RBS computers

Модель

RBSComputers Allwinner A20

Сайт разработчика
rainbowsoft.ru

Сайт проекта
rbs-computers.ru

Подготовка к работе

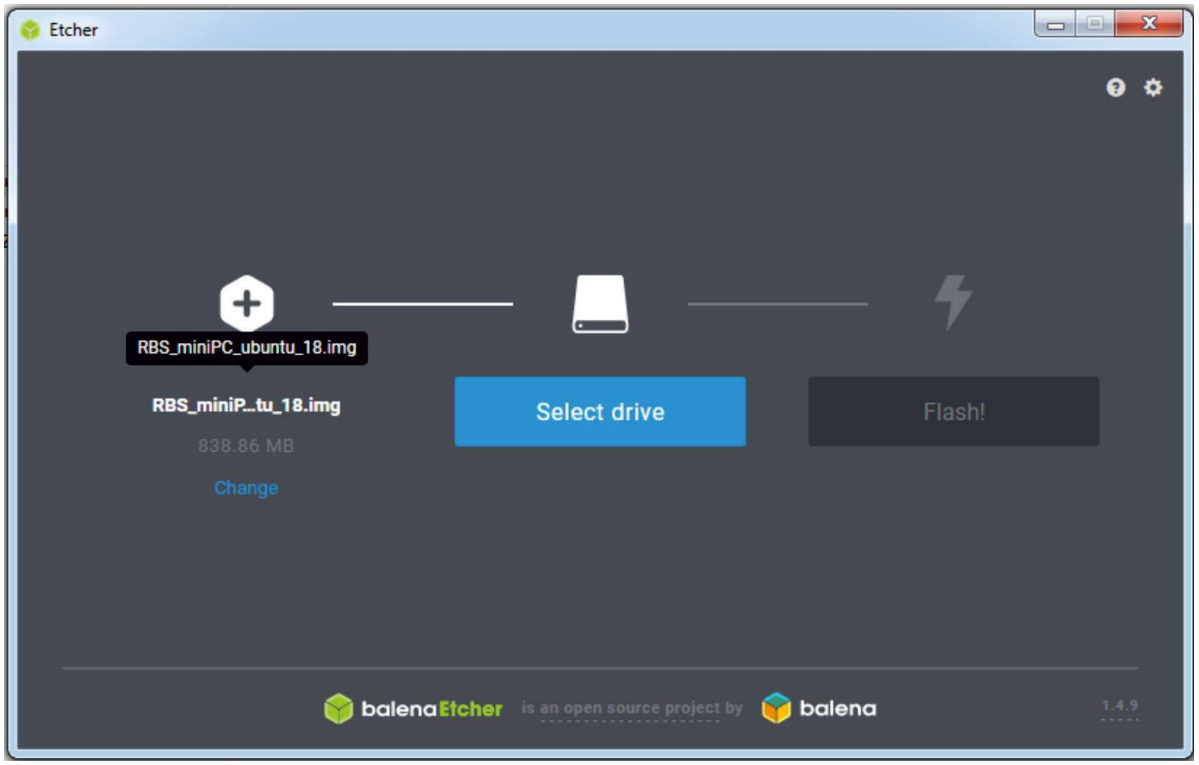
Установка операционной системы

Мини ПК RBS работает под управлением операционных систем на ядре Linux.
Операционная система находится на SD-карте.
SD-карта может быть объемом от 2Гб для серверной версии и от 4 Гб для десктоп версии и классом скорости от 4. Оптимальный вариант 8,16 Гб **класс 10**, еще лучше - **A1**, **A2, класс UHS U1, U3/**
Среди брендов отдавайте предпочтение Samsung, SanDisk, Toshiba, Transcend.

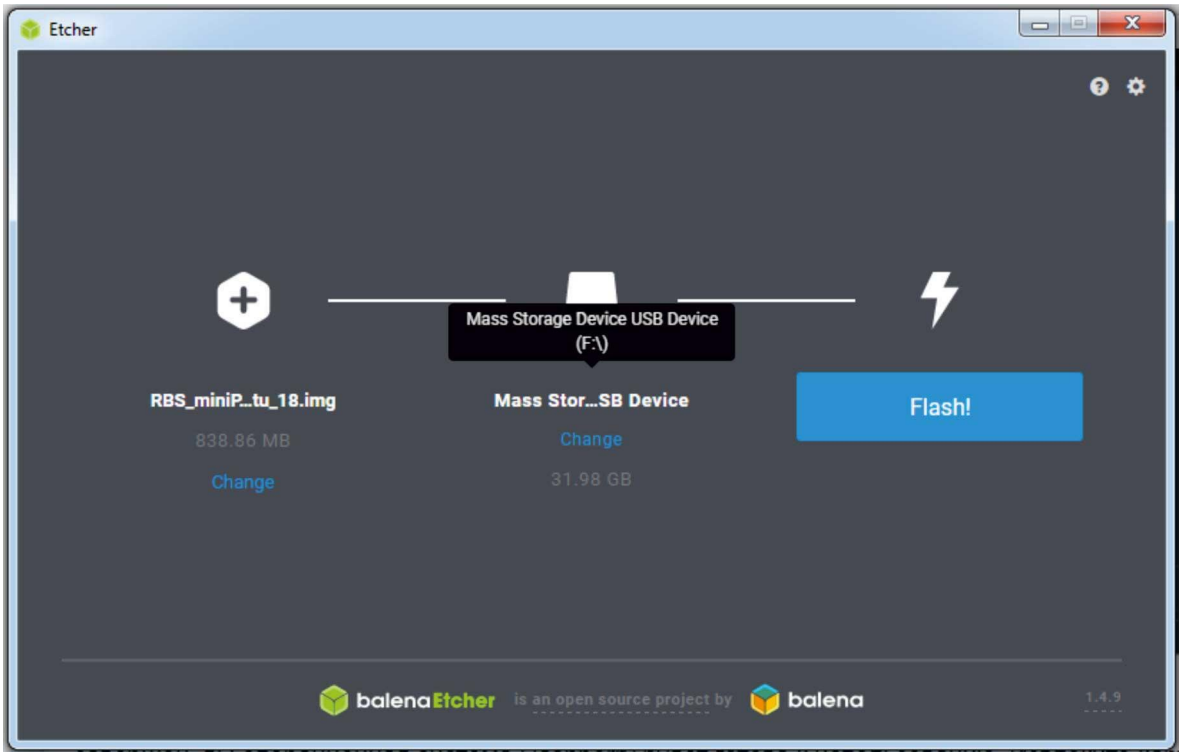
MicroSD карта памяти

Запись образа на карту памяти

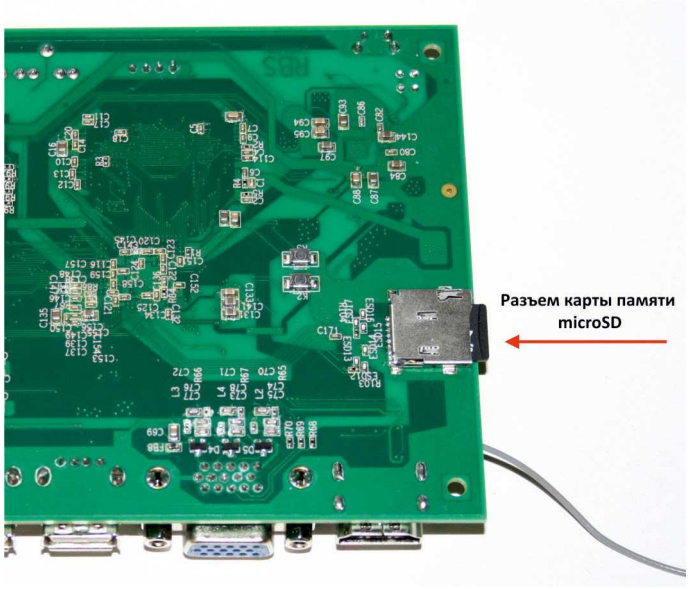
Для работы предоставляется образ с установленной операционной системой. Для записи образа на карту памяти, скачайте и установите утилиту Ether. Вставьте карту памяти в устройство для чтения/записи. Запустите утилиту Etcher и выберите нужный образ в разделе **"Select Image"**.



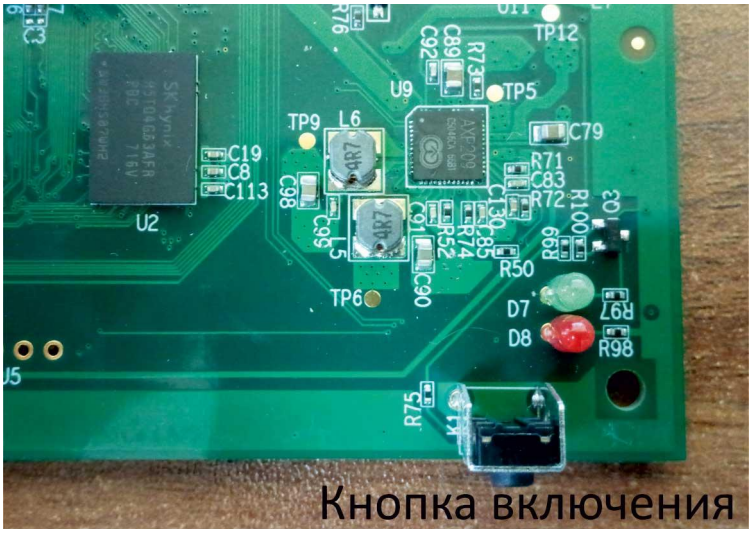
Убедитесь, что выбран правильный диск. Если устройство нужно изменить, нажмите кнопку **"Change"**



Для начала записи нажмите кнопку **"Flash"**. После завершения установите карту памяти в соответствующий разъем на печатной плате мини ПК.



Подключите необходимые устройства - монитор, клавиатуру, мышь. Последним подключайте блок питания. После подключения блока питания устройство включится. После завершения работы вновь включить устройство можно кнопкой, для этого удерживайте кнопку включения около 2х секунд.
Если блока питания нет в комплекте, то блок питания должен соответствовать следующим характеристикам: Напряжение 5V ток не менее 2А разъем 5,5*2,1 мм.



По завершению загрузки ОС введите логин и пароль
(первая загрузка системы займет больше времени, мигание зеленого светодиода говорит о активности CPU, без паники!)
Логин / пароль по умолчанию:
root / rbs //Суперпользователь, полные права.
user / rbs //Пользователь с ограниченными правами.

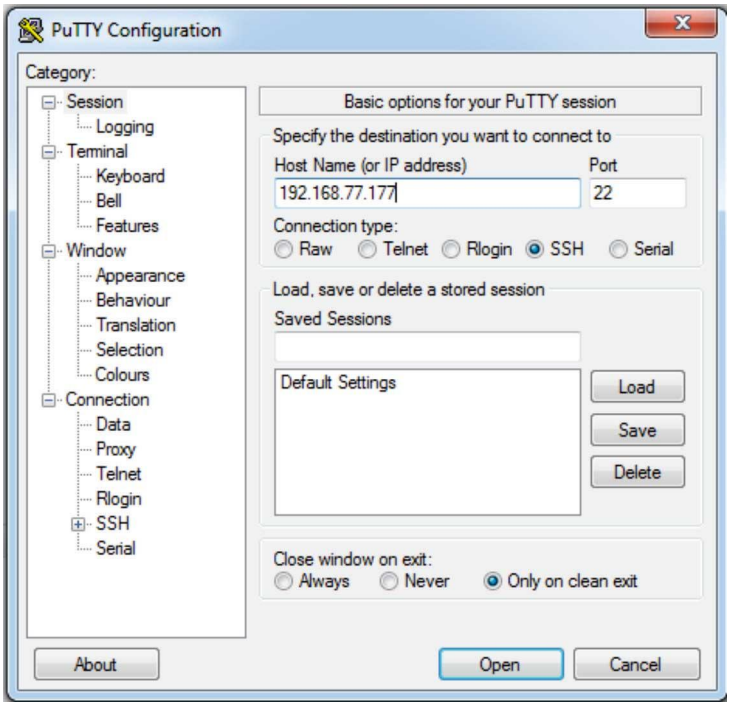
Теперь нужно увеличить свободное место файловой системы на весь диск, откройте консоль и выполните команды:
(Пример для флешки на 4Гб)
parted
(parted) p //посмотрели размер диска Disk /dev/mmcblk0
(parted) resizepart 1 // изменим размер первой партиции
(parted) xxxxXX // указали нужный размер
(parted) p //убедились в изменениях
(parted) q
resize2fs /dev/mmcblk0p1 // resize2fs /dev/mmcblk2p1 для второй версии мини ПК
проверяем:
df -h
Filesystem Size Used Avail Use% Mounted on
/dev/root 3,7G 551M 3,0G 16% /

Удаленная работа с мини ПК по локальной сети

При наличии настроенной локальной сети работать с мини пк возможно удаленно с помощью SSH клиента.
SSH (англ. Secure SHell — защищенная оболочка) — сетевой протокол прикладного уровня, предназначенный для безопасного удаленного доступа к UNIX-системам.
Скачайте и установите SSH клиент PuTTY

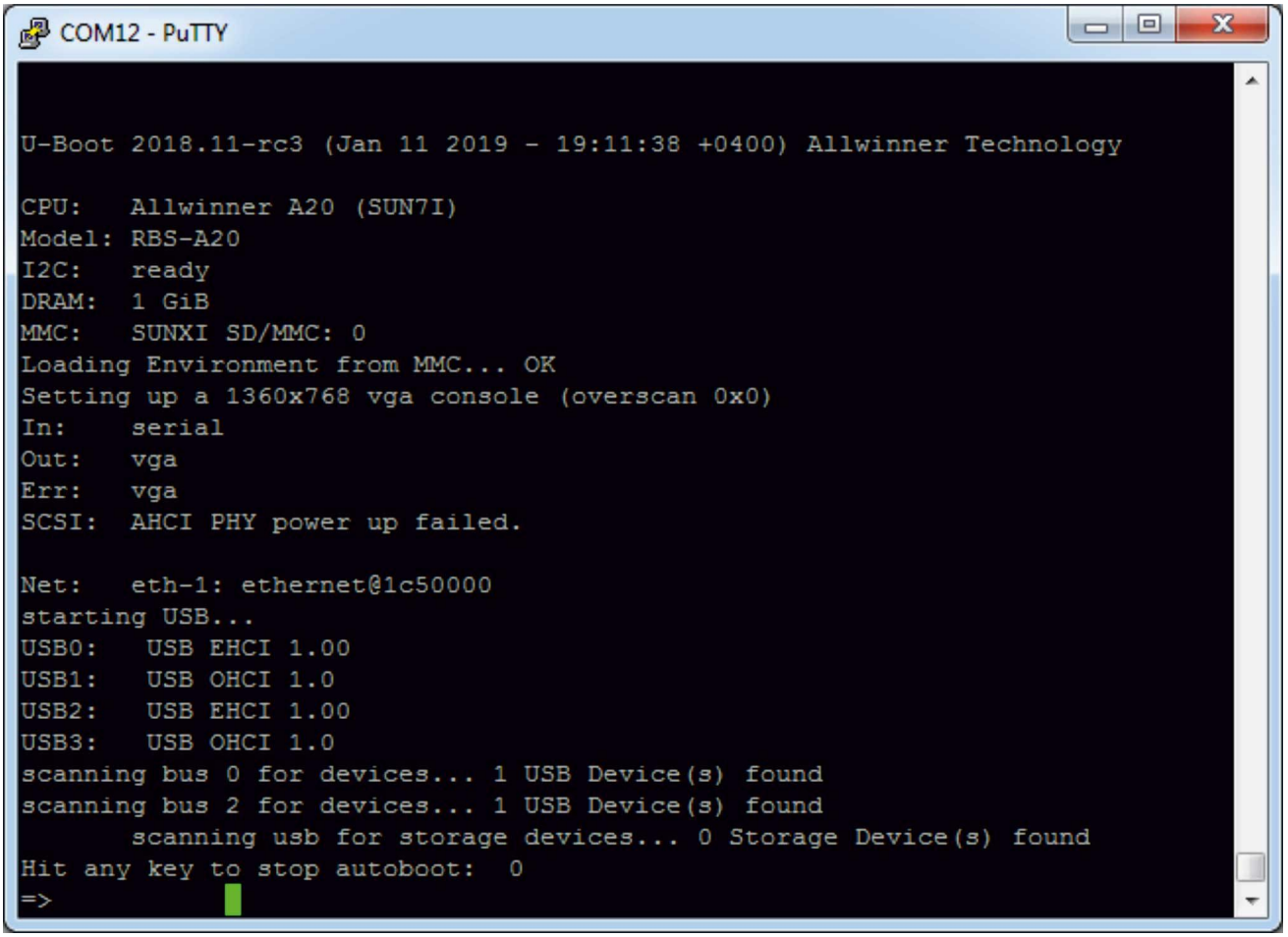
Узнайте сетевой адрес вашего устройства с помощью команды:
ip a
Установите на мини ПК сервер SSH с помощью команды:
apt install ssh

Откройте SSH клиент на машине с которой вы будете управлять мини ПК, введите адрес в соответствующее поле. После нажмите “open” при первом подключении к серверу, ssh вас спрашивает, доверяете ли вы ключу. Отвечаем согласием.



Изменение разрешения

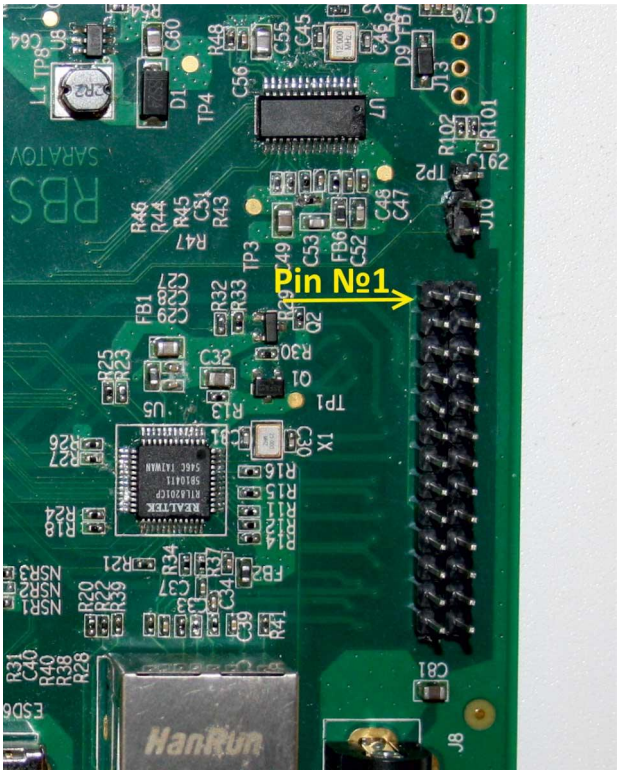
Для мониторов с HDMI разъемом оптимальное разрешение определяется автоматически.
Для мониторов подключенных к VGA разъему оптимальное разрешение указывается следующим образом:
Для изменения разрешения необходимо:
Подключиться к мини ПК через Serial порт.
Включить мини ПК,и в этот же момент многократно нажимать на клавиатуре любой символ, при этом будет остановлен процесс загрузки.

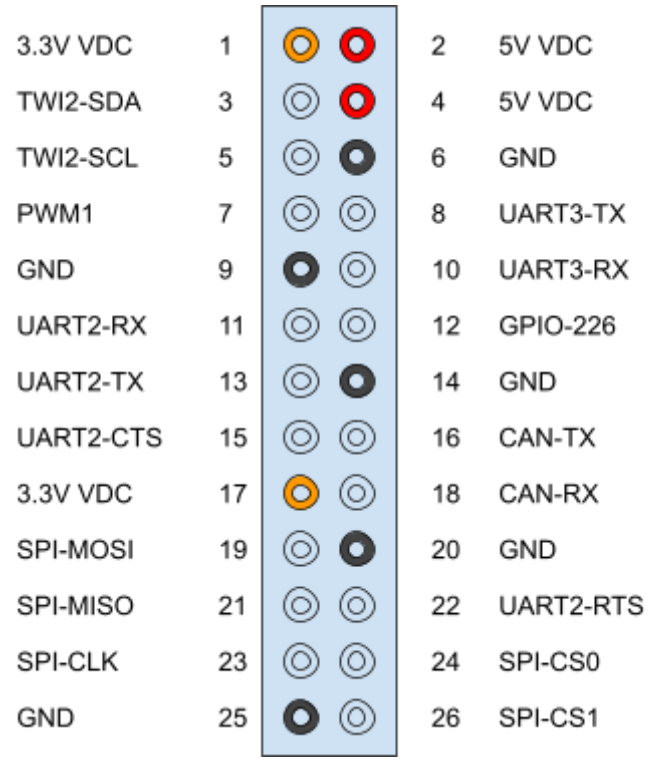


Далее поочередно набираем или копируем из инструкции строки с командами:
setenv video-mode sunxi:1360x768-24@60,monitor=vga,edid=1 (1360x768 нужное нам разрешение)
saveenv
reset
Произойдет перезагрузка, загрузится мини ПК теперь с новым разрешением - 1360x768

Работа с портами ввода вывода

Распиновка разъема





Распиновка и внешний вид на печатной плате

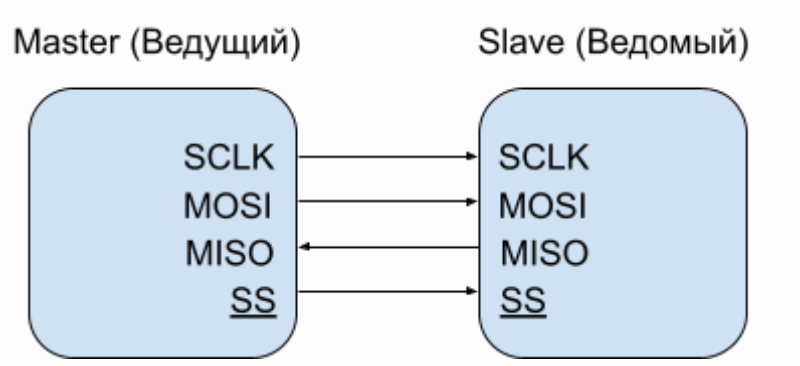
Вывод питание 3.3V	1	2	Вывод питание 5V
TWI (I2C) - SDA линия данных	3	4	Вывод питание 5V
TWI (I2C) - SCL линия тактовая	5	6	Общая шина (земля)
PWM - ШИМ	7	8	UART3-TX Последовательный порт прием
Общая шина (земля)	9	10	UART3-RX Последовательный порт передача
UART2-RX Последовательный порт прием	11	12	GPIO-226 Интерфейс ввода/вывода
UART2-TX Последовательный порт передача	13	14	Общая шина (земля)
UART2-CTS Готовность к передаче	15	16	CAN-TX Передача
Вывод питание 3.3V	17	18	CAN-RX Прием
SPI-MOSI выход ведущего, вход ведомого	19	20	Общая шина (земля)
SPI-MISO вход ведущего, выход ведомого	21	22	UART2-RTS Запрос на передачу
SPI-CLK тактовый сигнал	23	24	SPI-CS0 выбор ведомого 0
Общая шина (земля)	25	26	SPI-CS1 выбор ведомого 1

Схема 1. Распиновка разъема с портами ввода вывода

Имеющиеся интерфейсы

GPIO - Интерфейс ввода/вывода общего назначения (англ. general-purpose input/output, GPIO) — интерфейс для связи между компонентами компьютерной системы, к примеру микропроцессором и различными периферийными устройствами. Контакты GPIO могут выступать как в роли входа, так и в роли выхода — это, как правило, конфигурируется. GPIO контакты часто группируются в порты.

SPI - (англ. *Serial Peripheral Interface, SPI bus* — последовательный периферийный интерфейс, шина SPI) — последовательный синхронный стандарт передачи данных в режиме полного дуплекса, предназначенный для обеспечения простого и недорогого высокоскоростного сопряжения микроконтроллеров и периферии. SPI также иногда называют четырёхпроводным (англ. *four-wire*) интерфейсом.

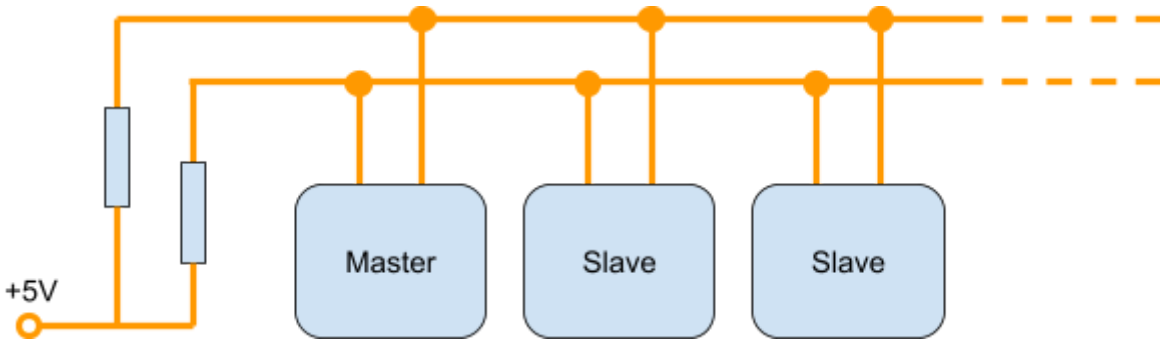


В отличие от стандартного последовательного порта (англ. *standard serial port*), SPI является синхронным интерфейсом, в котором любая передача синхронизирована с общим тактовым сигналом, генерируемым ведущим устройством (процессором). Принимающая (ведомая) периферия синхронизирует получение битовой последовательности с тактовым сигналом. К одному последовательному периферийному интерфейсу ведущего устройства-микросхемы может присоединяться несколько микросхем. Ведущее устройство выбирает ведомое для передачи, активируя сигнал «выбор кристалла» (англ. *chip select*) на ведомой микросхеме. Периферия, не выбранная процессором, не принимает участия в передаче по SPI.

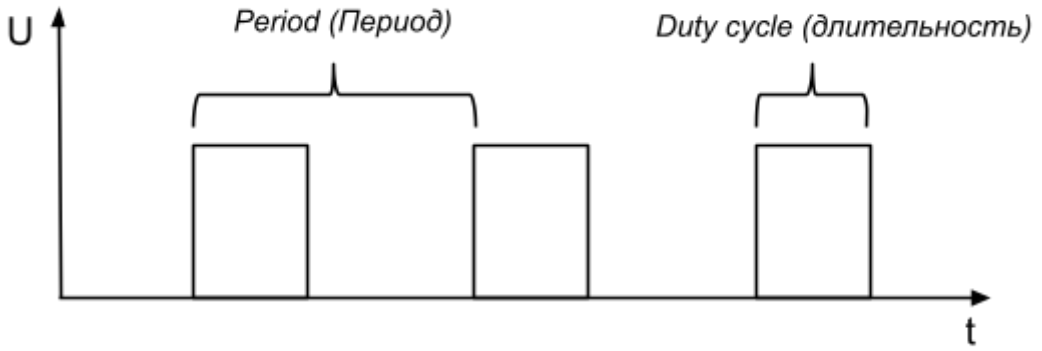
В SPI используются четыре цифровых сигнала:

- MOSI — выход ведущего, вход ведомого (англ. *Master Out Slave In*). Служит для передачи данных от ведущего устройства ведомому.
- MISO — вход ведущего, выход ведомого (англ. *Master In Slave Out*). Служит для передачи данных от ведомого устройства ведущему.
- SCLK или SCK — последовательный тактовый сигнал (англ. *Serial Clock*). Служит для передачи тактового сигнала для ведомых устройств.
- CS или SS — выбор микросхемы, выбор ведомого (англ. *Chip Select, Slave Select*).

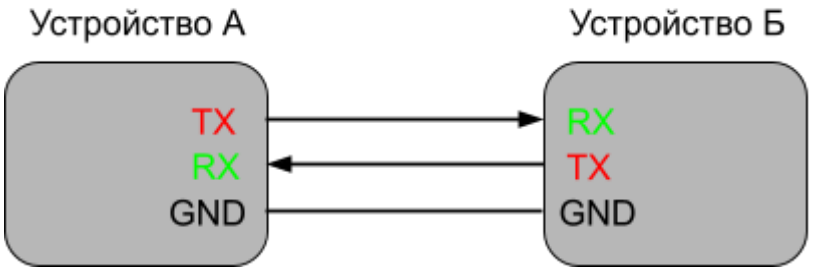
I2C - I²C (IIC, англ. *Inter-Integrated Circuit*) — последовательная асимметричная шина для связи между интегральными схемами внутри электронных приборов. Использует две двунаправленные линии связи (SDA и SCL), применяется для соединения низкоскоростных периферийных компонентов с процессорами и микроконтроллерами (например, на материнских платах, во встраиваемых системах, в мобильных телефонах).



PWM - Широтно-импульсная модуляция (ШИМ, англ. *pulse-width modulation (PWM)*) — процесс управления мощностью, подводимой к нагрузке, путём изменения скважности импульсов.



UART - Универсальный асинхронный приёмопередатчик (УАПП, англ. *Universal Asynchronous Receiver-Transmitter, UART*) — узел вычислительных устройств, предназначенный для организации связи с другими цифровыми устройствами. Преобразует передаваемые данные в последовательный вид так, чтобы было возможно передать их по одной физической цифровой линии другому аналогичному устройству. Метод преобразования хорошо стандартизован и широко применяется в компьютерной технике (особенно во встраиваемых устройствах и системах на кристалле (SoC)).



CAN - (англ. *Controller Area Network* — сеть контроллеров) — стандарт промышленной сети, ориентированный, прежде всего, на объединение в единую сеть различных исполнительных устройств и датчиков. Режим передачи — последовательный, широковещательный, пакетный.

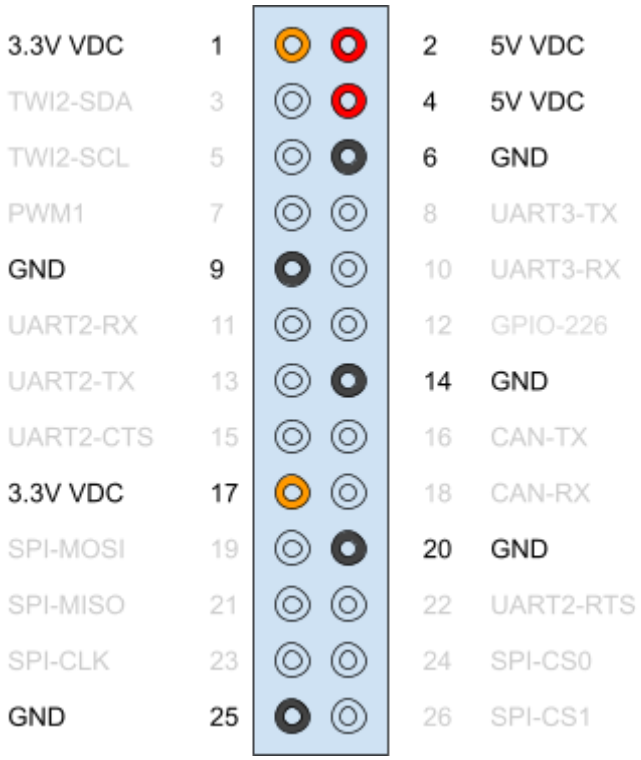
CAN разработан компанией Robert Bosch GmbH в середине 1980-х и в настоящее время широко распространён в промышленной автоматизации, технологиях «умного дома», автомобильной промышленности и многих других областях. Стандарт для автомобильной автоматики.

Руководство по использованию

Ограничение при работе с портами

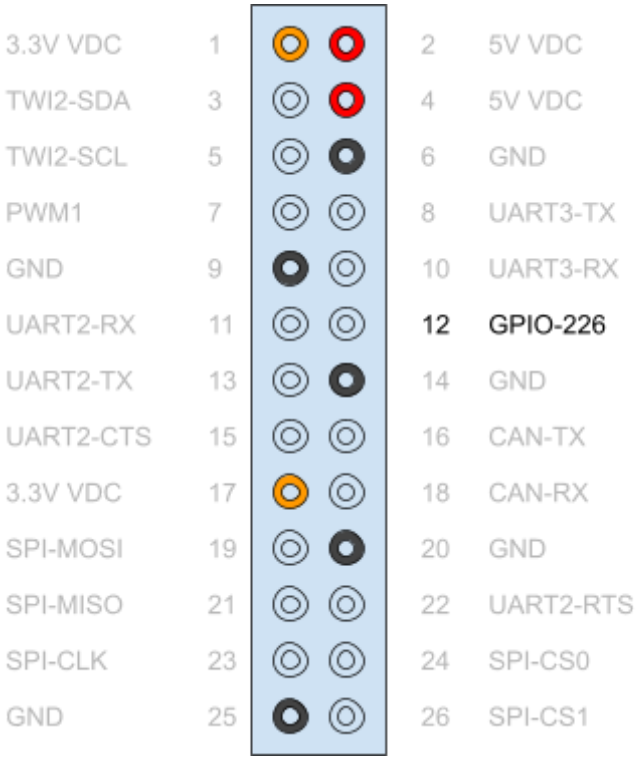
Защита от превышения напряжения и тока отсутствует!
Ток нагрузки не более - 20mA
(Подключение к примеру светодиода без токоограничивающего сопротивления повредит соответствующий вывод!)
Сопротивление токоограничивающего резистора от 150 Ом!
Напряжение логического уровня 1 - 3.3В
Максимальная нагрузка на линии 5В
Максимальная нагрузка на линии 3.3В

Линии питания



5V Pins 2, 4
3.3V Pins 1, 17
Общая шина (GND) Pins 9, 25, 20, 14, 6

Работа с GPIO



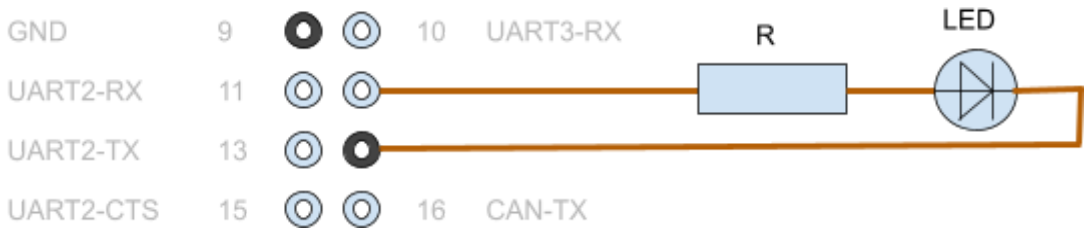
(gpio-112 port PD16 - занят, управляет светодиодом на плате в качестве индикатора активности CPU)
gpio-226 Pin 12 (port PH02)

Получить список всех портов доступных в системе, обозначенных в дереве устройств:
cat /sys/kernel/debug/pinctrl/1c20800.pinctrl/pinmux-pins |grep P

Литература:
https://elinux.org/images/7/74/Elce2017_new_GPIO_interface.pdf
https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/tree/Documentation/gpio/sysfs.txt
хороший пример
http://www.customelectronics.ru/rabota-s-portami-gpio-cubieboard/
Пример работы на питоне
http://docs.cubieboard.org/tutorials/common/using_python_program_control_gpios

Работа с использованием sysfs

//ls -L /sys/class/gpio/
Для наглядности подключим светодиод.



Пример работы в командной строке:
echo 226 > /sys/class/gpio/export

Использование как выход

echo out > /sys/class/gpio/gpio226/direction

Зажечь светодиод

echo 1 > /sys/class/gpio/gpio226/value

Погасить светодиод

echo 0 > /sys/class/gpio/gpio226/value

убрать порт из системы

echo 226 > unexport

Работа с использованием uapi Libgpiod

Libgpiod (**Lib** окон- чательно **G** бщая **P** ЗАДАЧА **I** Nput / **O** utput **d** evice) обеспечивает как API вызовы для использования в ваших собственных программах и следующие шесть приложений пользовательского режима для манипулирования GPIO линии:

- **gpiodetect** - перечисляет все присутствующие в системе gpiochips, их имена, метки и количество линий GPIO
- **gpioinfo** - перечисляет все строки указанных gpiochips, их имена, потребителей, направление, активное состояние и дополнительные флаги
- **gpioget** - читает значения указанных строк GPIO
- **gpioset** - устанавливает значения указанных линий GPIO, потенциально сохраняет экспортные линии и ожидает ожидания, ввода пользователя или сигнала
- **gpiofind** - находит имя gpiochip и смещение строки по имени строки
- **gpiomon** - дождаться событий на линиях GPIO, указать, какие события нужно просмотреть, сколько событий нужно обработать перед выходом или нужно ли сообщать о событиях на консоль

Установка инструмента **Libgpiod**

apt-get install libudev-dev autoconf-archive libtool

// https://github.com/brgl/libgpiod

wget https://git.kernel.org/pub/scm/libs/libgpiod/libgpiod.git/snapshot/libgpiod-1.1.2.tar.gz

tar -xzf libgpiod-1.1.2.tar.gz

cd libgpiod-1.1.2/

mkdir -p m4

./autogen.sh --enable-tools=yes --host=arm-linux-gnueabi

make

make install

ldconfig

Пример:

gpioset gpiochip0 226=1 //светодиод загорится

gpioset gpiochip0 226=0 //

Литература:

https://github.com/brgl/libgpiod

https://habr.com/ru/post/351512/

https://www.acmesystems.it/libgpiod

Информация

*ls -l /dev/gpi**

gpiodetect

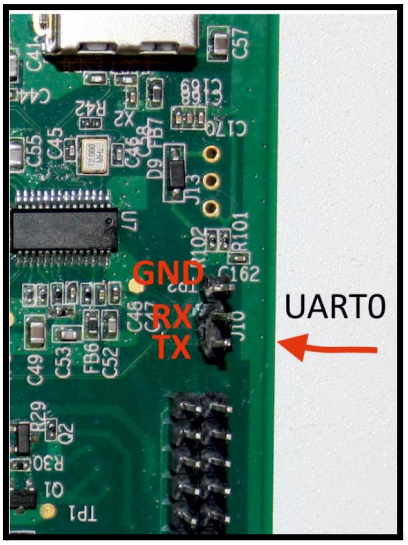
gpioinfo 1c20800.pinctrl

UART

UART0

Этот последовательный порт именуемый в системе как ttyS0 используется для отладки, с его помощью другой компьютер может подключиться к устройству. Это позволит увидеть логи загрузки операционной системы, войти в систему и управлять с помощью команд командной строки Linux, также настроить загрузчик.

Чтобы подключиться необходим последовательный порт с соответствующими логическими уровнями 3.3V. Обычный RS232 (COM Port) который присутствует почти на каждом ПК не подойдет, так как имеет уровни логических напряжений до +/-15v. Для этого понадобится либо преобразователь уровней (к примеру на специализированной микросхеме MAX232). Удобней всего воспользоваться готовым usb serial converter см. рис.



Расположение UART0 (ttyS0)

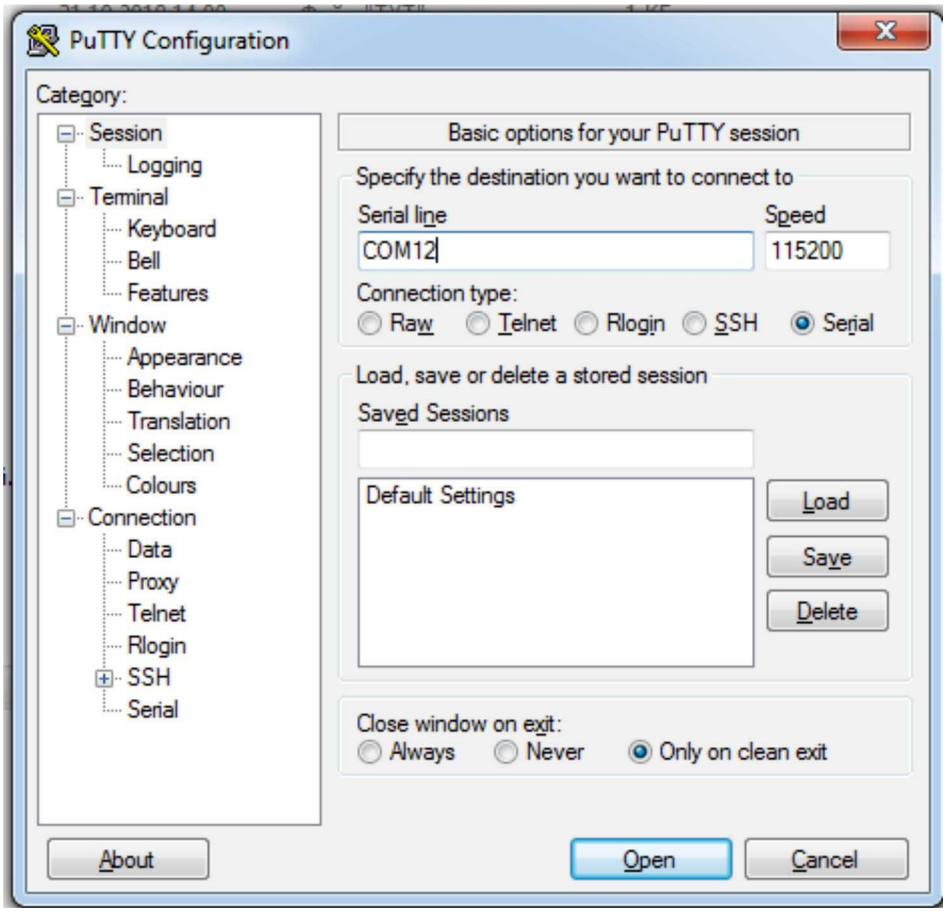
usb serial converter

Распиновка USB-serial шнурка:

Черный провод - GND
Зеленый провод - TX
Белый провод - RX
Красный провод - VCC (5V) (не используется)

Для работы понадобится клиентская программа для работы с терминалом. Если вы работаете под управлением ОС Windows, начиная с версии 7 исключена стандартная программа hyperterminal ,по этому требуется установить альтернативный клиент, например PuTTY.

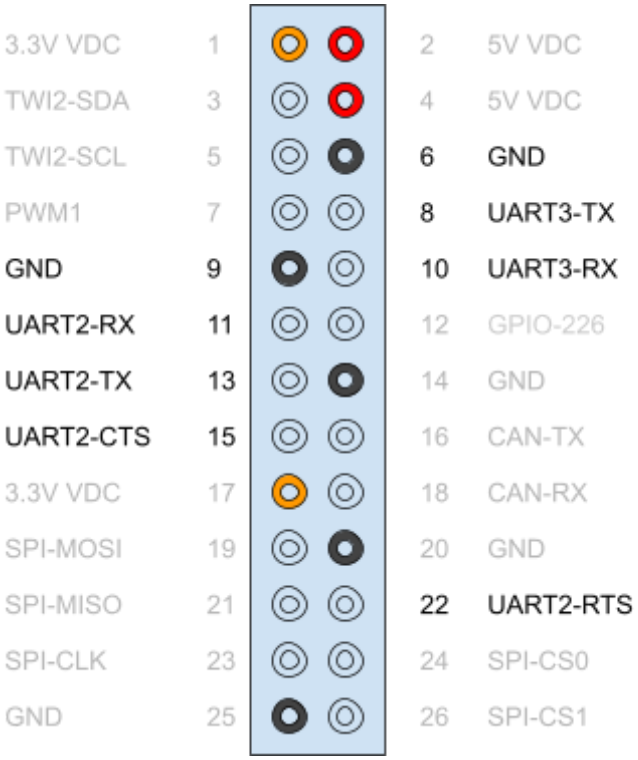
Запустите PuTTY и укажите скорость (по умолчанию 115200) и номер COM порта (в ОС Windows можно узнать в диспетчере устройств)



Наименование портов, назначение пинов:

UART2 (ttyS1)
uart_tx = Pin 13 (port PI18)
uart_rx = Pin 11 (port PI19)
uart_rts = Pin 22 (port PI16)
uart_cts = Pin 15 (port PI17)

UART3 (ttyS2)
uart_tx = Pin 8 (port PH00)
uart_rx = Pin 10 (port PH01)



В операционной системе Linux последовательные порты называется ttyS
Примечание:
Последовательный порт процессора uart 2 в системе именуется ttyS1
uart 3 именуется ttyS2
"tty" - сокращение от "телетайп", "S" означает последовательный порт.

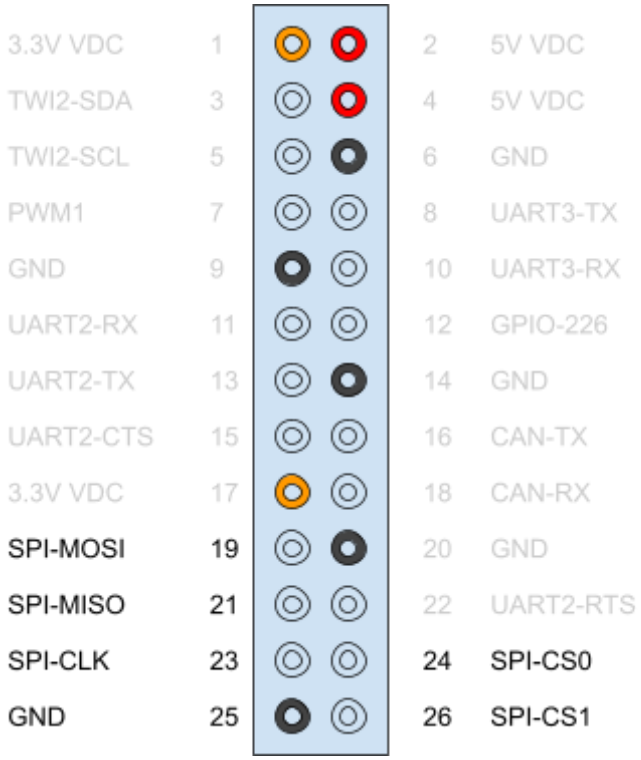
dmesg | grep tty
ttyS0 - консоль

Подключение GPS модуля

Нужно установить демон:
Преобразует NMEA-поток в удобный формат и раздает клиентским программам по TCP/IP
sudo apt-get install gpsd
настройка:
nano /etc/default/gpsd
 # Default settings for the gpsd init script and the hotplug wrapper.
 # Start the gpsd daemon automatically at boot time
START_DAEMON="true"
 # Use USB hotplugging to add new USB devices automatically to the daemon
USBAUTO="true"
 # Devices gpsd should collect to at boot time.
 # They need to be read/writeable, either by user gpsd or the group dialout.
DEVICES="/dev/ttyS1"
 # Other options you want to pass to gpsd
GPSD_OPTIONS="-n"

//Запустить:
//gpsd -b -N /dev/ttyS1
//gpsd /dev/ttyS1
проверка:
gpsmon
или так
gpscat -s 9600 /dev/ttyS1
появятся строчки с сообщениями от GSM модуля.
//карта, навигация
apt-get install foxtrotgps

SPI



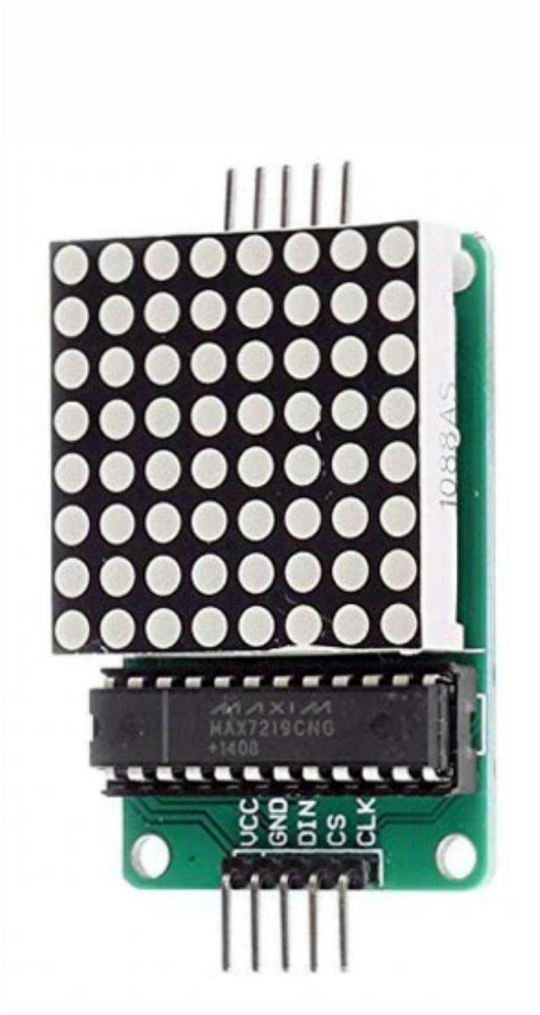
Наименование портов, назначение пинов:
SPI0
cs0 = port PI10
cs1 = port PI14
sclk = port PI11
mosi = port PI12
miso = port PI13

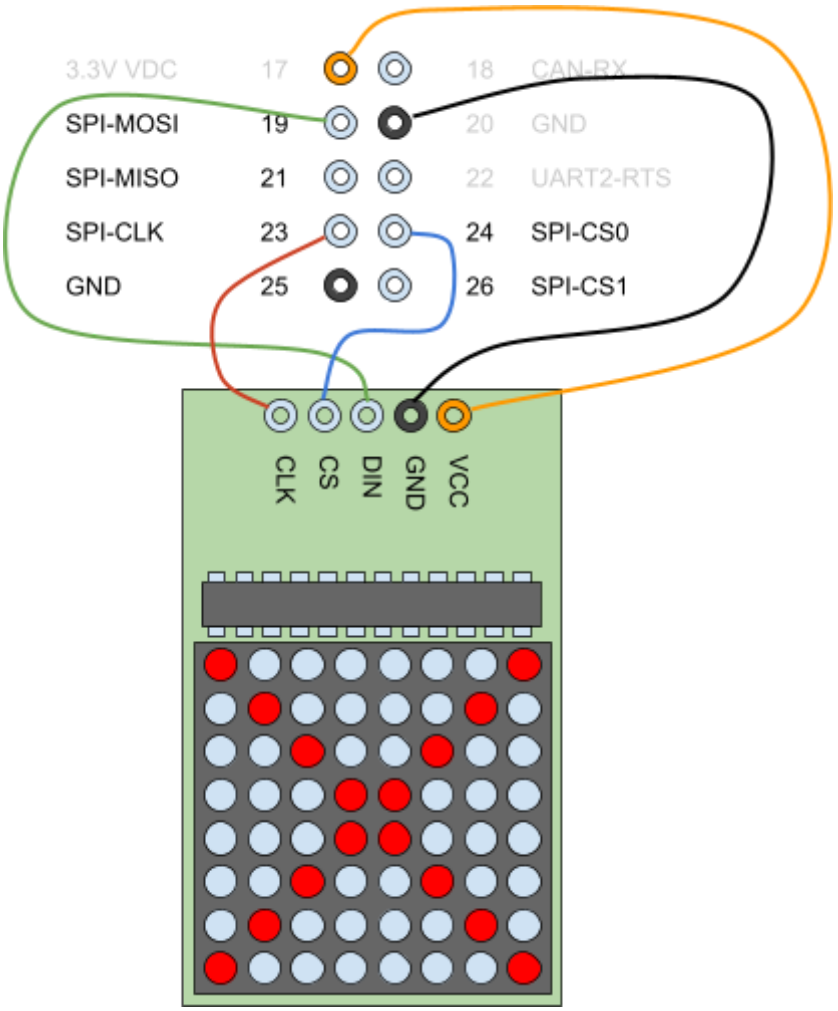
Проверка наличия в системе:
*ls /dev/spi**
/dev/spidev0.0

lsmod | grep -i spi
spidev 16384 0

*ls -l /dev/spi**
crw------- 1 root root 153, 0 янв 28 2018 /dev/spidev0.0

Подключение модуля светодиодной матрицы 8x8 на микросхеме MAX7219





Установка ПО:

apt-get install build-essential

apt-get install git

apt-get install python-dev python-pip libfreetype6-dev libjpeg-dev

sudo apt-get install python3-pip

*(**pip** - это система управления пакетами, которая используется для установки и управления программными пакетами, написанными на Python)*

//pip --version

//pip 9.0.1 from /usr/lib/python2.7/dist-packages (python 2.7)

//sudo -H pip install --upgrade luma.led_matrix

sudo -H pip3 install --upgrade luma.led_matrix

git clone https://github.com/rm-hull/max7219.git

cd max7219/

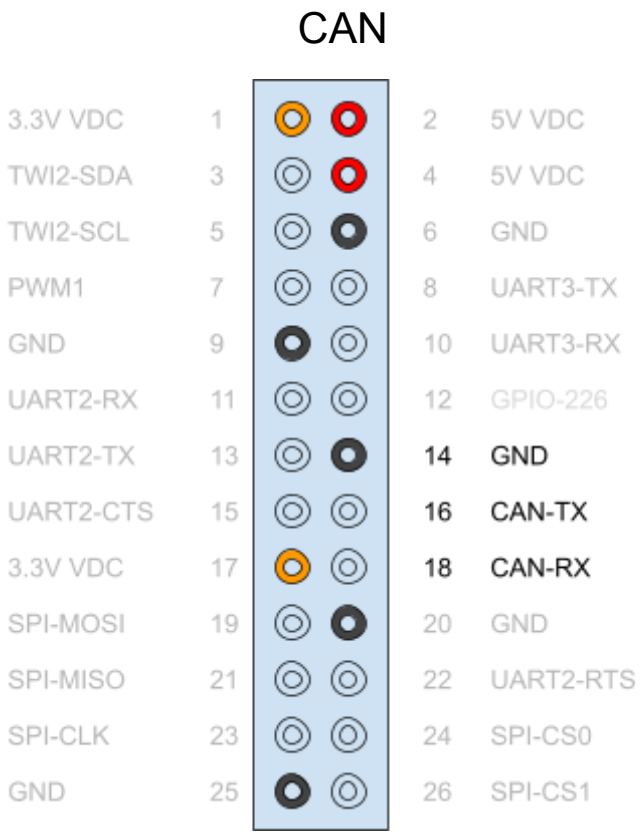
запуск теста

//python examples/matrix_demo.py

python3 examples/matrix_demo.py

Литература:

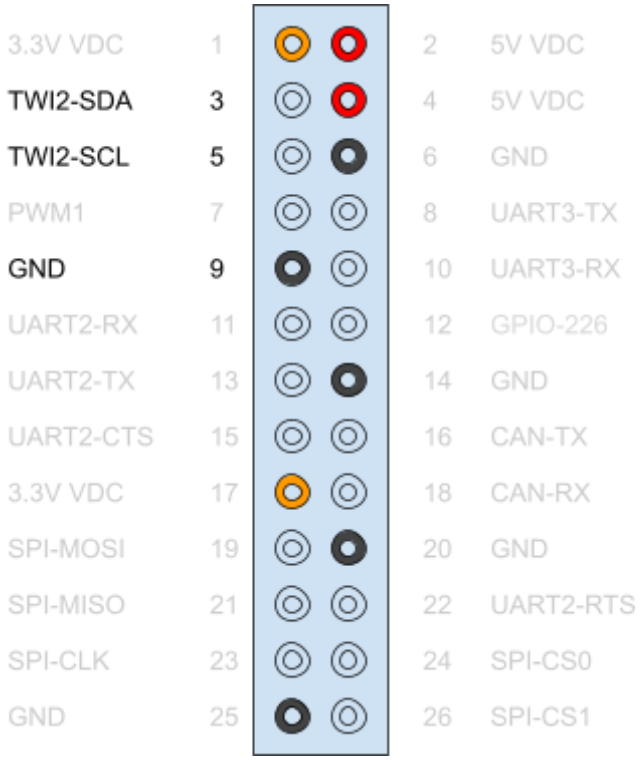
http://raspi.tv/2013/8-x-8-led-array-driven-by-max7219-on-the-raspberry-pi-via-python
https://luma-led-matrix.readthedocs.io/en/latest/install.html#installing-from-pypi



Наименование портов, назначение пинов:
can
can_tx Pin 16 (port PH20)
can_rx Pin 18 (port PH21)

В списке сетевых устройств можем увидеть наш сетевой интерфейс CAN:
ip a
3: can0: <NOARP,ECHO> mtu 16 qdisc noop state DOWN group default qlen 10
    link/can

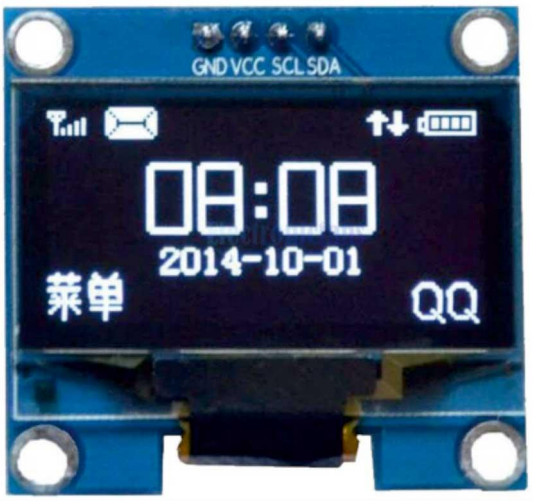
I2C



Наименование портов, назначение пинов:
twi2 (I2C)
twi2_sda Pin 3 (port PB21)
twi2_scl Pin 5 (port PB20)

Список доступных портов в системе:
*ls /dev/i2c**
/dev/i2c-0 /dev/i2c-1

Подключение OLED 1.3 дисплея с интерфейсом I2C



apt-get install i2c-tools libi2c-dev
i2cdetect -l

i2c-1	i2c	mv64xxx_i2c adapter	I2C adapter
i2c-0	i2c	mv64xxx_i2c adapter	I2C adapter

Просмотр наличия устройств на шине:
i2cdetect -y 1

```
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- 3c -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

в таблице есть устройство которое отвечает по адресу 3c это наш экран

Установим python
apt-get install python-pip python-dev build-essential
apt-get install python-setuptools
//для Python3, используйте easy_install3 и sudo apt-get install python3-setuptools
//sudo easy_install3 pip
//sudo apt install python3-pip

pip install --upgrade pip
//pip --version
//pip 9.0.1 from /usr/lib/python2.7/dist-packages (python 2.7)
//pip3 install --upgrade pip //если установленно 2 питона и нужный сам не обновляется
//apt-get install python3-dev python3-pip

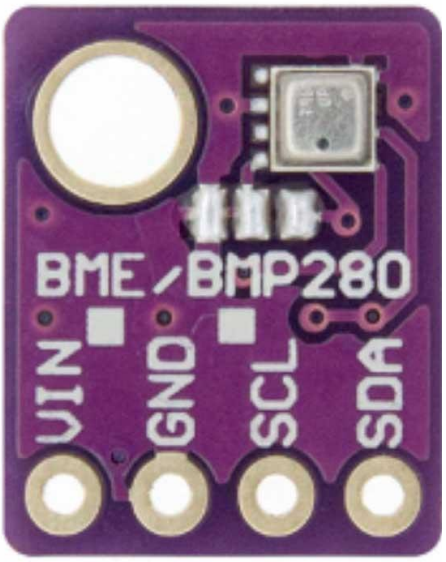
//sudo easy_install pip // ?
запустить easy_install с помощью полного пути:
python /usr/lib/python2.7/dist-packages/easy_install.py pip

pip install requests //pip3 install requests
//https://luma-oled.readthedocs.io/en/develop/install.html
Пример:
Luma.Examples
/// sudo usermod -a -G i2c,spi,gpio pi
sudo apt install python-dev python-pip libfreetype6-dev libjpeg-dev build-essential
sudo apt install libsdl-dev libportmidi-dev libsdl-ttf2.0-dev libsdl-mixer1.2-dev libsdl-image1.2-dev

git clone https://github.com/rm-hull/luma.examples.git
cd luma.examples
python3 -m pip install --upgrade pip //понадобилось обновить
sudo -H pip install -e . //точка в конце строки нужна!!!

Можно запустить пример:
python examples/clock.py --i2c-port 1 --width 128 --height 64 --display sh1106
//python3 examples/clock.py --i2c-port 1 --width 128 --height 64 --display sh1106

Подключение датчика окружающей среды BME280 с интерфейсом I2C



BME280 — датчик давления, температуры и влажности.

Подключаем датчик к соответствующим контактам мини ПК
В терминале вводим
i2cdetect -y 1

```
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- 76 --
```

Это означает что на шине есть устройство с адресом 76.

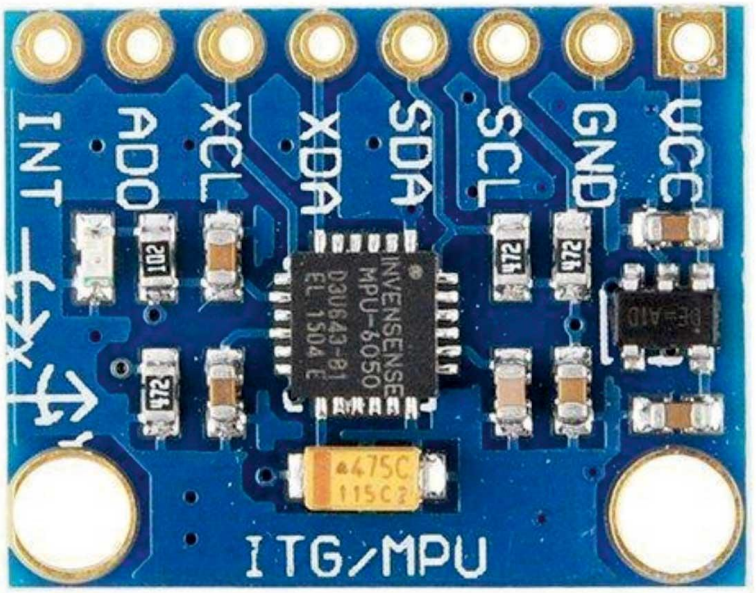
Пример на языке python

Скачаем и запустим исходный код примера работы с датчиком

wget -O bme280.py http://bit.ly/bme280py
или
wget https://bitbucket.org/MattHawkinsUK/rpispy-misc/raw/master/python/bme280.py
Запуск:
python bme280.py

```
Chip ID     : 96
Version     : 0
Temperature :  24.7 C
Pressure :  997.30178992 hPa
Humidity :  23.3212514326 %
```

Подключение 3-х осевой гироскоп и акселерометр MPU 6050 с интерфейсом I2C



Подключаем модуль и проверяем что он на шине.
i2cdetect -y 1

```
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Обнаружили датчик по адресу 68.

Пример на языке python

Создадим пустой файл

nano gyro.py

Вставим код

```
#!/usr/bin/python
import smbus
import math

# Register
power_mgmt_1 = 0x6b
power_mgmt_2 = 0x6c

def read_byte(reg):
    return bus.read_byte_data(address, reg)
```

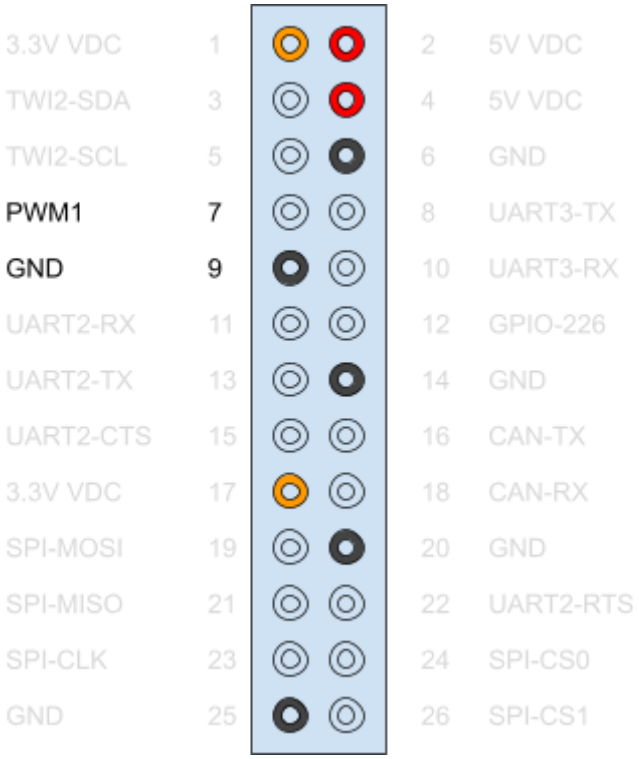
```
def read_word(reg):
    h = bus.read_byte_data(address, reg)
    l = bus.read_byte_data(address, reg+1)
    value = (h << 8) + l
    return value

def read_word_2c(reg):
    val = read_word(reg)
    if (val >= 0x8000):
        return -((65535 - val) + 1)
    else:
        return val

def dist(a,b):
    return math.sqrt((a*a)+(b*b))

def get_y_rotation(x,y,z):
    radians = math.atan2(x, dist(y,z))
    return -math.degrees(radians)

def get_x_rotation(x,y,z):
    radians = math.atan2(y, dist(x,z))
    return math.degrees(radians)

bus = smbus.SMBus(1) # bus = smbus.SMBus(0) fuer Revision 1
address = 0x68       # via i2cdetect

# Aktivieren, um das Modul ansprechen zu koennen
bus.write_byte_data(address, power_mgmt_1, 0)

print "Gyroskop"
print "--------"

gyroskop_xout = read_word_2c(0x43)
gyroskop_yout = read_word_2c(0x45)
gyroskop_zout = read_word_2c(0x47)

print "gyroskop_xout: ", ("%5d" % gyroskop_xout), " skaliert: ", (gyroskop_xout / 131)
print "gyroskop_yout: ", ("%5d" % gyroskop_yout), " skaliert: ", (gyroskop_yout / 131)
print "gyroskop_zout: ", ("%5d" % gyroskop_zout), " skaliert: ", (gyroskop_zout / 131)

print
print "Beschleunigungssensor"
print "---------------------"

beschleunigung_xout = read_word_2c(0x3b)
beschleunigung_yout = read_word_2c(0x3d)
beschleunigung_zout = read_word_2c(0x3f)

beschleunigung_xout_skaliert = beschleunigung_xout / 16384.0
beschleunigung_yout_skaliert = beschleunigung_yout / 16384.0
beschleunigung_zout_skaliert = beschleunigung_zout / 16384.0

print "beschleunigung_xout: ", ("%6d" % beschleunigung_xout), " skaliert: ", beschleunigung_xout_skaliert
print "beschleunigung_yout: ", ("%6d" % beschleunigung_yout), " skaliert: ", beschleunigung_yout_skaliert
print "beschleunigung_zout: ", ("%6d" % beschleunigung_zout), " skaliert: ", beschleunigung_zout_skaliert

print "X Rotation: " , get_x_rotation(beschleunigung_xout_skaliert, beschleunigung_yout_skaliert, beschleunigung_zout_skaliert)
print "Y Rotation: " , get_y_rotation(beschleunigung_xout_skaliert, beschleunigung_yout_skaliert, beschleunigung_zout_skaliert)
-------------------------------------
```

Запустить
python gyro.py

gyroskop_xout: -774 skaliert: -6
gyroskop_yout: 235 skaliert: 1
gyroskop_zout: 15 skaliert: 0

Beschleunigungssensor
----
beschleunigung_xout: -4088 skaliert: -0.24951171875
beschleunigung_yout: -3920 skaliert: -0.2392578125
beschleunigung_zout: 16932 skaliert: 1.03344726562
X Rotation: -12.6830009973
Y Rotation: 13.2362235306

PWM



Наименование порта, назначение пинов:
pwm - port PI03
узнать наличие в системе:
*ls /sys/class/pwm**
pwmchip0

Управляем светодиодом средством PWM

Пример управления светодиодом из консоли:

Экспорт канала ШИМ для управления пользователем.

cd /sys/class/pwm/pwmchip0/

//echo 0 > export

//у нас разведен PWM на порте процессора pwm1 и ему соответствует номер в дереве устройств 1

echo 1 > export

Выберите период сигнала ШИМ. Значение в наносекундах.

echo 10000000 > pwm1/period

Выберите рабочий цикл. Значение указывается в наносекундах и должно быть меньше периода.

echo 5000000 > pwm1/duty_cycle

Включить / отключить сигнал ШИМ.

echo 1 > pwm1/enable

Возможные значения:

- 1: включить
- 0: отключить

Измените полярность сигнала ШИМ. Полярность может быть изменена, только если ШИМ не включен.

echo "normal" > pwm1/polarity

Возможные значения:

- normal
- inversed

Для примера по мигаем светодиодом

cd /sys/class/pwm/pwmchip0/

echo 1 > export

echo 100000000 > pwm1/period

echo 30000000 > pwm1/duty_cycle

echo 1 > pwm1/enable

IR Приемник

pin 36 (PB4): 1c21800.ir (GPIO UNCLAIMED) function ir0 group PB4