin) IDC розеткой, и обычно соединяется с плоским кабелем. См. также [1].

Технические характеристика



UEXT pinout for IDC connector
(looking into connector on host board)

UEXT-разъемы в плате Wiren Board

На плате Wiren Board v.2.8 присутствуют два UEXT-разъема. **Но во втором разъеме линии MISO, MOSI, SCK дублированы с соответствующими выводами первого разъема. А линии TXD и RXD дублированы с входами RS-485.**

UEXT1					
Pin	Name	I/O	Logic	-	-
1	3.3V	S	S	+3.3 volt	
2	GND	S	S	Ground	
3	TXD	O	PP	Debug UART Tx	
4	RXD	I	PP	[Debug UART]] Rx	
5	SCL	O	OD	Аппаратный I ² C, SCL	
6	SDA	I/O	OD	Аппаратный I ² C, SDA	
7	MISO	I	PP	Шина SPI, MISO	
8	MOSI	O	PP	Шина SPI, MOSI	
9	SCK	O	PP	Шина SPI, SCK	
10	SSEL	O	PP	GPIO 19	Шина SPI, CS

UEXT2						
Pin	Name	I/O	Logic	-	-	примечания
1	3.3V	S	S	+3.3 volt		
2	GND	S	S	Ground		
3	TXD	O	PP	sc16is752 UART0 Tx		параллельно RS-485
4	RXD	I	PP	sc16is752 UART0 Rx		параллельно RS-485
5	SCL	O	OD	GPIO 51	I ² C, SCL	
6	SDA	I/O	OD	GPIO 53	I ² C, SDA	
7	MISO	I	PP	Шина SPI, MISO		
8	MOSI	O	PP	Шина SPI, MOSI		
9	SCK	O	PP	Шина SPI, SCK		
10	SSEL	O	PP	GPIO 91	Шина SPI, CS	

Порты 3,4 (UART) разъёма UEXT2 подключены к чипу SC16IS752 и присутствуют только в модификациях с этим чипом (опции GPS или RS-485). Эти порты подключены параллельно трансиверу RS-485, их невозможно использовать одновременно с RS-485.

Ссылки

- [UEXT connector webpage](#)
- [UEXT specification](#)
- [OLIMEX Ltd](#)

Питание USB-портов

- English
- русский

пакет hubpower из нашего репозитория.

Работа:

```
root@wirenboard:~# hubpower 1:2 status
Port 1 status: 0503 High-Speed Power-On Enabled Connected
Port 2 status: 0100 Power-On
Port 3 status: 0100 Power-On
Port 4 status: 0100 Power-On
Port 5 status: 0503 High-Speed Power-On Enabled Connected
```

Включение/выключение порта:

```
root@wirenboard:~# hubpower 1:2 power 4 off
Port 4 status: 0000 Power-Off
root@wirenboard:~# hubpower 1:2 power 4 on
Port 4 status: 0100 Power-On
```

Contents

Wiren Board 6

Wiren Board 5

Wiren Board 4

Wiren Board Smart Home rev. 3.5

Wiren Board 6

Второй внешний USB-порт:

```
# выключить
$ hubpower `lsusb | grep "0424:2514" | sed 's/^Bus 0*\([[[:digit:]]*\) Device 0*\([[[:digit:]]*\).*$/\1:\2/g'` power 4 off
# включить
$ hubpower `lsusb | grep "0424:2514" | sed 's/^Bus 0*\([[[:digit:]]*\) Device 0*\([[[:digit:]]*\).*$/\1:\2/g'` power 4 on
```

Модуль Wi-Fi:

```
# выключить
$ hubpower `lsusb | grep "0424:2514" | sed 's/^Bus 0*\([[[:digit:]]*\) Device 0*\([[[:digit:]]*\).*$/\1:\2/g'` power 1 off
# включить
$ hubpower `lsusb | grep "0424:2514" | sed 's/^Bus 0*\([[[:digit:]]*\) Device 0*\([[[:digit:]]*\).*$/\1:\2/g'` power 1 on
```

Вместо

```
$ hubpower `lsusb | grep "0424:2514" | sed 's/^Bus 0*\([[[:digit:]]*\) Device 0*\([[[:digit:]]*\).*$/\1:\2/g'` ...
```

в большинстве случаев можно писать

```
$ hubpower 2:2 ...
```

Wiren Board 5

Внешний USB-порт:

```
# выключить
$ hubpower 1:1 power 1 off
# включить
$ hubpower 1:1 power 1 on
```

Модуль Wi-Fi:

```
# выключить
$ hubpower 2:1 power 1 off
# включить
$ hubpower 2:1 power 1 on
```

Wiren Board 4

Номера портов (для управления питанием):

4 - встроенный WiFi
3 - порты USB-Hub, выход 5V
1 - Ethernet-часть LAN9514

Пример (отключение Wi-Fi):

```
hubpower 1:2 power 4 off
```

Wiren Board Smart Home rev. 3.5

Номера портов (для управления питанием):

4 - встроенный WiFi
3 - порты USB-Hub
1 - Ethernet-часть LAN9514

Wiren Board 2.8. Исходные файлы

- English
- русский

Файл:WB 2.8 shema.pdf

файлы для EAGLE: Файл:WB 2.8.rar

Бесплатная версия программы с официального сайта EAGLE

Debian/Ubuntu: apt-get install eagle

Периферийные устройства

- English
- русский

Contents

Периферия

Экраны

Serial TFT

Zigbee MiWi transciever

Интерфейсы

RS-232

RS-485

Считыватели бесконтактных карт и клавиатуры

Wiegand

Приборы учёта

Modbus-устройства

Кнопки и сухие контакты

Периферия

Экраны

Поддерживается подключение различных экранов по последовательному интерфейсу. Также поддерживается подключение символьных экранов в параллельном режиме.

Serial TFT

Многие дисплеи поддерживаются драйвером <https://github.com/notro/fbtf>, который пока не интегрирован в mainline ядро. Список поддерживаемых драйвером моделей: <https://github.com/notro/fbtf/wiki/LCD-Modules>

Ниже приведён список протестированных с Wiren Board экранов.

- 2.2' TFT LCD: TFT01-2.2SP от Elecfreacs и клоны, 320x240 цветной, чип ILI9341, от \$6.5 в рознице

Zigbee MiWi transciever

MOD-MRF24J40

Интерфейсы

RS-232

- MOD-RS232 - только TX/RX. Подключение до двух модулей к плате. Для подключения используется UEXT1 (необходимо отключить Debug UART) и UEXT2 (нельзя использовать одновременно с RS-485)
- Переходники USB-RS-232. До двух кабелей без использования внешнего USB-хаба. Возможна программная перезагрузка по питанию

RS-485

- <https://www.olimex.com/Products/Modules/Interface/MOD-RS485-ISO/> гальванически изолированный, full/half duplex, I2C-интерфейс, можно использовать совместно с Debug UART и RS-485. Возможно подключение больше двух устройств.
- <https://www.olimex.com/Products/Modules/Interface/MOD-RS485> - half duplex, использует UART. До двух модулей.

Считыватели бесконтактных карт и клавиатуры

Wiegand

Стабильная работа с wiegand не гарантируется, рекомендуется использовать другие протоколы

- Подключать линии D0 и D1 считывателей к GPIO необходимо через резисторы 1-20 кОм! (на Wiren Board 2.8)
- См. <https://github.com/contactless/wirenboard/tree/master/examples/wiegand>
- См. <https://github.com/contactless/wiegand-linux>

Приборы учёта

Приборы учёта с импульсным выходом

Modbus-устройства

Устройства с интерфейсом Modbus

Кнопки и сухие контакты

См.

<http://contactless.ru/forums/topic/%D0%BF%D0%BE%D0%B4%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%BD%D0%B8%D0%B%D0%BA%D0%BD%D0%BE%D0%BF%D0%BE%D0%BA-%D0%BA-gpio/#post-2384> См. Приборы учёта с импульсным выходом

Retrieved from "https://wirenboard.com/wiki/Служебная:Print/"

- Privacy policy
- About Wiren Board
- Disclaimers
-