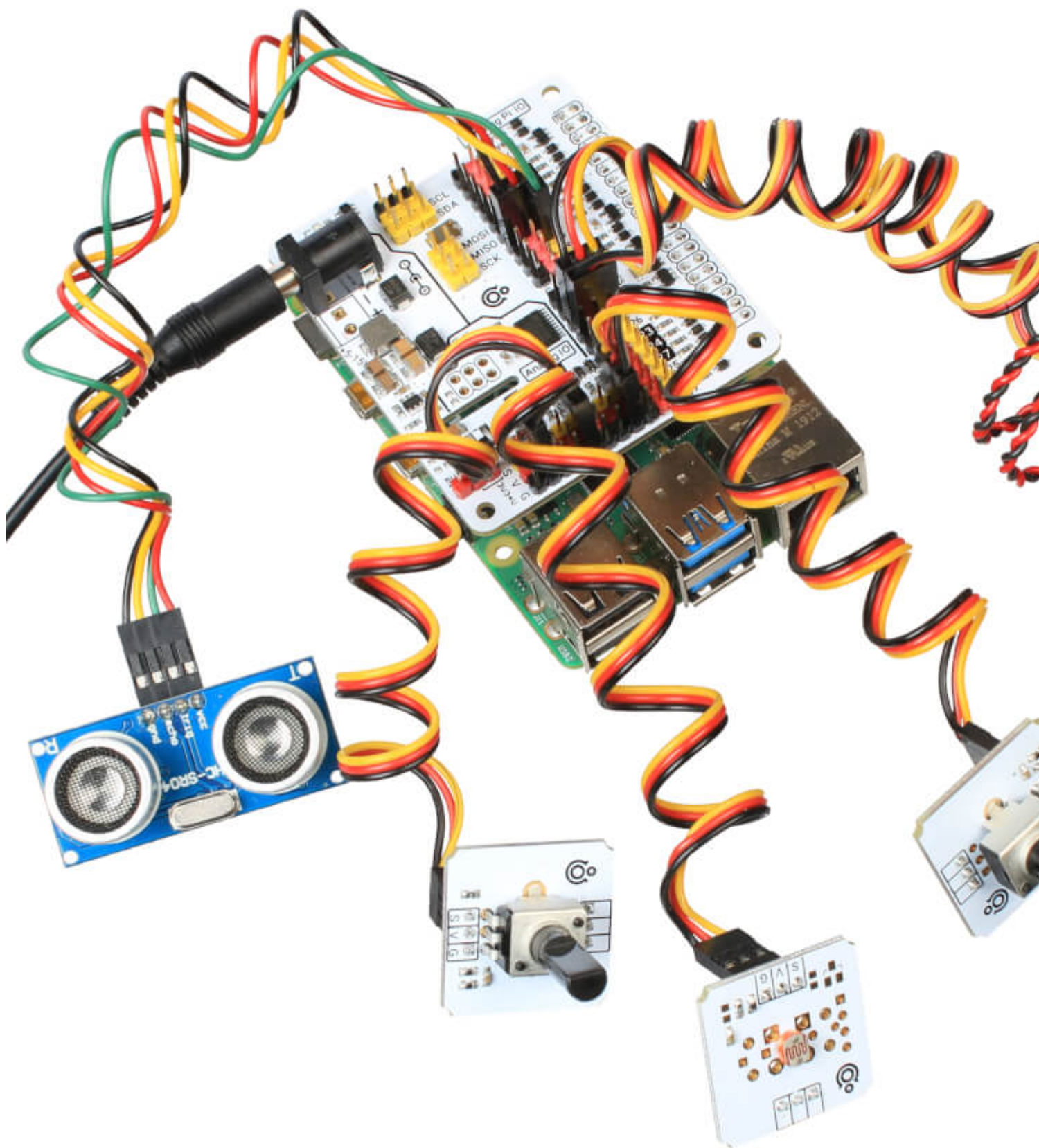


Troyka HAT: подключение, настройка и начало работы

Плата расширения [Troyka HAT](#) поможет подключить модули форм-фактора [Troyka](#) к одноплатникам [Raspberry Pi](#) без дополнительных проводов и макетных плат.

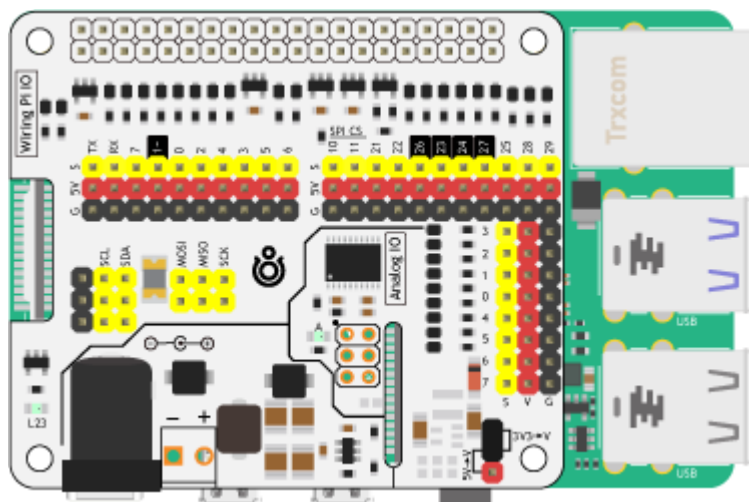


Платформа Тройка HAT также расширит возможности малинки, ведь у неё на борту расположен I²C-расширитель портов на микроконтроллере STM32F030F4P6 с вычислительным ядром ARM Cortex M0. Контроллер даёт восемь дополнительных пинов ввода-вывода с аппаратной поддержкой 12-битного АЦП и 16-битного ШИМ.

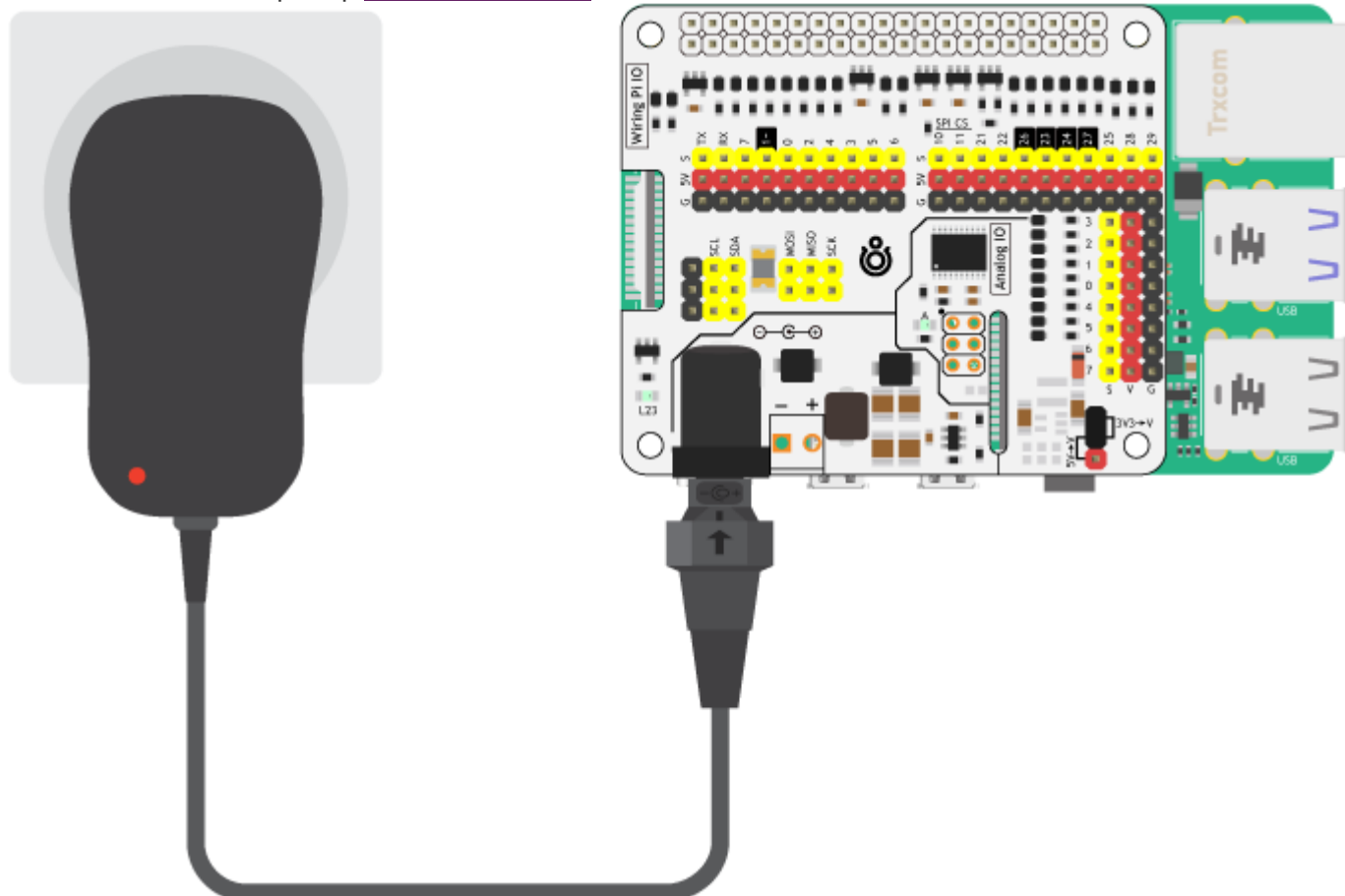
Подключение и настройка

Для старта необходимо [подключить железо](#) и [выполнить программную настройку](#).

Аппаратная часть



1. Установите Тройка HAT сверху на Raspberry Pi.
2. Подключите питание к DC-разъёму на Тройка HAT. Используйте блок питания с выходным напряжением от 7 до 12 вольт, например [Robiton TN2000S](#).

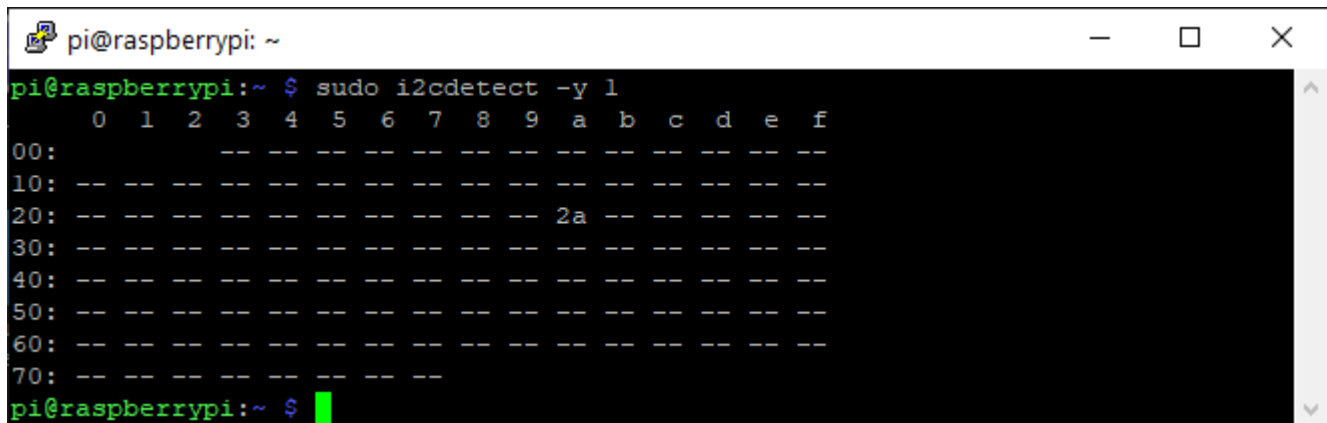


Программная часть

1. [Заведите Raspberry Pi](#).

2. [Включите шину I²C](#) и [установите I²C сканер](#)
3. Просмотрите подключённые I²C-устройства:

```
sudo i2cdetect -y 1
```



```
pi@raspberrypi: ~  
pi@raspberrypi:~$ sudo i2cdetect -y 1  
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
20: -- -- -- -- -- -- -- -- 2a -- -- -- -- -- --  
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
pi@raspberrypi:~$
```

Сканер должен найти адрес 0x2a, который является I²C-адресом расширителя портов по умолчанию.

4. Установите библиотеку [WiringPi](#) для работы с контактами Raspberry Pi:

```
sudo pip3 install wiringpi
```

5. Установите библиотеку [TroykaCapPython](#) для работы с контактами расширителя портов:

```
sudo pip3 install git+https://github.com/amperka/TroykaCapPython
```

На этом установка закончена, теперь смело переходите к экспериментам.

Примеры работы

Имена пинов на Troyka HAT относятся к нумерации [Wiring Pi](#), которая отличается от стандартной нумерации [BCM](#), как например в стартовом наборе [Малина](#). Для удобства в примерах ниже будем использовать именно нумерацию [Wiring Pi](#). А все подробности распиновки вы можете [найти в соответствующем разделе](#).

Маячок

Повторим третий эксперимент из набора [Малина](#) — маячок. Только вместо облачка, подключите к Troyka HAT [светодиод «Пиранья» \(Troyka-модуль\)](#) через стандартный трёхпроводной шлейф к 7-цифровому пину.

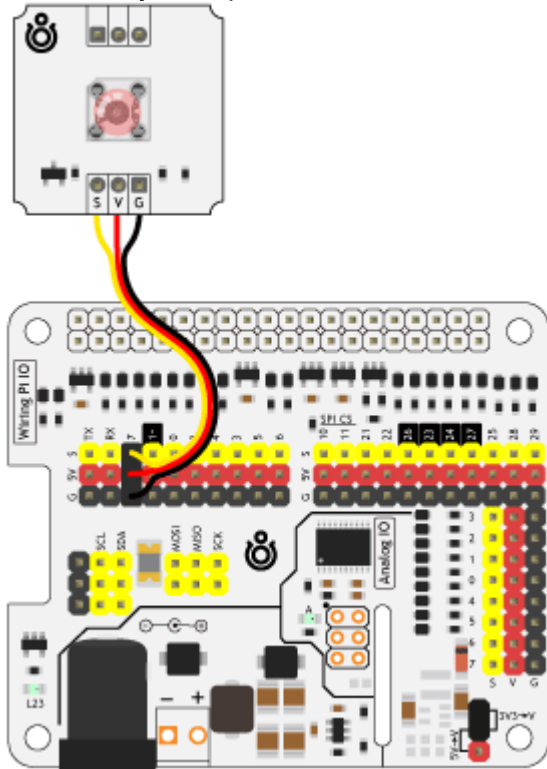
Код для Python

```
raspberrypi-troyka-hat-example-blink.py  
# Библиотека для работы с методами языка Wiring (Arduino)  
import wiringpi as wp  
# Инициализация WiringPi  
wp.wiringPiSetup()  
# Пин 7 в режим выхода  
wp.pinMode(7, wp.OUTPUT)  
  
while (True):  
    # Подаём на пин 7 высокий уровень  
    wp.digitalWrite(7, wp.HIGH)  
    # Ждём пол секунды  
    wp.delay(500)  
    # Подаём на пин 7 низкий уровень
```

```
wp.digitalWrite(7, wp.LOW)

# Ждём пол секунды
wp.delay(500)
```

После запуска скрипта, светодиод начнёт мигать раз в пол секунды.



Кнопочный выключатель

А теперь попробуем собрать простой кнопочный выключатель. Подключите к Тройка НАТ [светодиод «Пирания» \(Тройка-модуль\)](#) к 7 цифровому пину, а [кнопку \(Тройка-модуль\)](#) к 22 цифровому пину.

Код для Python

[raspberrypi-troyka-hat-example-digital-button-led.py](#)

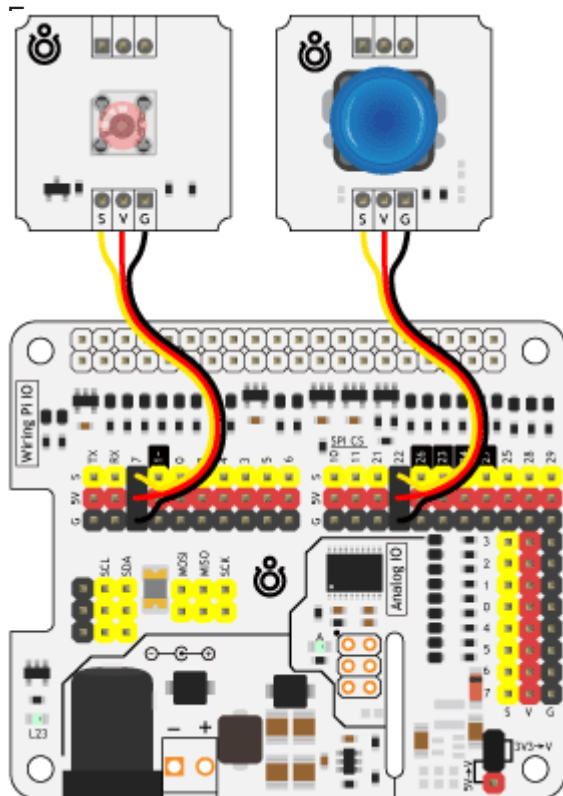
```
# Библиотека для работы с методами языка Wiring (Arduino)
import wiringpi as wp

# Инициализация WiringPi
wp.wiringPiSetup()

# Пин 22 в режим входа
wp.pinMode(22, wp.INPUT)

# Пин 7 в режим выхода
wp.pinMode(7, wp.OUTPUT)

while (True):
    # Если кнопка нажата
    if (not(wp.digitalRead(22))):
        # Включаем светодиод
        wp.digitalWrite(7, wp.HIGH)
    # Если кнопка отжата
    else:
        # выключаем светодиод
        wp.digitalWrite(7, wp.LOW)
```



я, а если отжата — погаснет.

Светильник с управляемой яркостью

В продолжении соберем светильник с управляемой яркостью. В Raspberry Pi всего два канала ШИМ, а АЦП вовсе отсутствует, поэтому задействуем контакты I²C расширителя, который поможет вести коммуникацию с аналоговыми датчиками. В качестве примера подключим [ползунковый потенциометр \(Тройка-модуль\)](#) и [Светодиод 5 мм \(Тройка-модуль\)](#) к пинам расширителя 3 и 6 на Тройка HAT соответственно.

Код для Python

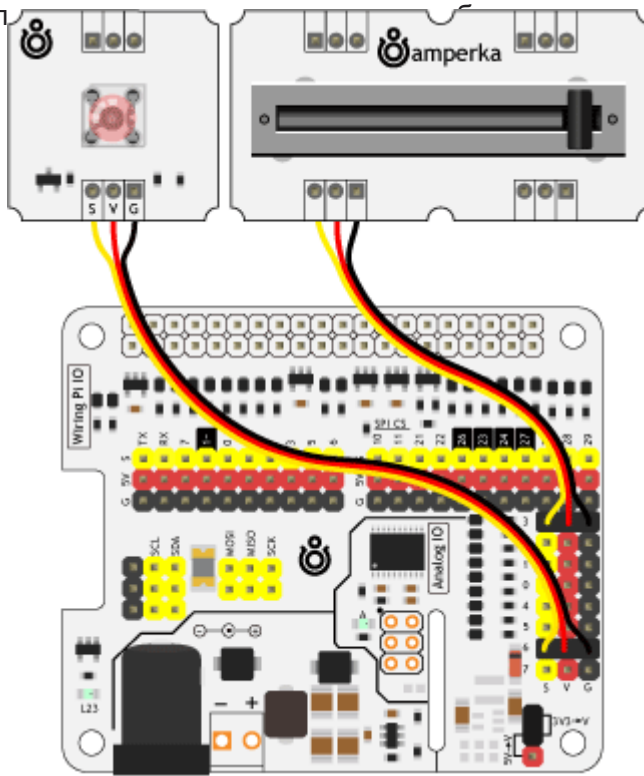
[rasberry-pi-troyka-hat-example-analog-pot-led.py](#)

```
# Библиотека для работы с расширителем портов
import gpioexp

# Создаём объект для работы с расширителем портов
exp = gpioexp.gpioexp()

while True:
    # Считываем состояние потенциометра
    potValue = exp.analogRead(3)
    # Включаем яркость светодиода
    # в зависимости от состояние потенциометра
    exp.analogWrite(6, potValue)
```

После зап. я в зависимости от перемещения ползунка



слайдера.

Аналоговая регулировка громкости

Выйдем за рамки светодиодной робототехники. Сделаем аналоговую регулировку громкости одноплатника Raspberry Pi.

1. Подключите [ползунковый потенциометр](#) к пину 7 I²C расширителя на плате Тройка HAT.
2. Установите библиотеку `pyalsaaudio` для контроля уровня громкости:

```
pip3 install pyalsaaudio
```

3. Запустите нижеприведённый скрипт.

Код для Python

[raspberry-pi-troyka-hat-example-analog-control-volume.py](#)

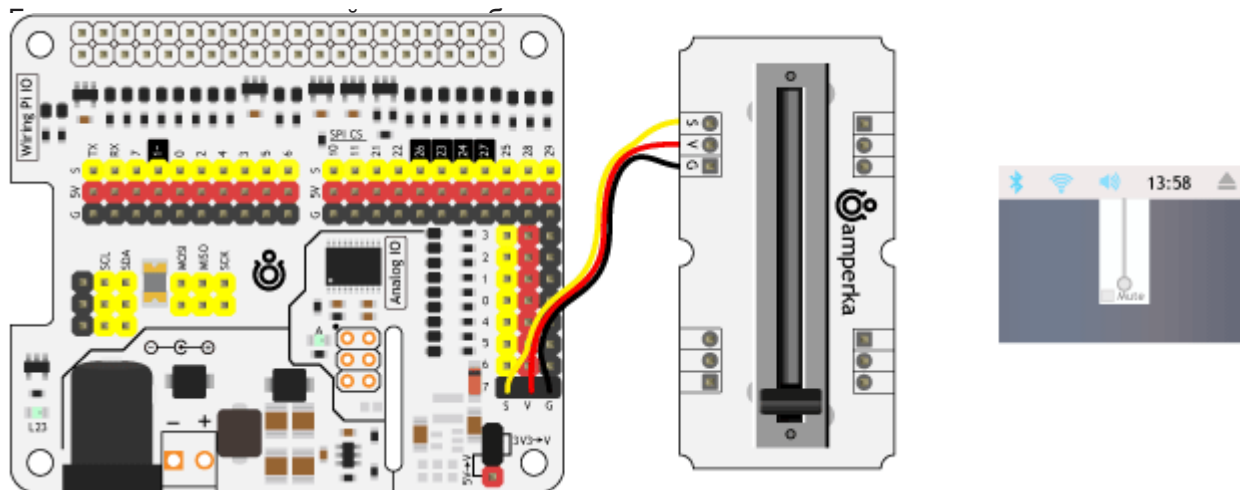
```
# Библиотека для работы с расширителем портов
import gpioexp

# Библиотека для контроля уровня громкости
import alsaaudio

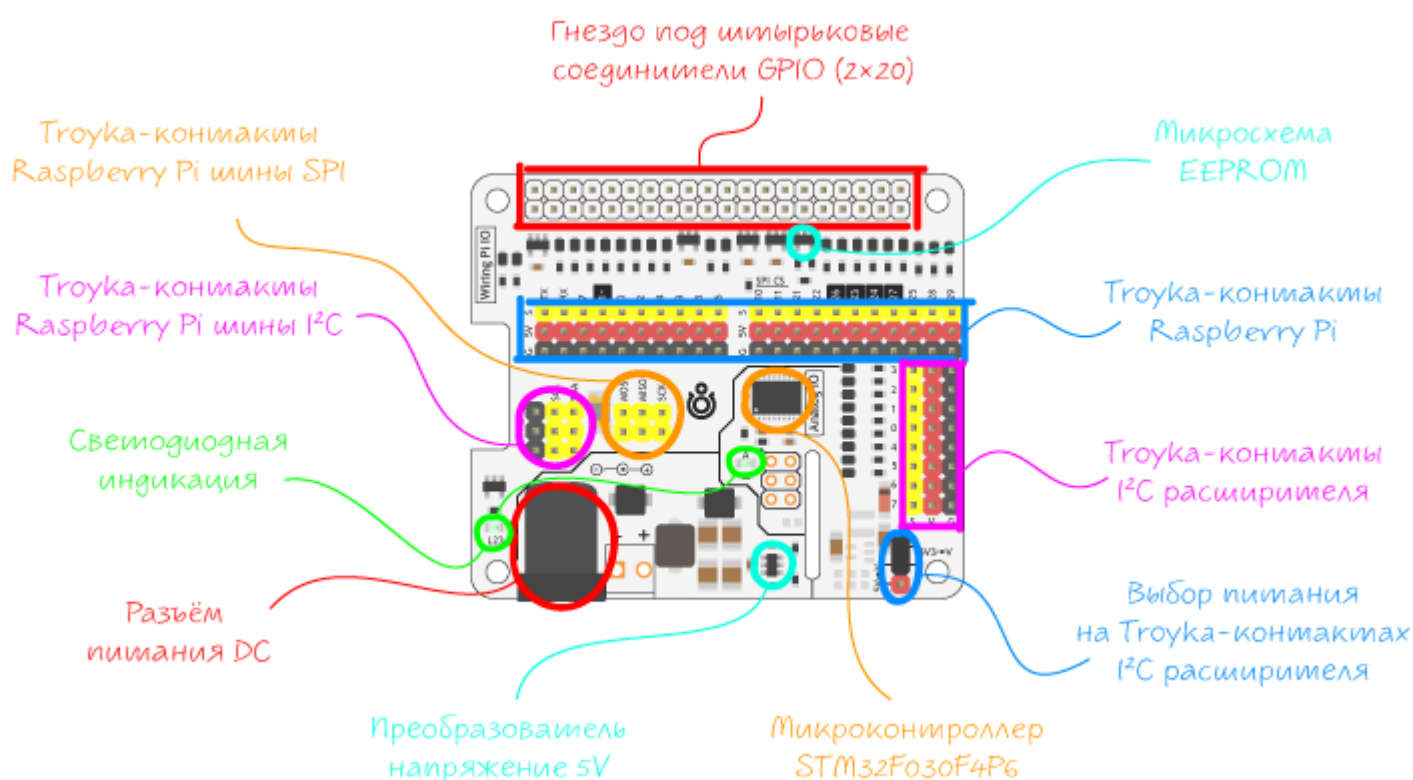
# Создаём объект для работы с расширителем портов
exp = gpioexp.gpioexp()

# Создаём объект для работы с регулировкой громкости
mixer = alsaaudio.Mixer('Master')

while True:
    # Считываем состояние слайдера
    potValue = exp.analogRead(7)
    # Умножаем полученное значение со слайдера на 100
    # и преобразовываем переменную float в int
    volume = int(potValue * 100)
    # Устанавливаем громкость
    mixer.setvolume(volume)
```

Элементы платы



Микроконтроллер STM32F030F4P6

Raspberry Pi обладает всего двумя каналами ШИМ и не имеет аналого-цифрового преобразователя. Тройка HAT решает проблему аналоговых интерфейсов через встроенный [микроконтроллер STM32F030F4P6](#) на архитектуре ARM Cortex M0+, который получает команды от малинки по шине I²C и дальнейшие операции по железу берет на себя. Ножки микроконтроллера выведены на соответствующую колодку [Тройка-контактов I²C расширителя](#).

Тройка-контакты Raspberry Pi

На плате установлены группы штыревых Тройка-контактов «S-5V-G»:

- сигнал (S) — цифровой пин Raspberry Pi с толлерантностью к 5 В.
- питание (5V) — питание модуля, постоянно и равно 5 В
- земля (G) — общая земля

Отдельными пинами на плате выведены три группы контактов аппаратного интерфейса I²C и две группы SPI.

Колодки позволяют быстро подключать типовые [Тройка-модули](#) к малинке без дополнительных проводов и макетных плат. Подробности про сигналы и интерфейсы Тройка HAT читайте в разделе [распиновка](#).

Тройка-контакты I²C расширителя

Пины [I²C расширителя](#) выведены на отдельную группу Тройка-контактов «S-V-G»:

- Сигнал (S) — сигнальный пин расширителя GPIO
- Питание (V) — питание модуля 3,3 / 5 В ([выбирается джампером](#))
- Земля (G) — общая земля

Колодки позволяют быстро подключать типовые [Тройка-модули](#) к малинке без дополнительных проводов и макетных плат. Подробности про сигналы и интерфейсы Тройка HAT читайте в разделе [распиновка](#).

Выбор питания на Тройка-контактах I²C расширителя

На линии питания I²C-расширителя есть возможность выбора питания установкой джампера:

- **5V→V** — на линии [V](#) будет присутствовать напряжение 5 вольт. Режим пригодится при подключении модулей с рабочим напряжением 5 вольт, например [микросервопривод FS90](#) или [ультразвуковой дальномер SR-04](#).
- **3V3→V** — на линии [V](#) — 3,3 вольта. Т.к. микроконтроллер ожидает принимать диапазон входного напряжения от 0 до 3,3 вольт, то режим полезен при подключении аналоговых сенсоров, которые возвращают напряжение от 0 до значения на линии питания, например [модуль потенциометр](#) или [датчик освещённости](#).

Светодиодная индикация

Имя светодиода	Назначение
L23	Светодиод контакта 23 по нумерации WiringPi. При подачи высокого уровня — светодиод загорается, при низком — гаснет.
ACT	Мигает при обмене данными между одноплатником Raspberry Pi и I ² C расширителем портов. Если светодиод не мигает, скорее всего микроконтроллер не прошит.

Разъём питания DC

Коннектор DC Barrel Jack служит для подключения внешнего источника напряжения в диапазоне от 7 до 15 вольт, который будет питать одновременно Тройка HAT и подключенную Raspberry Pi. Если запитать устройство от порта Raspberry Pi, то напряжение на [Тройка-контактах Raspberry Pi](#) будет отсутствовать.

Преобразователь напряжения

Понижающий импульсный преобразователь [TPS565208](#) обеспечивает питание Тройка HAT и подключенной Raspberry Pi при напряжения через [разъём питания DC](#). Диапазон входного напряжения от 7 до 15 вольт. Выходное напряжение 5 В с максимальным выходным током 5 А.

Микросхема ID EEPROM

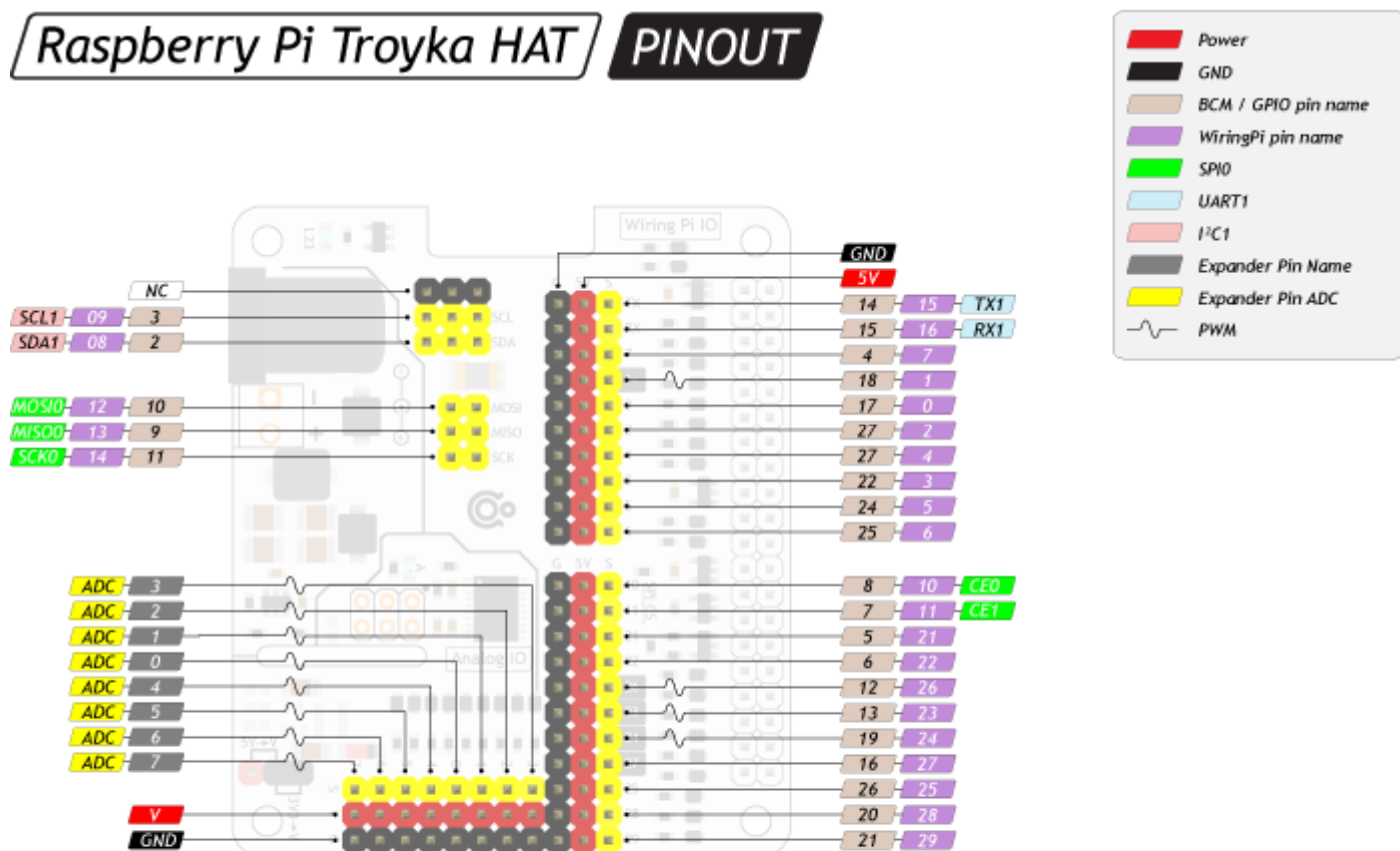
На плате Troykaa HAT расположена микросхема EEPROM-памяти [AT24C32D](#), которая содержит информацию о производителе платы, настройке GPIO и описание подключенного оборудования, которое позволяет Linux автоматически загружать необходимые драйверы.

Микросхема ID EEPROM одно из основных требований компании [Raspberry Pi Foundation](#) в спецификации производств плат расширения с именем [Raspberry Pi HATs](#).

Входные пины ввода-вывода

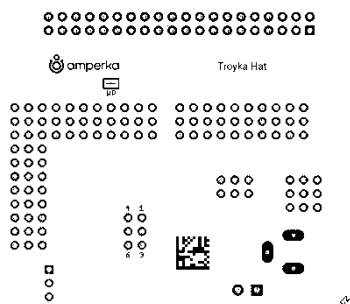
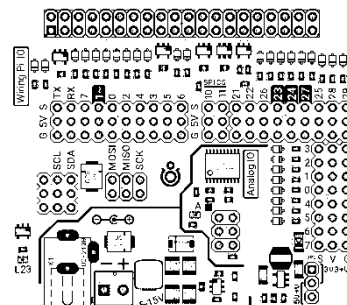
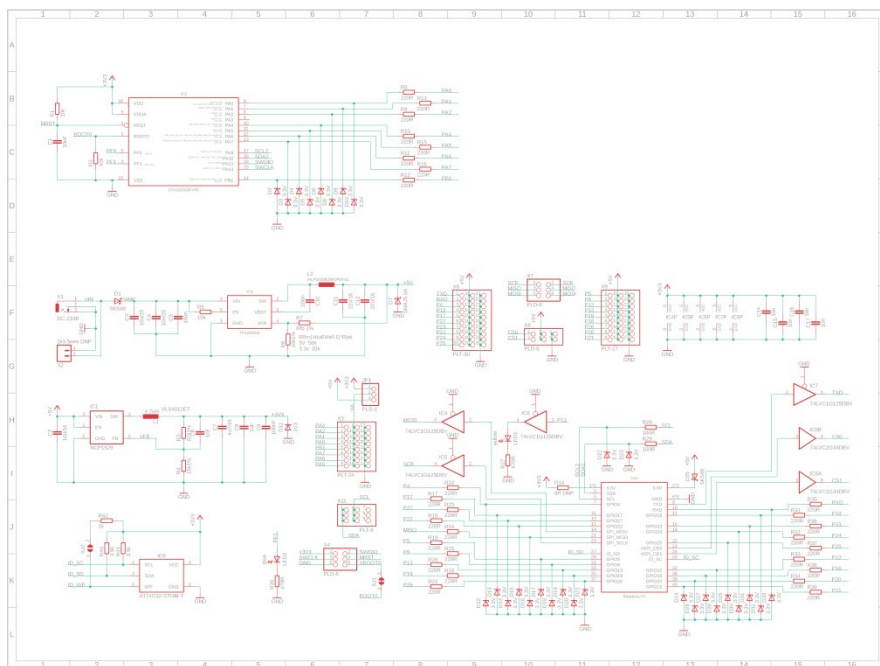
Troyka HAT подключается к одноплатникам Raspberry Pi сверху через входной 40-пиновый разъём.

Распиновка

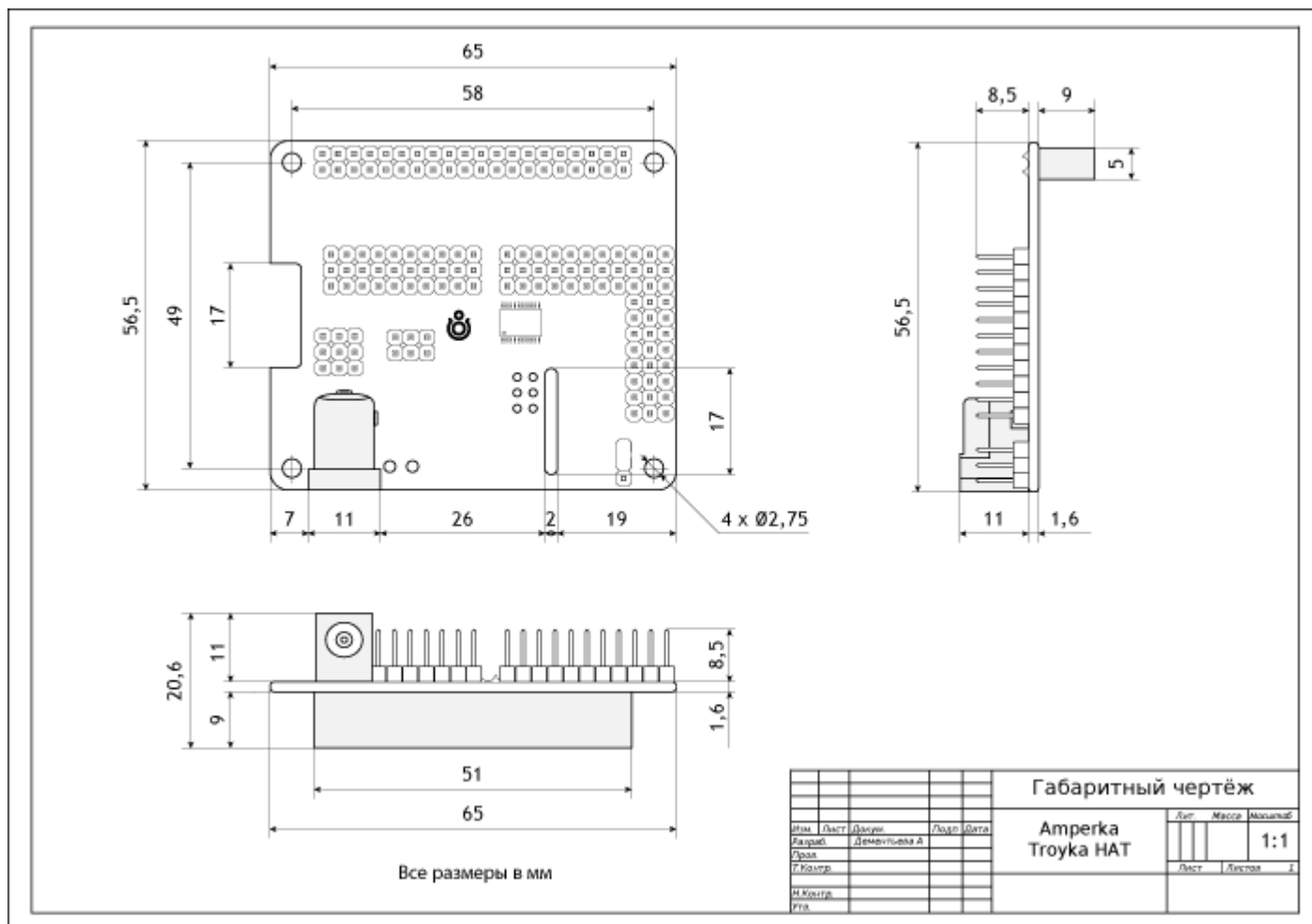


All pins 5V tollerant

Принципиальная и монтажная схемы



Габаритный чертёж



Характеристики

- Модель: Amperka Troyka HAT
- Совместимость: компьютеры Raspberry Pi
- Аппаратный интерфейс: штырьковые соединители GPIO
- Программный интерфейс: контакты Raspberry Pi и I²C-расширитель
- Входное напряжение через DC-разъём: 7–15 В
- Размеры: 65×56,5×20,5 мм

Контакты Raspberry Pi

- Тройка-контактов ввода-вывода общего назначения: 21
- Тройка-контактов SPI: 3
- Тройка-контактов I²C: 2
- Напряжение питания периферии: 5 В
- Напряжение логических уровней: 3,3 В (все пины толерантны к 5 В)

Контакты I²C расширителя

- Микроконтроллер: STM32F030F4P6
- I²C-адрес: по умолчанию 0x2A
- Тройка-контактов ввода-вывода общего назначения: 8
- Контакт с поддержкой ШИМ: 8
- Разрядность ШИМ: 8 бит
- Контакт с АЦП: 8
- Разрядность АЦП: 12 бит

- Напряжение питания периферии: 3,3 / 5 В (выбирается джампером)
- Напряжение логических уровней: 3,3 В (толерантность к 5 В)