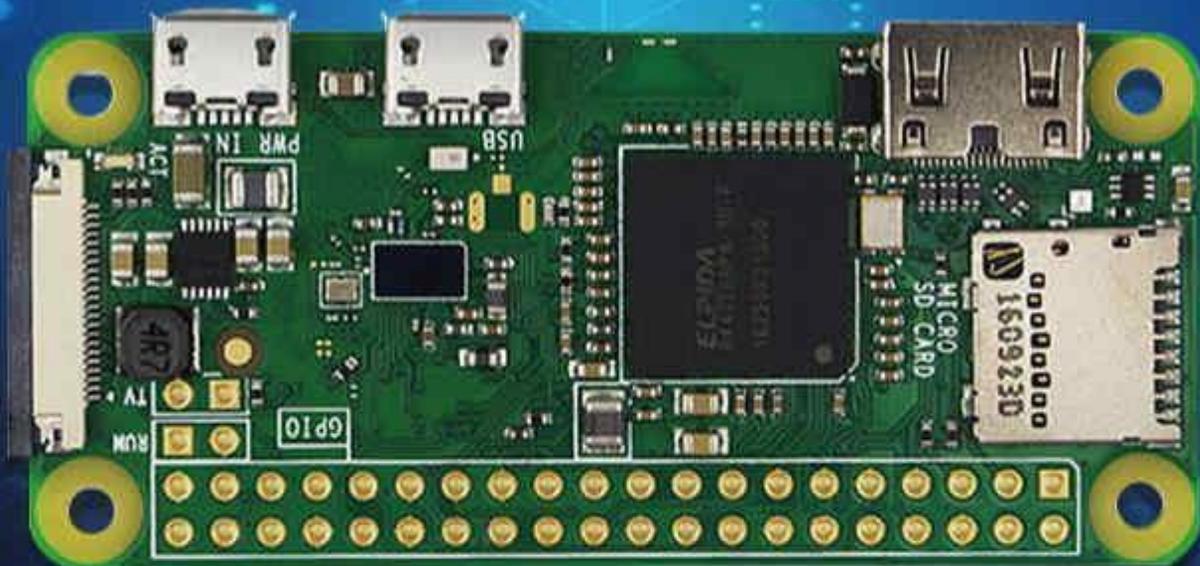


# ZERO W/ZERO WH



Опираясь на опыт сообщества

## Беспроводные проекты на Raspberry Pi Zero W

Станьте мобильнее с самым маленьким в мире Raspberry Pi Zero W

Василис Циварас

# **Беспроводные проекты на Raspberry Pi Zero W**

Станьте мобильнее с самым маленьким в  
мире Raspberry Pi Zero W

**Василис Циварас**

# Об авторе

Василис Циварасис, инженер по информатике, живет в Греции. Он является автором книги «Создание квадрокоптера с помощью Arduino», а также председателем студенческого отделения Университета IEEE в Янине. В настоящее время он работает над проектами, связанными с робототехникой, домашней автоматикой и интеллектуальными системами безопасности. Он также является энтузиастом технологий Интернета вещей и дронов.

# Оглавление

<b>Предисловие</b>	1
<b>Глава 1: Введение в Raspberry Pi Zero W</b>	6
<b>Семейство Raspberry Pi</b>	6
<b>Raspberry Pi Zero W</b>	11
<b>Характеристики</b>	12
<b>Поддержка камеры</b>	15
<b>Аксессуары</b>	15
OTG-кабель	16
PowerHub	17
Разъем GPIO	18
Карта microSD и адаптер карты	19
Кабель HDMI-mini HDMI	19
Кабель HDMI-VGA	20
Разъемы RCA	21
<b>Корпус Raspberry Pi Zero W</b>	21
<b>Распределения</b>	22
Дистрибутив NOOBS	23
Дистрибутив Raspbian	24
<b>Дистрибьюторы</b>	26
<b>Общие проблемы</b>	27
Шаги отладки	27
Проблема с картой microSD	28
Защита корпуса	28
<b>Резюме</b>	28
<b>Глава 2: Интернет вещей и сети</b>	29
<b>Интернет вещей</b>	29
<b>Основные протоколы связи</b>	30
Интернет-протоколы	31
IoT Protocols	31
Bluetooth	32
Zigbee	32
Z-Wave	32
6LowPAN	33
NFC	33
Wi-Fi	33

LoRaWAN	33
<b>Подключение вашего Pi</b>	34
Настройка Pi Zero W	34
Интернет	34
Локальная сеть	35
Перенаправление портов	37
Подключение к интернету	37
Сетевое администрирование	38
Подключиться для выполнения команды	39
Автоматическая аутентификация	40
<b>Подключение для передачи файлов</b>	41
FileZilla	41
SCP	43
<b>Защита Raspberry Pi</b>	43
Брандмауэр	44
<b>Резюме</b>	45
<b>Глава 3: Чат-бот</b>	46
<b>Пользовательский ввод</b>	46
Установка сервера	47
Разработка чат-бота	51
<b>Услуги чата</b>	54
Чатфуэль	54
Frow XO	55
Converse	55
Мессенджер Facebook	56
Умный робот Google	57
<b>Безопасная передача данных</b>	58
<b>Резюме</b>	58
<b>Глава 4: Мобильный робот</b>	59
<b>Основы робототехники</b>	59
Двигатели постоянного тока	61
Крутящий момент	62
Колесо	63
Энкодеры	64
<b>Обзор компонентов</b>	66
Мотор DC и колеса	69
Энкодер	70
Arduino	71

Моторшилд	73
Сервопривод	74
Ультразвуковой датчик	75
Bluetooth	78
Макетная плата	80
Батарея	81
Другие компоненты	82
<b>Пайка двигателя</b>	82
<b>Программирование контроллера</b>	84
Основная концепция	85
Разработка контроллера	85
<b>Идеи будущего</b>	88
Четырехмоторные мобильные роботы	89
<b>Резюме</b>	93
<b>Глава 5: Домашний бот</b>	94
<hr/>	
<b>Знакомство с домашними ботами</b>	94
<b>Программирование сокетов</b>	95
Простое взаимодействие клиент-сервер	96
Простой сервер	97
Простой клиент	100
Расширенная связь клиент-сервер	102
Расширенный сервер	102
Расширенный клиент	107
<b>Домашняя автоматика</b>	108
<b>Резюме</b>	111
<b>Глава 6: Камера наблюдения</b>	112
<hr/>	
<b>Установка камеры</b>	112
<b>Установка MotionPie</b>	115
Настройка нескольких сетевых камер	116
Подключение камеры за пределами вашей локальной сети	116
<b>Настройка параметров MotionPie</b>	117
Общие настройки	118
Беспроводная сеть	119
Видеоустройство	119
Файловое хранилище	121
Наложение текста	121
Потоковое видео	121
Неподвижные изображения	123
Обнаружение движения	123

Кинофильмы	123
Уведомления о движении	124
График работы	124
<b>Установка OpenCV</b>	125
<b>Распознавание лиц</b>	126
<b>Резюме</b>	131
<b>Глава 7: Портативные динамики</b>	<b>132</b>
<hr/>	
<b>Market speakers</b>	132
<b>Настройка программы</b>	134
<b>Networking</b>	136
<b>Музыкальная шкатулка</b>	136
<b>Аудио</b>	137
<b>Музыкальные файлы</b>	137
Музыкальные онлайн-сервисы	138
Безопасность	138
<b>Звуковые проигрыватели</b>	138
Альсамиксер	139
<b>Подключение динамика</b>	140
<b>Музыка на Pi</b>	141
<b>Выбор настроения</b>	143
Аппаратные средства	144
Code	145
<b>Резюме</b>	146
<b>Глава 8: Хостинг WebPi</b>	<b>147</b>
<hr/>	
<b>Web хостинг</b>	147
Определение	147
Услуги хостинга	149
Доменное имя	150
DNS	151
Настройка	151
<b>Взаимодействие клиент-сервер</b>	152
Запрос клиента	152
Протоколы связи	153
Ответ сервера	154
Прокси-серверы и кеширование	155
<b>Разработка сайта</b>	156
ХАМРР/LАМРР	157
Доступ к веб-сайту	158

<b>пду</b>	158
<b>SSH</b>	159
SSH-клиенты	159
Windows	159
Linux	163
Автоматизация процедуры SSH	164
<b>SFTP</b>	165
FileZilla	166
Терминал	167
<b>Networking</b>	168
Доступ к роутеру	169
Перенаправление портов	170
Домашний IP-адрес	171
Слои безопасности	172
Безопасность	173
Интерфейс роутера	173
Компьютер	175
ISP	175
<b>Резюме</b>	175
<b>Глава 9: AlexaPi</b>	176
<b>Создание аккаунта разработчика Amazon</b>	176
<b>Настройка Raspberry Pi</b>	184
<b>Установка Alexa</b>	185
<b>Распознавание голоса</b>	186
<b>Официальная Alexa против AlexaPi</b>	187
<b>Сетевое администрирование</b>	188
<b>Резюме</b>	194
<b>Глава 10: WeatherPi</b>	195
<b>Модуль Sense HAT</b>	195
<b>Метеостанция</b>	197
<b>Исходное состояние</b>	204
<b>Запуск</b>	209
<b>Резюме</b>	212

# Предисловие

Новый член семейства Raspberry Pi оснащен модулями беспроводной связи и Bluetooth. С помощью этой книги вы сможете создавать потрясающие проекты, которые позволят вам изучить возможности новой платы. Всего за 10 долларов вы можете развить навыки и стать частью нового мира, где вы сможете разрабатывать своих роботов, системы домашней автоматике, повышать уровень безопасности в своем доме и многое другое.

## Что охватывает эта книга

В главе 1 «Введение в Raspberry Pi Zero W» будет представлена новая плата с беспроводным расширением. Мы упомянем некоторые модули, которые вы можете купить, и дадим общий обзор новой платы.

В главе 2 «Интернет вещей и сети» речь пойдет об Интернете вещей. Плата Raspberry Pi Zero W очень мала, что делает ее идеальной платой для проектов IoT. Она также оснащена Bluetooth Low Energy (BLE), который минимизирует потребление энергии. Наконец, мы дадим введение в сеть и как вы будете взаимодействовать с платой.

В главе 3 «Чат-бот» будет разработан чат-бот. В настоящее время чат-боты распространены повсеместно и обычно состоят из клиента и сервера. Итак, мы начнем с создания клиента, а затем закончим главу реализацией серверной части. Мы также упомянем некоторые протоколы, которые используют чат-боты.

Глава 4 «Мобильный робот» будет посвящена робототехнике. Вы узнаете, как построить двухколесного мобильного робота, и в этой главе вы изучите основы робототехники, необходимые навыки для разработки любого типа робота.

В главе 5 «Домашний бот» объясняется, как создать домашнего бота и автоматизировать свой дом с помощью команд с локального веб-сайта или мобильного телефона. Будет реализована и описана общая система домашней автоматике.

В главе 6 «Камера видеонаблюдения» будет продемонстрирована реализация системы безопасности. Добавление камеры в ваш дом делает его умнее, и вы всегда можете видеть и записывать то, что происходит.

**Глава 7, Портативные колонки**, глава посвящена музыке. Поскольку Raspberry Pi Zero W довольно мал, его довольно легко смонтировать с несколькими динамиками в портативные колонки, чтобы слушать любимую музыку в любом месте. Вам нужен только внешний аккумулятор.

**Глава 8, Хостинг WebPi**, посвящена веб-хостингу и разработке. С платой Raspberry Pi Zero W вы можете размещать веб-сайты и разрабатывать простые сайты, которые помогут вам управлять своим домом и автоматизировать работу. В этой главе вы узнаете, как создать веб-сайт, загрузить его на Raspberry Pi и опубликовать в мировом сообществе.

**Глава 9, AlexaPi**, посвящена Alexa Voice Service (AVS) от Amazon. Мы загрузим AVS на нашу плату Raspberry Pi Zero W и разработаем систему, которая позволит нам управлять всем, что подключено к AVS. Мы также можем поговорить с Alexa и получить ответы на такие вопросы, как «Который час?».

В главе 10, WeatherPi, будет разработана метеостанция, с помощью которой вы сможете получать измерения температуры и влажности в своем доме или любом другом месте.

## Что нужно для этой книги

Даже если вы работаете во всех операционных системах, лучше иметь операционную систему Linux. Очевидно, вам нужен базовый комплект Raspberry Pi Zero W с камерой. Кроме того, вам понадобится модуль [SenseHAT](#) и микрофон или гарнитура. Вам также понадобится [динамик Bluetooth](#) и, наконец, все компоненты, необходимые для двухколесного мобильного робота.

## Для кого эта книга

Если вы любитель или энтузиаст, который хочет купить новейшую версию Raspberry Pi Zero W и использовать ее для создания интересных беспроводных проектов, эта книга для вас. Знание предварительного программирования с некоторым опытом работы с электроникой было бы полезно.

# 1

## Введение в Raspberry Pi Zero W

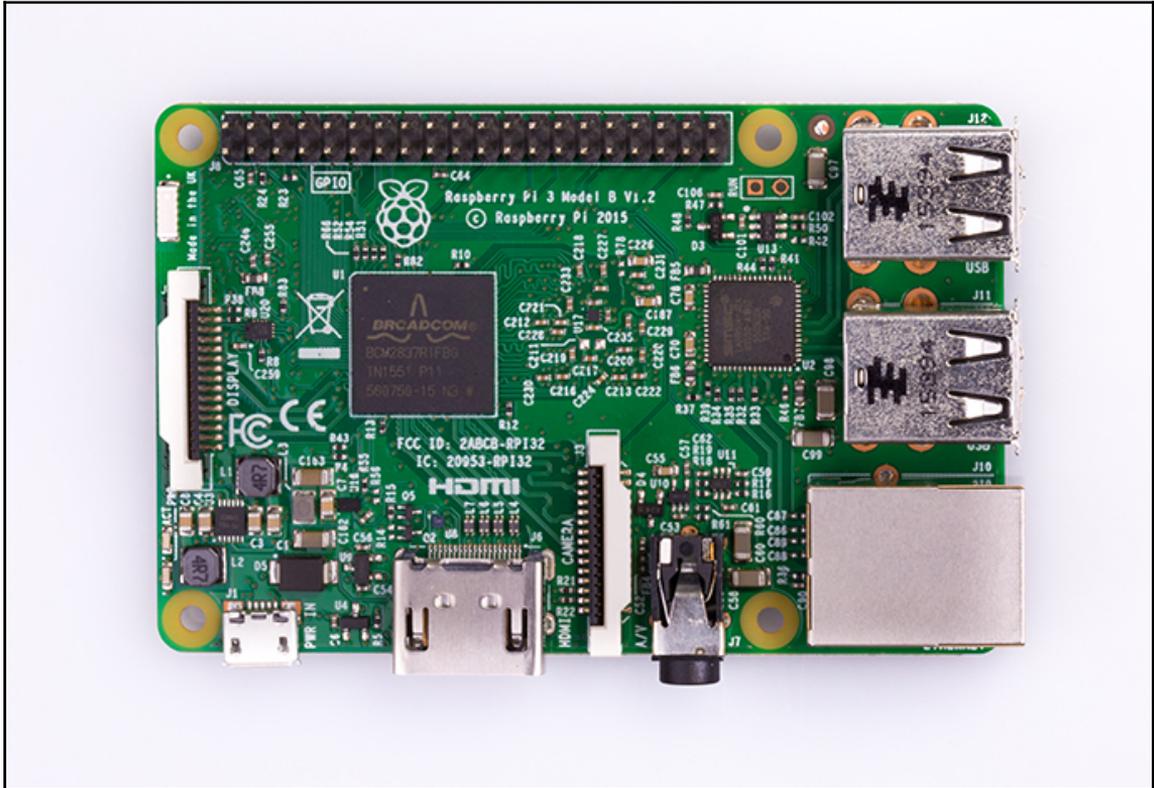
Raspberry Pi Zero W — новый продукт семейства Raspberry Pi Zero. В начале 2017 года сообщество Raspberry Pi анонсировало новую плату с беспроводным расширением. Он предлагает беспроводную функциональность, и теперь каждый может разработать свой собственный проект без кабелей или других компонентов. Сравнивая новую плату с Raspberry Pi 3 Model B, мы легко видим, что она намного меньше, с множеством возможностей для Интернета вещей. Однако что такое Raspberry Pi Zero W и зачем он вам нужен? Давайте пройдемся по остальным членам семьи и представим новую плату. В этой главе мы рассмотрим следующие темы:

- Обзор семейства Raspberry Pi
- Введение в новый Raspberry Pi Zero W
- Распространение
- Дистрибьюторы
- Общие проблемы

### Семейство Raspberry Pi

Как упоминалось ранее, Raspberry Pi Zero W является новым членом семейства плат Raspberry Pi. На протяжении многих лет Raspberry Pi развивался и становился все более удобным для пользователя благодаря безграничным возможностям. Давайте посмотрим на остальную часть семейства, чтобы понять, чем отличается плата Pi Zero.

На данный момент последняя плата называется Raspberry Pi 3 Model B. Это лучшее решение для таких проектов, как распознавание лиц, отслеживание видео, игры или что-то еще, что требует больших ресурсов:



A Raspberry Pi 3 model

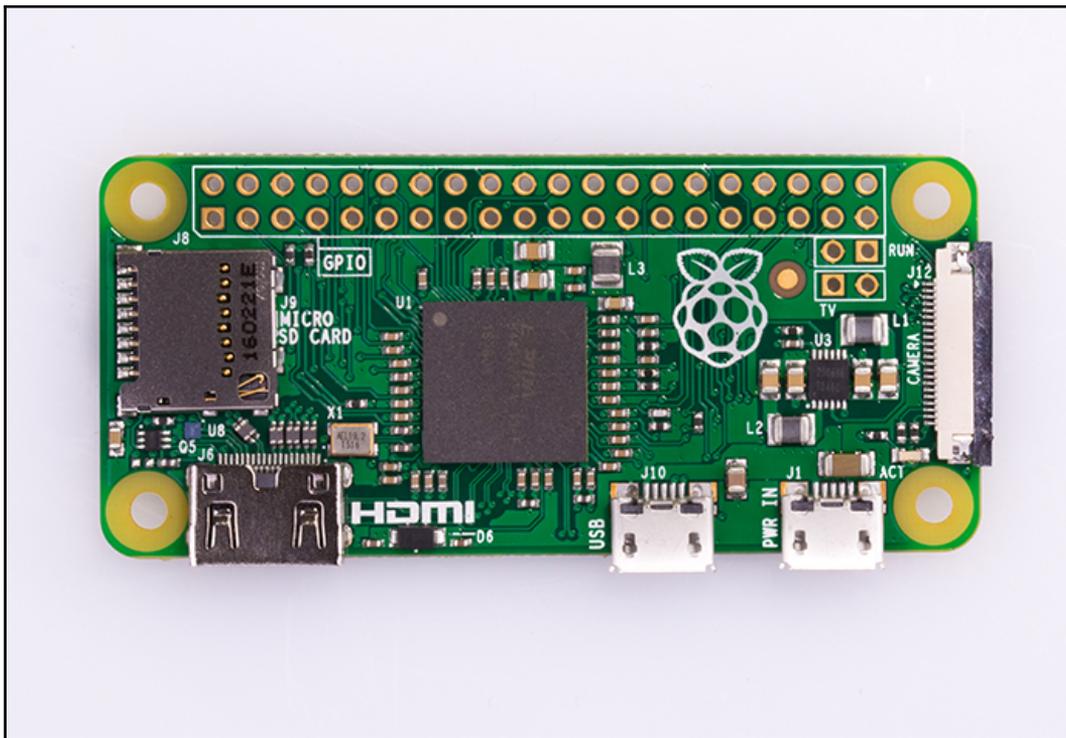
Это третье поколение плат Raspberry Pi после Raspberry Pi 2, имеющее следующие характеристики:

- 64-разрядный четырехъядерный процессор ARMv8 с тактовой частотой 1,2 ГГц.
- Беспроводная локальная сеть 802.11n
- Bluetooth 4.1
- Bluetooth Low Energy (BLE) - Bluetooth с низким энергопотреблением

Как и Pi 2, он также имеет:

- 1 ГБ оперативной памяти
- 4 порта USB
- 40 контактов GPIO
- Полный порт HDMI
- Порт Ethernet
- Комбинированный аудиоразъем 3,5 мм и композитное видео
- Интерфейс камеры (CSI)
- Интерфейс дисплея (DSI)
- Слот для карты MicroSD
- Video Core IV 3D графическое ядро

Следующая плата © Raspberry Pi Zero, на которой основана Zero W, небольшая и недорогая силовая плата, способная на многое:



Плата Raspberry Pi Zero

Технические характеристики этой платы следующие:

- 1 ГГц, одноядерный процессор
- 512 МБ ОЗУ
- Порт мини-HDMI
- OTG-порт микро-USB
- Питание микро-USB
- HAT-совместимый 40-контактный разъем
- Композитное видео
- Разъем камеры CSI (только v1.3)

На этом этапе помимо плат, упомянутых ранее, есть несколько других доступных модулей и компонентов, таких как Sense Hat или Raspberry Pi Touch Display, которые хорошо подойдут для продвинутых проектов.

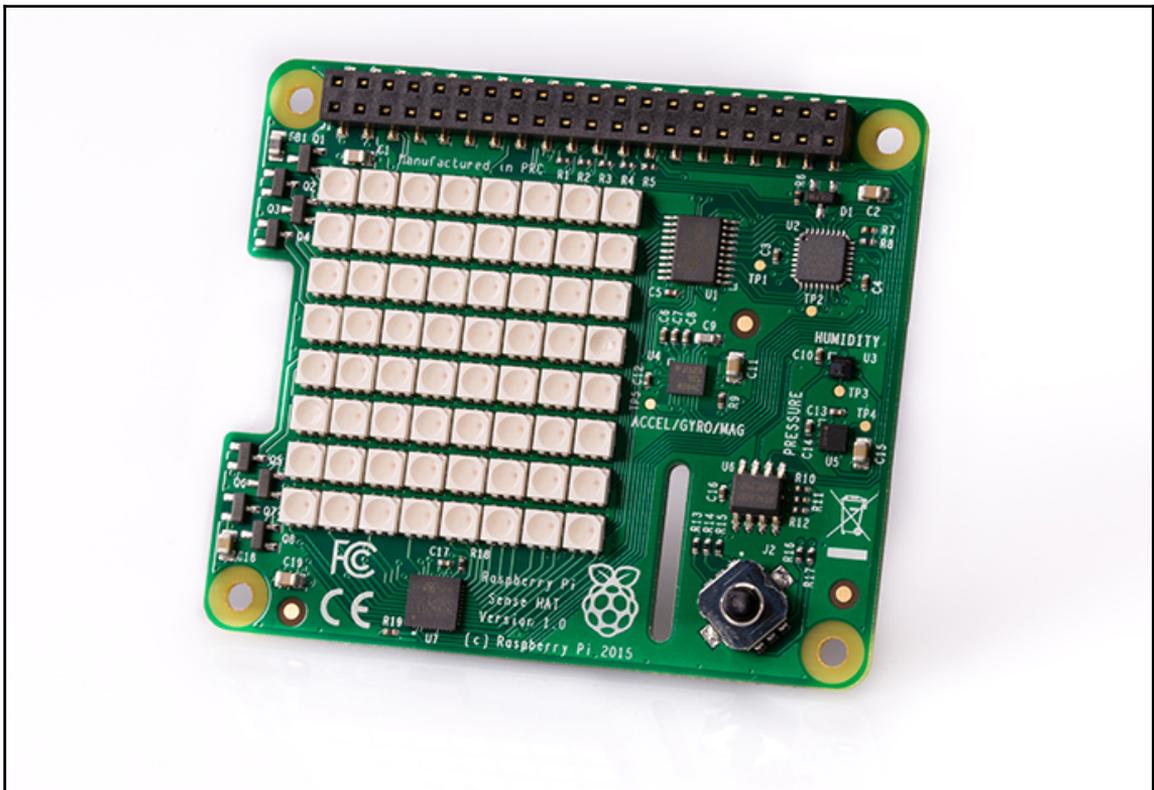
7-дюймовый сенсорный монитор Raspberry Pi дает пользователям возможность создавать универсальные интегрированные проекты, такие как планшеты, информационно-развлекательные системы и встроенные проекты:



Сенсорный дисплей Raspberry Pi

Sense HAT — это дополнительная плата для Raspberry Pi, созданная специально для миссии Astro Pi. Sense HAT имеет светодиодную матрицу RGB 8x8, пятикнопочный джойстик и включает в себя следующие датчики:

- Гироскоп
- Акселерометр
- Магнитометр
- Температура
- Барометр
- Влажность

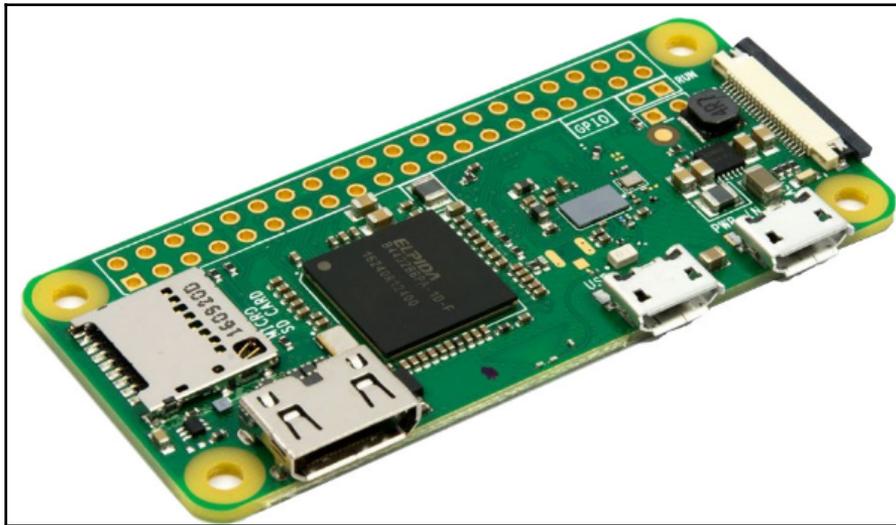


Плата Sense HAT

Следите за новыми платами и модулями на официальном сайте <https://www.raspberrypi.org>.

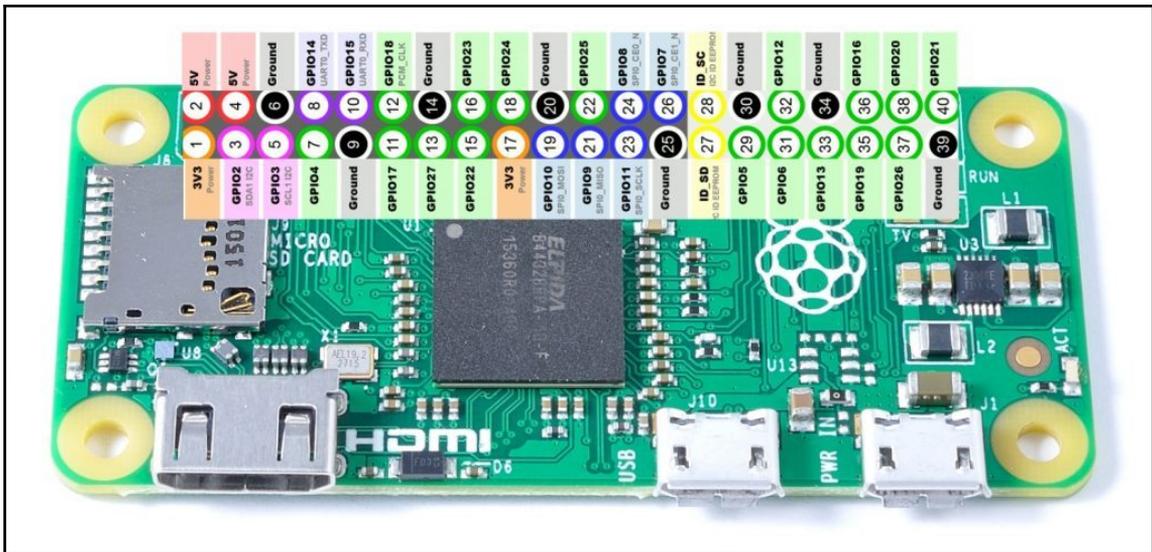
## Raspberry Pi Zero W

Raspberry Pi Zero W — это небольшое устройство, которое можно подключить как к внешнему монитору, так и к телевизору и, конечно же, к Интернету. Операционная система различается, так как на официальной странице много дистрибутивов и почти каждый дистрибутив основан на Linux:



Raspberry Pi Zero W

С Raspberry Pi Zero W у вас есть возможность делать практически все, от автоматике до игр! Это небольшой компьютер, который позволяет легко программировать через контакты GPIO и некоторых других компонентов, таких как камера. Его возможности безграничны! В следующей главе вы проведете несколько замечательных проектов с этой новой платой. Поскольку почти все входы и выходы на плате Raspberry Pi Zero W проходят через контакты GPIO, важно знать распиновку. Ниже приведена схема разводки платы Raspberry Pi Zero W, которая может пригодиться при пайке кнопок или других типов датчиков на плату Raspberry Pi:

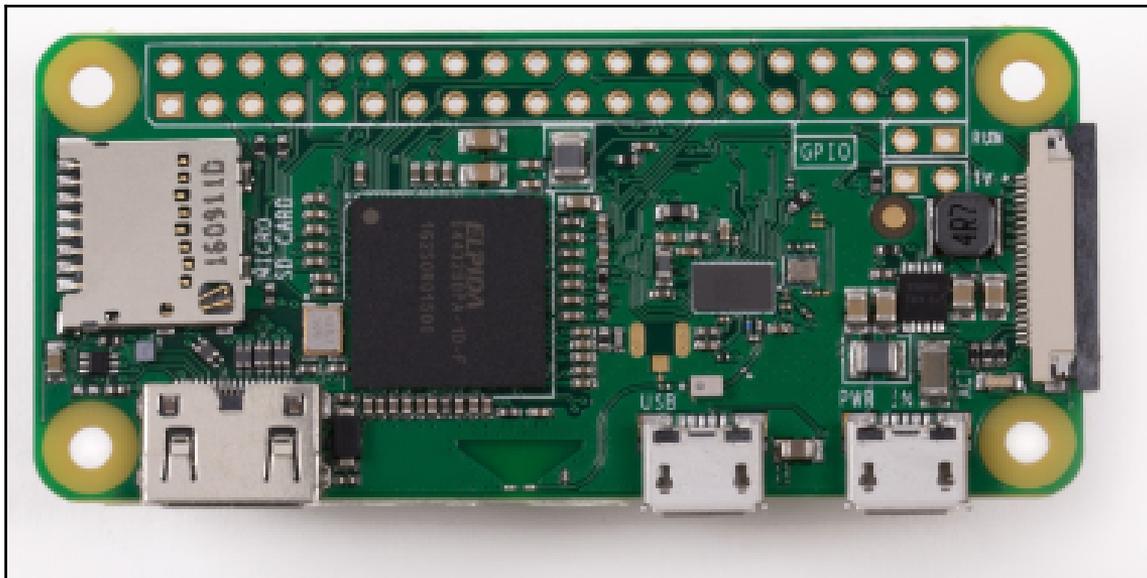


## Характеристики

Если вы купили Raspberry Pi 3 Model B, вы знакомы с беспроводным чипом Cypress CYW43438. Он обеспечивает подключение к беспроводной локальной сети 802.11 n и Bluetooth 4.0. Новый Raspberry Pi Zero W также оснащен этим беспроводным чипом. Ниже приведены технические характеристики платы:

- **Размеры:** 65 mm × 30 mm × 5 mm
- **SoC:** чип Broadcom BCM 2835
  - ARM11 с тактовой частотой 1 ГГц, одноядерный процессор
  - 512 МБ ОЗУ
- **Память:** карта MicroSD
- **Видео и аудио:** HD-видео 1080p и стереозвук через разъем mini HDMI
- **Питание:** 5 В через разъем micro-USB.
- **Wireless:** беспроводная локальная сеть 802.11n 2,4 ГГц
- **Bluetooth:** классический Bluetooth 4.0 с низким энергопотреблением (BLE)
- **Выход:** микро-USB
- **GPIO:** 40-контактный GPIO, незаполненный

На следующем изображении мы видим Raspberry Pi Zero W :



Raspberry Pi Zero W

Обратите внимание, что все компоненты находятся в верхней части платы и поэтому вы можете легко подобрать для нее корпус и сохранить плату в целости и сохранности. Что касается антенны, то она формируется путем травления меди на каждом слое печатной платы. Она может быть не видна, как в других подобных платах, но отлично работает и предлагает довольно много функций:

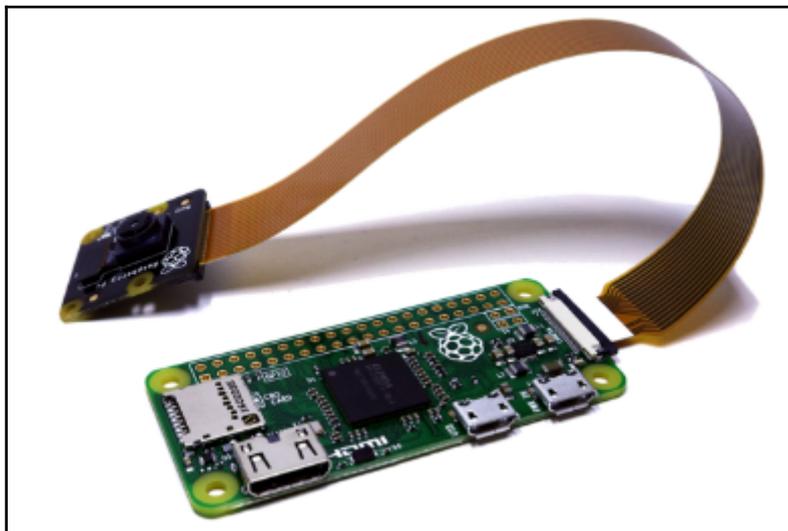


Конденсаторы Raspberry Pi Zero W

Кроме того, продукт ограничен только одной штукой на покупателя и стоит 10 долларов. Вы можете купить полный комплект с картой MicroSD, чехлом и некоторыми дополнительными компонентами примерно за 45 долларов или выбрать полный комплект с камерой, содержащей небольшой компонент камеры, за 55 долларов.

## Поддержка камеры

Для проектов обработки изображений, таких как отслеживание видео или распознавание лиц, требуется камера. На следующем изображении вы можете увидеть официальную поддержку камеры для Raspberry Pi Zero W. Камеру можно легко установить сбоку платы с помощью кабеля, аналогично плате Raspberry Pi 3 Model B.



Официальная поддержка камеры для Raspberry Pi Zero W

В зависимости от вашего дистрибутива вам может потребоваться включить камеру через командную строку. Более подробная информация об использовании этого модуля будет упомянута в главах проекта.

## Аксессуары

При создании проектов с помощью этой платы есть несколько других гаджетов, которые можно использовать для работы. Ниже приведен список некоторых важных компонентов. Обратите внимание, что если вы покупаете комплект Raspberry Pi Zero W, он включает в себя некоторые из них.

- OTG-кабель
- PowerHub
- Разъем GPIO

- Карта MicroSD и адаптер карты
- Кабель HDMI-mini HDMI
- Кабель HDMI-VGA

## **OTG-кабель**

Во-первых, кабель OTG всегда пригодится. Вы можете использовать этот кабель для питания вашего RaspberryPi от блока питания или любого другого источника питания.



OTG-кабель

## PowerHub

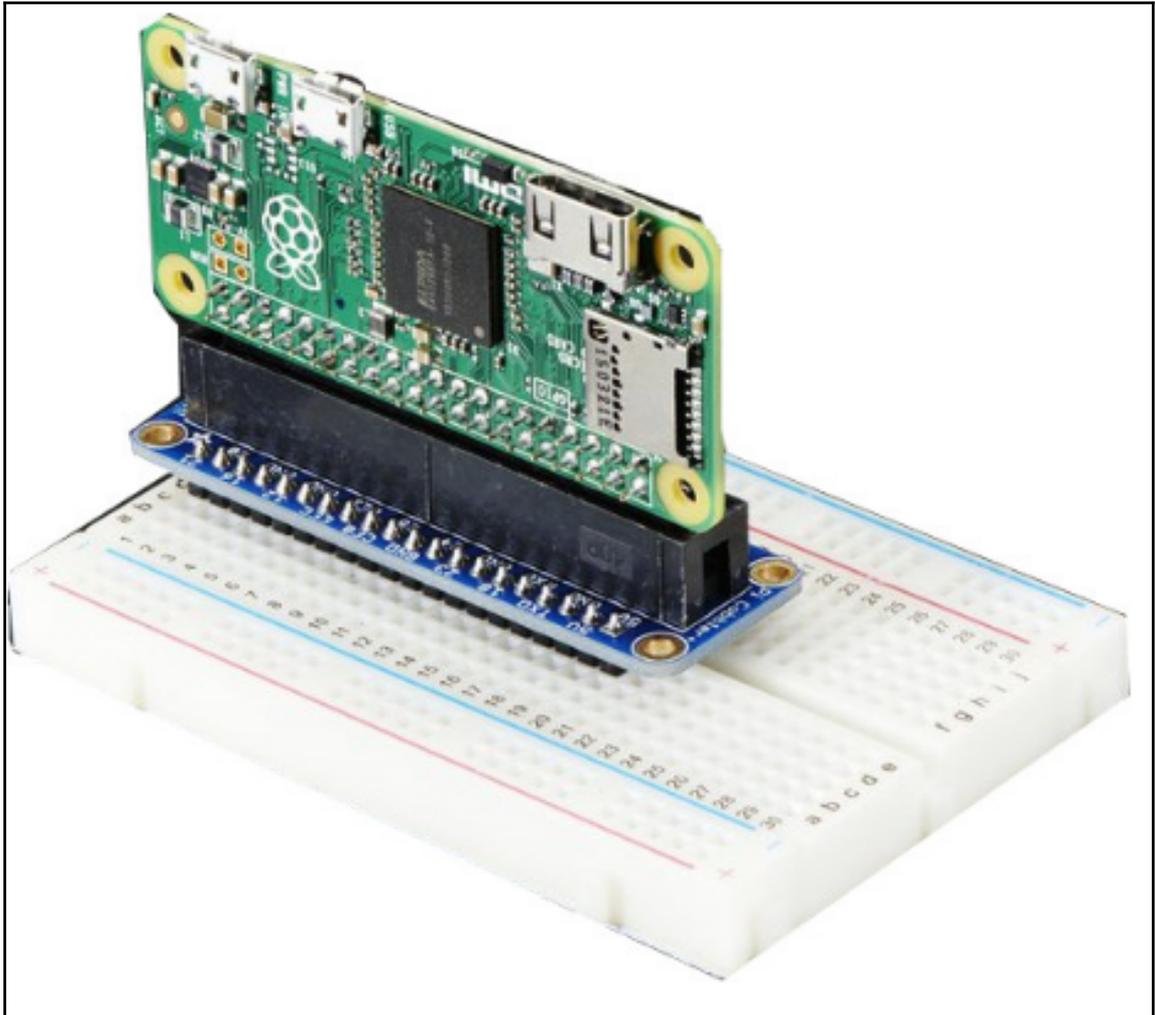
Вторым по важности компонентом является PowerHub. PowerHub — это устройство с питанием от USB или внешнего источника питания, имеющее четыре или более USB-портов:



PowerHub

## Разъем GPIO

Далее вам может пригодиться модуль разъема GPIO. Поскольку Raspberry Pi Zero W поставляется без припаянных контактов, полезно подключить контакты GPIO поверх макетной платы:



Разъем GPIO

## Карта MicroSD и адаптер карты

Вам также может понадобиться адаптер карты MicroSD, поскольку не на каждом компьютере есть слот для карты MicroSD для чтения и записи данных. Он стоит довольно много долларов, но экономит ваше время.



Карта MicroSD и адаптер карты

## Кабель HDMI-mini HDMI

К сожалению, новый Raspberry Pi Zero W не имеет нормального порта HDMI. Он немного меньше, и вам нужен кабель HDMI-mini HDMI, чтобы расширить порт до обычного HDMI. Затем вы можете подключить к порту любое HDMI-совместимое устройство. Итак, следующий кабель необходим:



Кабель HDMI-mini HDMI

## Кабель HDMI-VGA

Поскольку многие мониторы не совместимы с HDMI, кабель HDMI-VGA позволяет подключать кабель HDMI-мини-HDMI к внешнему монитору или телевизору. Когда вам нужно что-то посмотреть на мониторе, а он не поддерживает HDMI, необходим этот кабель:



Кабель HDMI-VGA

## Разъемы RCA

Наконец, некоторые телевизоры все еще используют разъемы RCA. С помощью этого модуля вы можете подключить плату Pi к разъему RCA на телевизоре, просто соединив два провода + (сигнал) и - (земля) модуля.



Разъем RCA

## Корпус Raspberry Pi Zero W

Поскольку Raspberry Pi Zero W будет использоваться для различных проектов, рекомендуется потратить еще несколько долларов и купить корпус или купить полный комплект, который поставляется с официальным корпусом, как показано на следующем изображении. Официальный чехол достаточно защитный и открывает только необходимые детали, позволяя вам полностью использовать свою плату. Вся работа выполняется в Kinneir Dufort и T-Zero. Доступны следующие варианты:

- Пустой
- Один с выемкой для доступа к GPIO
- Один с выемкой и точкой крепления камеры

Мы можем видеть корпус Raspberry Pi Zero W на следующем изображении:



Корпус Raspberry Pi Zero W

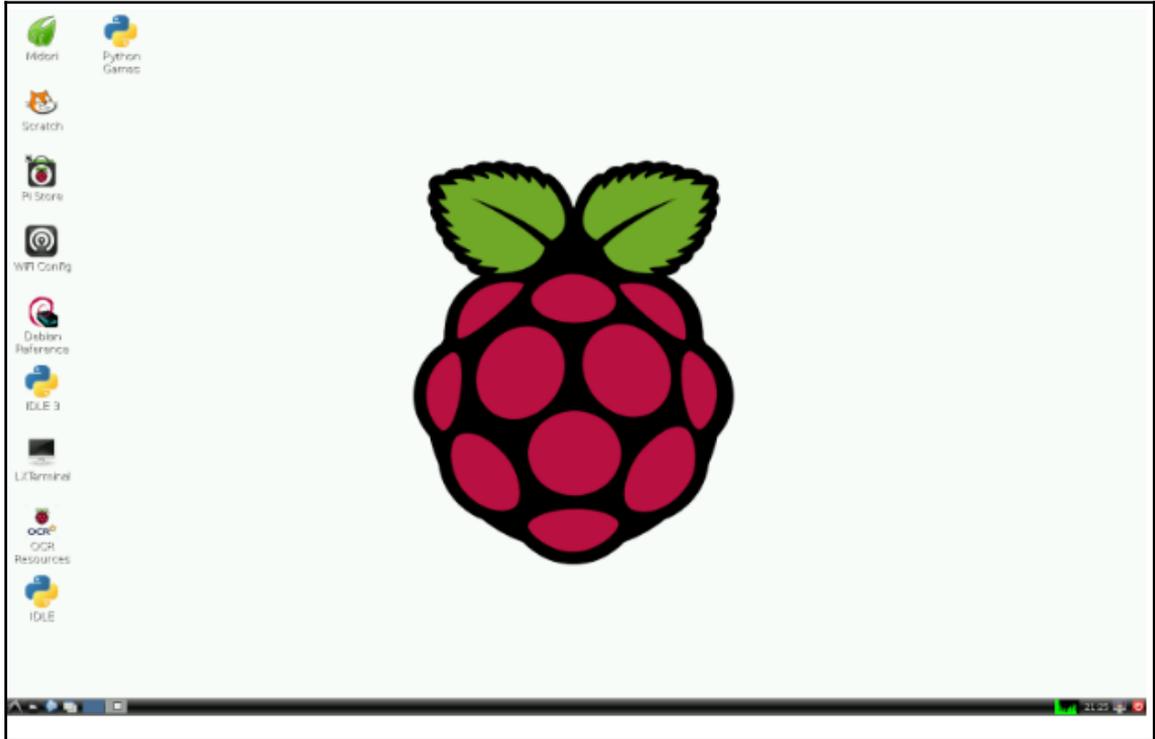
В официальный набор кейсов также входят:

- Адаптер для короткой камеры, гибкий
- Набор резиновых ножек, чтобы ваша новая плата Zero W не соскальзывала со стола.

## Дистрибутивы

Официальный сайт <https://www.raspberrypi.org/downloads/> содержит несколько дистрибутивов для скачивания. Две основные операционные системы, которые мы проанализируем позже, — это Raspbian и NOOBS. Вы можете увидеть, как выглядит среда рабочего стола на следующем изображении. И Raspbian, и NOOBS позволяют выбирать из двух версий. Есть полная версия операционной системы и облегченная. Очевидно, что облегченная версия не содержит всего, что вы могли бы использовать, поэтому, если вы собираетесь использовать Raspberry с окружением рабочего стола, выберите и загрузите полную версию.

С другой стороны, если вы собираетесь просто использовать SSH и делать некоторые базовые вещи, выберите облегченный вариант. Это действительно зависит от вас, и, конечно же, вы можете легко загрузить все, что вам нравится, и перезаписать карту microSD.



## Дистрибутив NOOBS

Загрузите NOOBS с <https://www.raspberrypi.org/downloads/noobs/>. Дистрибутив NOOBS предназначен для новых пользователей, которые не обладают достаточными знаниями о системах Linux и платах Raspberry Pi. Как говорится на официальной странице, это действительно «Новое программа из коробки». загрузите NOOBS и запишите свою собственную карту microSD. Если у вас возникли проблемы с определенным дистрибутивом, взгляните на следующие ссылки:

- Полное руководство :<https://www.raspberrypi.org/learning/software-guide/>
- Посмотрите видео:<https://www.raspberrypi.org/help/videos/#noobs-setup>

Операционная система NOOBS содержит Raspbian и предоставляет различные другие операционные системы, доступные для загрузки.

## Дистрибутив Raspbian

Скачайте Raspbian с официальной страницы <https://www.raspberrypi.org/downloads/raspbian/>. Raspbian — официально поддерживаемая операционная система. Его можно установить через NOOBS или загрузив файл образа по следующей ссылке и пройдя руководство на официальном сайте.

Файл изображения можно скачать с: <https://www.raspberrypi.org/documentation/installation/installing-images/README.md>

Он имеет множество предустановленных программ, таких как Python, Scratch, Sonic Pi, Java и Mathematica.

Кроме того, дополнительные дистрибутивы, такие как Ubuntu MATE, Windows 10 IoT Core или Weather Station, предназначены для установки для более конкретных проектов, таких как IoT или метеостанции. В заключение, соответствующий дистрибутив для установки на самом деле зависит от вашего проекта и вашего опыта в системном администрировании Linux.

Raspberry Pi Zero W нужна карта MicroSD для размещения любой операционной системы. Вы сможете загружать Raspbian, Noobs, Ubuntu MATE или любую другую операционную систему, которая вам нравится. Итак, все, что вам нужно сделать, это просто записать свою операционную систему на эту карту MicroSD. Прежде всего, вам необходимо загрузить файл изображения с <https://www.raspberrypi.org/downloads/>, который обычно поставляется в виде файла .zip. После загрузки разархивируйте zip-файл; полный образ весит около 4,5 гигабайт. В зависимости от вашей операционной системы вы можете использовать различные программы следующим образом:

- 7-Zip от Windows
- Разархиватор для Mac
- Unzip от Linux

Теперь мы должны записать образ на карту MicroSD. Вы можете легко записать файл .img на карту MicroSD, следуя одному из этих руководств, в зависимости от вашей системы.

Для пользователей Linux рекомендуется инструмент dd. Перед подключением карты MicroSD с адаптером к компьютеру выполните следующую команду:

```
df -h
```

Теперь подключите карту и снова запустите ту же команду. Вы увидите несколько новых записей. Например, если новое устройство называется `/dev/sdd1`, имейте в виду, что карта будет `/dev/sdd` (без 1).

Следующим шагом будет использование команды `dd` и копирование файла `.img` на карту MicroSD. Мы можем сделать это с помощью следующей команды:

```
dd if=<path to your image> of=</dev/**>
```

□де `if` — входной файл (файл изображения или дистрибутив), а `of` — выходной файл (MicroSDcard). Опять же, будьте осторожны и используйте только `/dev/sdd` или как там у вас без цифр. Если у вас возникли проблемы с этим, воспользуйтесь полным руководством по ссылке

<https://www.raspberrypi.org/documentation/installation/install-images/linux.md>.

Хороший инструмент, который может помочь вам в этой работе, — `GParted`. Если он не установлен в вашей системе, вы можете легко установить его с помощью следующей команды:

```
sudo apt-get install gparted
```

Затем запустите `sudo gparted`, чтобы запустить инструмент. Он очень легко обрабатывает разделы, и вы можете форматировать, удалять или находить информацию обо всех ваших смонтированных разделах.

Более подробную информацию о `dd` можно найти

на <https://www.raspberrypi.org/documentation/installation/installing-images/linux.md>.

- Для пользователей Mac OS всегда рекомендуется инструмент `dd` (<https://www.raspberrypi.org/documentation/installation/installing-images/mac.md>)
- Для пользователей Windows рекомендуется утилита `Win32DiskImager` (<https://www.raspberrypi.org/documentation/installation/installing-images/windows.md>)

Существует несколько других способов записи файла изображения на карту microSD. Поэтому, если у вас возникнут какие-либо проблемы при следовании предыдущим руководствам, не стесняйтесь использовать любое другое руководство, доступное в Интернете. Теперь, предполагая, что все в порядке и образ готов, можно аккуратно вставить карту MicroSD в плату Raspberry Pi Zero W.

Помните, что вы всегда можете подтвердить успешность загрузки с помощью кода `sha1`. В системах Linux вы можете использовать `sha1sum`, за которым следует имя файла (изображение), и распечатать код `sha1`. Это должно и должно быть таким же, как в конце официальной страницы, где вы скачали образ.

## Дистрибьюторы

Как упоминалось ранее, к сожалению, плата Raspberry Pi Zero W ограничена одной платой на пользователя. Ниже приведен список официальных дистрибьюторов, упомянутых на официальном сайте Raspberry Pi. Таким образом, в зависимости от вашего местоположения, обратитесь к магазину соответственно.

Вы можете купить плату Raspberry Pi Zero W отдельно или в комплекте с некоторыми дополнительными компонентами. Предполагая, что Raspberry Pi Zero поставляется с мини-HDMI и нуждается в карте microSD, вы можете купить полный комплект, в зависимости от вашей комплектации. Также практически везде есть комплект, содержащий модуль камеры. В зависимости от ваших будущих проектов вам может понадобиться комплект камеры.

В результате рекомендуется сначала решить, что вы хотите делать и что вы планируете делать с Raspberry Pi Zero W, а затем заказать плату с дополнительными функциями или без.

## Общие проблемы

Иногда работа с платами Raspberry Pi может привести к проблемам. Мы все сталкивались с некоторыми из них и надеемся, что никогда не столкнемся с ними снова. Pi Zero настолько минимален, что трудно сказать, работает он или нет. Поскольку на плате нет светодиода, иногда бывает удобно быстро проверить, работает ли он правильно или что-то пошло не так.

## Шаги отладки

Выполнив следующие шаги, вы, вероятно, найдете его статус:

1. Возьмите свою плату, в которой ничего нет ни в одном слоте или разъеме. Извлеките даже карту microSD!
2. Возьмите обычный кабель micro-USB-USB-ADATA и подключите одну сторону к компьютеру, а другую Ⓢ к USB-порту Pi (не к PWR\_IN).
3. Если Zero жив:
  - В Windows ПК издаст звуковой сигнал, чтобы сообщить о наличии нового оборудования, и вы увидите загрузку BCM2708 в диспетчере устройств.
  - В Linux с идентификатором 0a5c:2763 Broadcom Corp сообщение от dmesg попробуйте запустить dmesg в терминале перед подключением USB. После этого вы найдете там новую запись.

Ниже приведен пример вывода:

```
[226314.048026] usb 4-2: new full-speed USB device number 82
using uhci_hcd [226314.213273] usb 4-2: New USB device found,
idVendor=0a5c, idProduct=2763 [226314.213280] usb 4-2: New USB
device strings: Mfr=1, Product=2, SerialNumber=0
[226314.213284] usb 4-2: Product: BCM2708 Boot
[226314.213] usb 4-2: Manufacturer: Broadcom
```

Если вы видите какой-либо из предыдущих вариантов, значит все хорошо и знайте - Зеро не умер.

## Проблема с картой памяти MicroSD

Помните, что если вы загружаете свой Raspberry и ничего не работает, возможно, вы неправильно записали карту microSD. Это означает, что ваша карта может не содержать ни одного загрузочного раздела, как положено, и она не может загрузить первые файлы. Эта проблема возникает, когда дистрибутив записывается в /dev/sdd1, а не в /dev/sdd, как должно быть. Это распространенная ошибка, и она не идентифицируется.

## Защита корпуса

Платы Raspberry Pi — это электроника, и мы никогда не размещаем электронику на металлических поверхностях или рядом с магнитными объектами. Итак, совет: потратьте немного денег на корпус Raspberry Pi и защитите свою плату от всего подобного.

## Резюме

Raspberry Pi Zero W — это новая многообещающая плата, позволяющая любому подключить свои устройства к Интернету и использовать свои навыки для разработки проектов, включая программы и компоненты. Эта плата — новая игрушка для любого инженера, интересующегося Интернетом вещей, безопасностью, автоматизацией и многим другим! Мы ознакомились с новой платой Raspberry Pi Zero и остальной частью ее семейства, а также с кратким анализом некоторых дополнительных компонентов, которые вам также следует приобрести. В следующей главе мы рассмотрим IoT и сеть, чтобы вы могли понять, как подключить Raspberry Pi Zero к вашей сети, настроить ее и начать создавать проекты.

# 2

## Интернет вещей и сети

В предыдущей главе мы представили новую плату Raspberry Pi Zero W. Мы также упомянули несколько аксессуаров, которые можно купить вместе с платой, чтобы облегчить себе жизнь, не пытаясь соединить все вместе. Перед разработкой проекта с этой платой необходимо кое-что упомянуть и проанализировать с разных точек зрения. Поскольку Raspberry Pi Zero W будет подключен к Интернету, пришло время поговорить об Интернете вещей и сетях. В этой главе мы рассмотрим следующие темы:

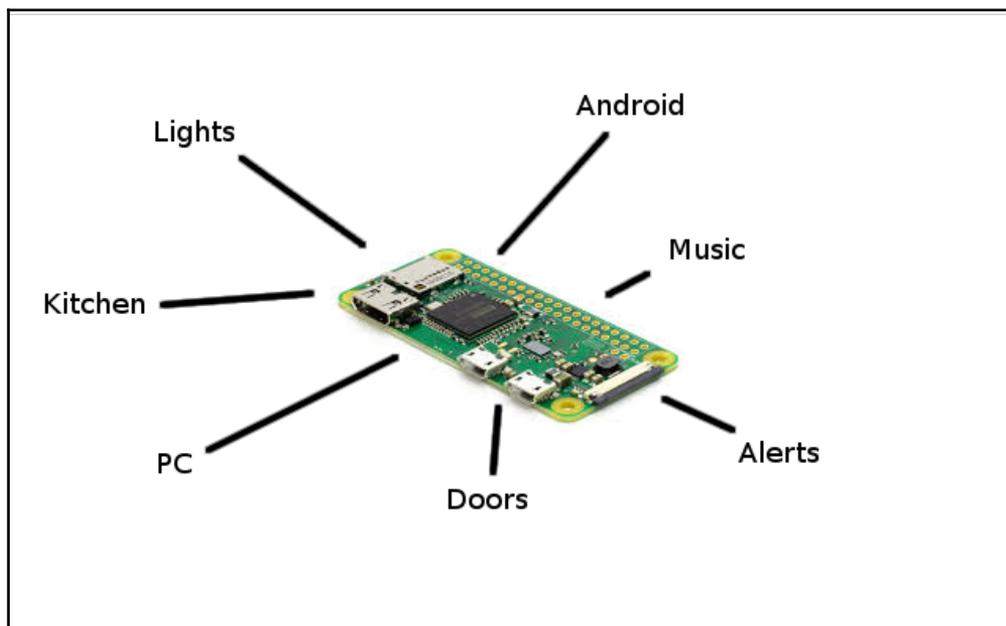
- Интернет вещей
- Основные протоколы связи
- Настройка Pi Zero
- Настройка роутера
- Сетевое администрирование
- Безопасность

### Интернет вещей

Два слова, скрывающие новый мир. Через несколько лет каждая вещь и каждое ваше устройство будет подключено к Интернету. С новой эрой Интернета вещей будет собираться огромное количество данных для ввода в алгоритмы или системы с искусственным интеллектом. Представьте, что ваш тостер оснащен датчиками, и вы знаете, какой хлеб вы используете, в какое время вы его активируете и сколько раз в день вы хотите съесть тост. Эта информация может быть бесполезной для вас, но для крупных компаний, готовых потратить миллиарды и рискнуть всем ради того, сколько тостов они должны произвести, это не просто что-то, это меняет правила игры. Итак, Интернет вещей — это все, что подключено к Интернету, например, ваш компьютер, свет, холодильник, дверь гаража и все остальное, к чему вы можете получить доступ через Интернет.

С другой стороны, автоматизация — это любая система, которая думает, что принимает решения. Например, система, которая определяет, когда ваша машина подъезжает к вашему дому, и автоматически открывает дверь гаража, является автоматизацией. Эти два слова проведут вас через оставшуюся часть книги и будут весьма полезны в вашей жизни.

Raspberry Pi, Arduino и все подобные платы предназначены для управления устройствами и вещами и подключения их к Интернету. Итак, эта книга об устройстве под названием Pi Zero W, которое позволяет вам управлять вещами в вашем доме и подключать их к Интернету, чтобы вы могли получить к ним доступ из любой точки мира:



## Основные протоколы связи

Предположим, что у нас есть Raspberry Pi в качестве нашего клиента и <https://www.facebook.com> в качестве нашего сервера. Когда наш клиент запрашивает веб-страницу с сервера, должны быть какие-то протоколы, какие-то правила, по которым будет обрабатываться этот запрос. Другими словами, должен быть определенный способ передачи и получения данных друг от друга. Интернет-протоколы (IP) — это именно те правила, по которым общаются клиент и сервер.

Есть несколько основных IP-адресов, с помощью которых современные браузеры связываются с серверами и получают данные. Однако когда мы говорим об IoT и оборудовании с низким энергопотреблением, мы не можем сохранить эти протоколы и использовать их. В результате важно пройтись по некоторым базовым вещам и понять идею всех коммуникаций.

## Интернет-протоколы

Поскольку этот раздел не является темой книги, мы рассмотрим некоторые основные протоколы, которыми мы все пользовались (возможно, даже не подозревая об этом). Протокол HTTP используется при запросе веб-страницы. Например, когда кто-то набирает <https://facebook.com/>, он запрашивает веб-страницу под названием facebook. Протокол FTP используется, когда клиент подключается к файловому серверу. Например, когда кто-то набирает <ftp.ntua.gr/>, он подключается к серверу, где есть только файлы, а не веб-сайт или блог. Кроме того, TCP и UDP — это протоколы, используемые для передачи данных. UDP используется для видеовызова или голосового вызова, когда нам нужно передать огромное количество данных, тогда как TCP используется для HTTP-запроса или подобных действий.

Причина, по которой мы рассмотрели эти протоколы, заключается в том, что если вы прочитаете о них дальше, вы увидите, что каждый из них имеет довольно много байтов, необходимых для правильной работы, что означает питание по кабелю, что, в свою очередь, означает потребление энергии.

## Протоколы Интернета вещей

Интернет вещей создал новую потребность в расширенных протоколах на многих уровнях. Нет необходимости в протоколах связи устройств, семантике, идентификации, инфраструктуре и многом другом. Существует множество протоколов, разработанных для такого рода работы, и никто не уверен, что будет работать идеально, а что нет в будущем. IoT — одно из главных исследований, которым интересуются многие университеты, и оно очень сложное. В этом разделе главы мы рассмотрим следующие основные протоколы IoT, необходимые для работы над связанными проектами.

- Bluetooth
- Zigbee
- Z-Wave
- 6LowPan
- Thread
- Wi-Fi

- Cellular
- NFC
- LoRaWAN

## Bluetooth

Этот протокол связи является коротким и известен почти всем. Это очень важно и, как ожидается, станет ключевым для носимых продуктов. Raspberry Pi Zero W поддерживает bluetooth, и это отличная возможность для вас создавать проекты и знакомиться с этим протоколом. Новый протокол Bluetooth Low Energy (BLE) или Bluetooth Smart является важным протоколом для Интернета вещей, и даже если он имеет аналогичный диапазон, он изначально был разработан для снижения энергопотребления. В заключение, говоря об IoT, все дело в энергопотреблении. Ваш Raspberry Pi Zero W может работать вечно, если он подключен к домашнему источнику питания, но если вы используете его для носимых устройств, таких как смарт-часы, то энергопотребление имеет большое значение.

**Bluetooth** работает на частоте 2,4 ГГц (ISM) с радиусом действия 20-150 м (BLE) и скоростью передачи данных до 1 Мбит/с (Smart/BLE).

## Zigbee

Протокол Zigbee также работает на частоте 2,4 ГГц и основан на протоколе IEEE 802.15.4. Он обменивается данными с низкой скоростью в ограниченном диапазоне 100 м. В результате Zigbee подходит для дома или малого бизнеса. Кроме того, этот протокол имеет некоторые преимущества, предлагая работу с низким энергопотреблением и высокую безопасность с масштабируемостью. Скорость передачи данных Zigbee достигает 250 Кбит/с.

## Z-Wave

**Z-Wave** — это технология радиочастотной связи с низким энергопотреблением, разработанная для умных домов и автоматизации. Скорость передачи данных составляет до 100 кбит/с, и он работает в диапазоне менее 1 ГГц. Он поддерживает полносвязные сети без необходимости в центральном узле, координирующем поток данных. Частота составляет 900 МГц (ISM) с дальностью действия до 30 м. В целом, это хороший протокол для связи устройств в домашней автоматизации.

## 6LowPAN

**6LowPAN** это сетевой протокол, предназначенный для механизмов инкапсуляции и сжатия заголовков. Он часто используется среди других протоколов, таких как Ethernet или Wi-Fi, и преимущество этого протокола в том, что он поддерживает IPv6, что является целью почти всех протоколов. Разработанный для домашней автоматизации, он обещает производить сложные системы управления и связываться с устройствами через беспроводную сеть с низким энергопотреблением.

## NFC

**Связь ближнего поля (NFC)** — это технология, которая обеспечивает тесную передачу или взаимодействие между двумя устройствами. Почти все новейшие устройства Android и IOS поддерживают эту технологию, и она весьма полезна для идентификации или запуска ситуаций. Он будет использоваться для домашней автоматизации или для проверки безопасности предприятий. Это фактически расширяет возможности карточной технологии и позволяет устройствам обмениваться информацией. Он работает на частоте 13,56 МГц (ISM) и имеет диапазон 10 см.

## Wi-Fi

**Wi-Fi** есть почти везде. Мы все подключены к сети через Wi-Fi, поэтому мы в значительной степени понимаем, как это работает. Он обрабатывает довольно большие объемы данных, и в настоящее время общепринятым стандартом Wi-Fi является 802.11n, обеспечивающий пропускную способность в сотни мегабит в секунду. Однако, поскольку у каждой хорошей вещи есть недостаток, Wi-Fi очень энергозатратен, поэтому этот протокол не очень хорош для IoT, когда речь идет о носимых устройствах или чем-то еще, где каждое потребление имеет решающее значение. Он работает на частоте 2,4 ГГц или 5 ГГц и поддерживает расстояние до 50 м.

## LoRaWAN

**LoRaWAN** предназначен для приложений глобальной сети (WAN) и представляет собой недорогой протокол с масштабируемостью. Идеально подходит для умных городов, он может соединять множество устройств со скоростью передачи данных от 0,3 до 50 Кбит/с. Его дальность действия составляет от 5 км до 15 км при отсутствии препятствий.

## Подключение вашего Pi

Подключение вашего пиНа этом этапе нам нужно подключить Raspberry Pi Zero W к Интернету. Чтобы сделать это, мы сначала должны понять, как работает Интернет. Как все подключено и как нам удастся подключить Raspberry Pi. В этом разделе книги показано, как управлять различными роутерами. Поскольку каждый процесс отличается, я попытаюсь описать здесь основную идею, и, надеюсь, вы сможете найти свое решение.

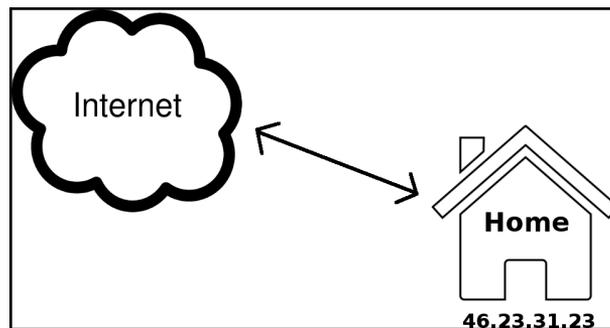
## Настройка Pi Zero W

Во-первых, чтобы следовать остальной части книги, вам нужно подключить Raspberry Pi Zero к монитору с помощью HDMI, а затем сделать следующее. Создайте новую папку с именем SH в папке /boot и перезапустите pi. Вам будет предложено сообщение о SSH. Если это сделано, вы можете продолжить оставшуюся часть главы и работать удаленно, используя SSH, как описано далее.

Также обратите внимание, что вам нужен последний образ. Старые образы с вашей старой платы Raspberry Pi или чего-либо еще не будут работать, и адаптер Wi-Fi не будет найден. Так что имейте в виду, что вам нужно установить последний образ.

## Интернет

Интернет состоит из ребер и узлов. Каждый компьютер — это узел, а каждый кабель — это край. ☐еспроводная связь имеет невидимый кабель, который называется воздушным:



Как вы можете видеть на предыдущем рисунке, мы можем представить Интернет как облако. Нам не обязательно понимать, что там внутри. Ваш Интернет-провайдер (ISP) предоставляет вам подключение к Интернету. Таким образом, обычно у вас дома есть роутер. Поскольку к Интернету подключены миллиарды узлов, которые обмениваются данными по протоколам и отправляют/получают сотни сообщений, должен быть способ определить, кто есть кто. Это означает, что роутер должен иметь уникальную идентификацию, такую как адрес в вашем доме. Этот адрес называется IP-адресом или Интернет-протоколом, и существует два протокола. Старый и классический — версия 4, а новый — версия 6. Для этого не требуется дополнительных объяснений, просто помните, что в каждом доме есть роутер, и каждый роутер имеет уникальный адрес IPv4. Итак, теперь все, кто находится за пределами вашего дома, знают, что каждый пакет вашего IPv4-адреса будет достигать вашего роутера.

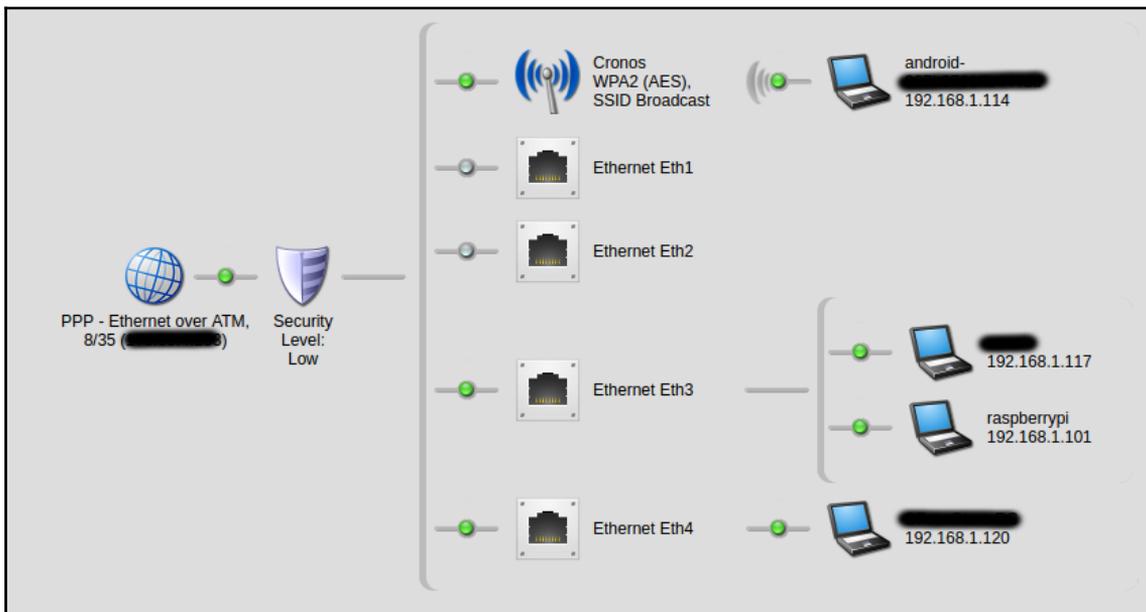
## **Локальная сеть**

Предположим, что на ваш роутер пришел пакет. Крайне важно понимать, что именно происходит и как роутер определяет место назначения пакета. В вашей локальной сети (дома) может быть 10 устройств, таких как планшеты, смартфоны, ПК и ноутбуки. Роутер должен решить, в какой из них он должен отправить пакет. Это делается с помощью протокола преобразования сетевых адресов (NAT). Каким-то образом он выбирает одно устройство и отправляет пакет. Каждое устройство в вашем доме имеет IP-адрес. Это не то же самое, что ваш домашний IP-адрес, о котором мы упоминали ранее. Когда вы подключаете устройство в локальной сети (роутер), оно автоматически получает IP-адрес, например 192.168.1.101.

Подводя итог, Интернет состоит из клиентов (домов) и серверов (конечных точек). Например, клиент — это ваш дом, а сервер — <https://www.facebook.com>. Клиент подключается к серверу через какие-то узлы (коммутаторы), обычно с помощью кабелей, и их там много. То, что мы описали ранее, — это Интернет. Идя дальше, локальная сеть состоит из коммутатора, который в данном случае называется роутером, и некоторых устройств, таких как ваш смартфон или ноутбук. Ваш роутер отвечает за доставку всего, что приходит к вам домой, на соответствующее устройство. Ваш роутер и каждое отдельное устройство имеют IP-адрес.

Структура вашей локальной сети представляет собой дерево. Это означает, что все начинается с роутера и расширений. У вас может быть еще одна локальная сеть в вашем гараже или во дворе. Например, на следующем снимке экрана видно, что моя сеть состоит из нескольких устройств и еще одного коммутатора, состоящего из еще нескольких устройств.

На следующем снимке экрана мы видим страницу моего роутера, и должно быть совершенно ясно, что помимо Wi-Fi есть два включенных порта Ethernet, один из которых подключен к компьютеру, а другой к чему-то, называемому коммутатором, который имеет два устройства. Первый — снова компьютер, а второй — наш Raspberry Pi Zero W. Вы можете найти IP-адрес Raspberry Pi на следующем снимке экрана:



Ваш Raspberry Pi Zero W будет одним устройством, подключенным к вашему роутеру, и будет иметь IP-адрес, как и любое другое устройство. Это устройство зависит от вашего роутера. Обычно IP-адреса в вашем доме имеют формат 192.168.1.X, где X находится в диапазоне от 1 до 254, но даже если числа различаются, идея та же. Обычно, поскольку маршрутизатор является первым устройством, подключенным к Интернету, он имеет IP-адрес 192.168.1.1, второе устройство может иметь 192.168.1.2, третье устройство — 192.168.1.3 и т. д., как вы можете видеть на следующем изображении. моя малина с IP 192.168.1.101. Вся предыдущая информация важна, потому что мы будем подключаться к нашему Raspberry Pi, используя SSH по сети. Но, конечно, чтобы подключиться к нему, мы должны сначала его найти.

## Перенаправление порта

На этом этапе разница между локальной сетью и всем Интернетом должна быть кристально ясной. Теперь посмотрим, как подключиться к нашему пи из внешнего мира? Если мы подключимся к нашему домашнему IP-адресу, мы ничего не получим, потому что наш роутер не сервер, а просто роутер. Наш сервер — Raspberry Pi, который находится внутри нашего дома с локальным IP-адресом. Ответ на этот вопрос — перенаправление портов. К сожалению, не все роутеры поддерживают это, но это необходимо для подключения к вашему Raspberry Pi из внешнего мира. Если у вас есть роутер, который не поддерживает переадресацию портов, подумайте о покупке хорошего роутера, поскольку хорошие роутеры оснащены опциями для лучшей безопасности и обработки соединений. Далее вы можете увидеть скриншот, на котором мы проложили маршрут из внешнего мира на определенный IP-адрес в локальной сети. Чтобы создать новый маршрут, вам необходимо войти в интерфейс роутера и щелкнуть вкладку «Port forwarding - Переадресация портов».

Description	Traffic Coming from	External Port	Internal Port	Local Host
Rasp Server	Any Host PPP - Ethernet over ATM, 8/35	TCP 1770	2880	raspberrypi (192.168.1.101)

Из предыдущего скриншота мы можем понять, что любое подключение со всего мира, пришедшее к вам домой на порт 1770, будет перенаправлено на порт 2880 на компьютере с IP 192.168.1.101, который является нашим Raspberry Pi Zero.

Остался сервер, который будет отвечать и работать на нашем Raspberry Pi Zero W.

## Подключение к Интернету

Предполагая, что мы настроили переадресацию всех внешних подключений к нашему Raspberry Pi Zero W, мы можем легко подключиться к нему из любой точки мира, просто используя внешний IP-адрес нашего дома. Например, если вы гуглите мой ip и получаете результат типа 45.12.64.23, то все, что вам нужно сделать, это закрыть соединение Wi-Fi с вашего мобильного телефона и, используя данные, попытаться перейти на <http://45.12.64.23/> (очевидно, измените это на свой IP-адрес). Вы должны иметь возможность общаться с Raspberry Pi через порт 80. HTTP-порт — это номер 80, и это порт, который вы запрашиваете, когда вводите URL-адрес, например <http://XXX>. Вы можете указать свой порт в URL-адресе, набрав <http://45.12.64.23:<port>> и в результате вы соединяете свое устройство с вашим домом в порт <port>, где порт — это число от 1000 до 4000. На самом деле, портов больше, но если вы не продвинутый пользователь, остановитесь на этом диапазоне.

## Сетевое администрирование

В этом разделе мы рассмотрим некоторые основные команды Linux, которые помогут определить будущие проблемы. Не стесняйтесь вернуться сюда и прочитать это снова. Есть много команд, которые мы можем выполнить в терминале для отладки нашего проекта или настройки связи Raspberry Pi Zero W, которую мы сделали. Сначала самая простая команда — это `ifconfig`. Команда `ifconfig` выводит некоторые интерфейсы и много информации о них. Например, вы можете легко узнать IP-адрес вашего устройства Raspberry Pi Zero W, просто набрав:

### Ifconfig

Как показано на следующем снимке экрана, IP-адрес можно легко найти. Если вы видите IP-адрес, это означает, что ваше устройство Raspberry Pi Zero W подключено к Интернету:

```
pi@raspberrypi:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:ed:23:bb
          inet addr:192.168.1.101  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::fed5:3971:f6a8:61a6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:483259 errors:0 dropped:40 overruns:0 frame:0
          TX packets:158544 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:135705578 (129.4 MiB)  TX bytes:19824896 (18.9 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:625 errors:0 dropped:0 overruns:0 frame:0
          TX packets:625 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:56274 (54.9 KiB)  TX bytes:56274 (54.9 KiB)

wlan0     Link encap:Ethernet  HWaddr b8:27:eb:b8:76:ee
          inet6 addr: fe80::bc24:b350:3b5b:c186/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:102707 errors:0 dropped:102707 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:36998689 (35.2 MiB)  TX bytes:0 (0.0 B)
```

Следующая команда, которая помогает при проблемах с сетью, — `ping`. С помощью `ping` мы можем определить, не работает ли сервер. Другими словами, пропинговав работающий сервер, мы можем определить, подключены ли вы к Интернету или нет. В следующем примере мы пропингуем `www.google.com` и посмотрим, получим ли мы ответ или нет. Введите следующую команду:

### ping -c 3 www.google.com

Если все в порядке, у вас должен быть результат, подобный следующему, где мы видим, что есть два ответа по 64 байта каждый:

```
pi@raspberrypi:~$ ping -c 3 www.google.com
PING www.google.com (216.58.208.36) 56(84) bytes of data:
64 bytes from fra15s12-in-f36.1e100.net (216.58.208.36): icmp_seq=1 ttl=54 time=135 ms
64 bytes from fra15s12-in-f36.1e100.net (216.58.208.36): icmp_seq=3 ttl=54 time=164 ms

--- www.google.com ping statistics ---
3 packets transmitted, 2 received, 33% packet loss, time 2005ms
rtt min/avg/max/mdev = 135.914/150.445/164.977/14.536 ms
pi@raspberrypi:~$ _
```

Аргумент `-c 3`, который мы дали, означает, что мы запрашиваем только три пинга, а затем останавливаем команду.

Теперь мы запустим простой сервер в нашем *Raspberry Pi*, чтобы мы могли убедиться, что все в порядке, и у нас есть доступ к нашему *Pi* со всего мира. Для этого мы используем язык *Python* и запустим демон, который предоставляет нам *Python*. Если у вас не установлен *Python*, введите следующую команду:

```
sudo apt-get install python
```

Затем введите следующее:

```
python
```

Убедитесь, что все в порядке. Следующим шагом будет запуск сервера. Для этого нам нужно ввести следующее:

```
python -m SimpleHTTPServer 8000
```

Если все прошло хорошо, у нас будет результат, аналогичный следующему, где мы видим, что сервер запущен и работает на порту 8000:

```
~$ python -m SimpleHTTPServer 8000
Serving HTTP on 0.0.0.0 port 8000 ...
_
```

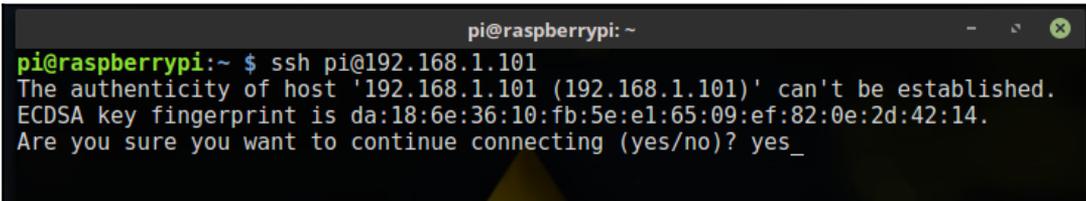
## Подключиться для выполнения команды

В этом разделе мы увидим, как мы можем подключиться к вашему Raspberry Pi Zero W по сети с помощью SSH. SSH означает безопасную оболочку и представляет собой простой способ подключения практически к любому устройству, такому как Raspberry Pi Zero W. Во-первых, мы должны определить IP-адрес нашей малины, как показано ранее. Предполагая, что IP-адрес 192.168.1.101, все, что нам нужно сделать, это использовать ssh-клиент.

В операционных системах Linux и Mac просто откройте терминал и введите следующее:

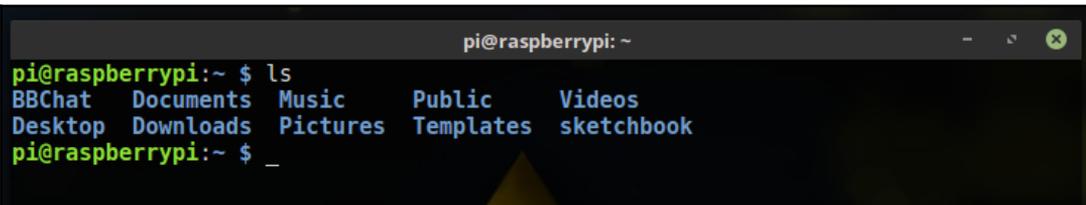
```
ssh <user>@<ip-address>
```

Здесь, <user> is pi и <ip-address> равны 192.168.1.101:



```
pi@raspberrypi: ~
pi@raspberrypi:~ $ ssh pi@192.168.1.101
The authenticity of host '192.168.1.101 (192.168.1.101)' can't be established.
ECDSA key fingerprint is da:18:6e:36:10:fb:5e:e1:65:09:ef:82:0e:2d:42:14.
Are you sure you want to continue connecting (yes/no)? yes_
```

После этого просто введите свой пароль и нажмите Enter:



```
pi@raspberrypi: ~
pi@raspberrypi:~ $ ls
BBChat  Documents  Music      Public     Videos
Desktop Downloads  Pictures   Templates  sketchbook
pi@raspberrypi:~ $ _
```

Теперь вы можете видеть, что bash изменился, и мы находимся в нашем Raspberry Pi Zero W.

## Автоматическая аутентификация

Иногда полезно автоматически аутентифицировать себя через плату Raspberry Pi. Это означает, что нам не будет предложено ввести наш пароль или что-либо еще. Следующий скрипт с именем cr1 (подключение к Raspberry Pi) — полезная команда, позволяющая легко получить доступ к нашей плате через SSH. На самом деле скрипт состоит всего из одной строки. Создайте новый файл с именем cr1 и введите следующую команду:

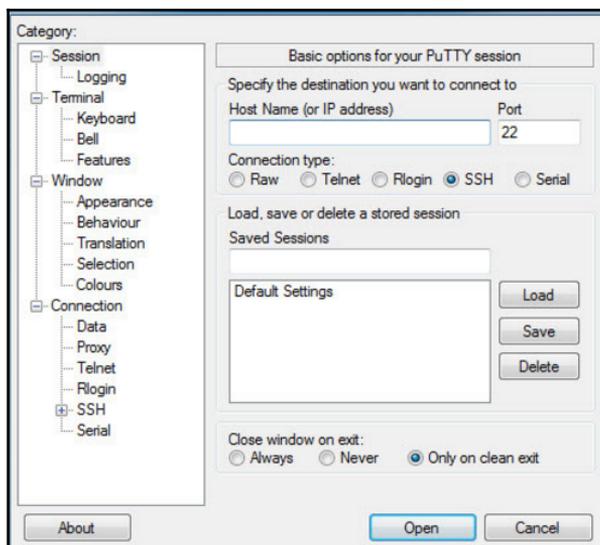
```
ssh pi@<Raspberry Pi Address>
```

Сохраните его в папку /bin с помощью sudo. Далее вам нужно создать rsa keygen и перенести публичный ключ (name.pub) в файл .ssh/authorized\_keys в Raspberry Pi Zero W.

В Windows проще всего это сделать с помощью ssh-клиента, такого как PuTTY. Вы можете бесплатно скачать PuTTY по следующей ссылке:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Обратите внимание, что нет необходимости в установке. Просто откройте исполняемый файл, который вы только что загрузили, и настройте соединение. Вам необходимо указать имя хоста или IP-адрес вашего устройства Raspberry Pi и установить порт 22 для всех соединений SSH (порт по умолчанию). Нажмите «Open - Открыть», после чего вам будет предложено ввести имя пользователя (pi) и пароль (raspberry). После нажатия Enter вы должны легко войти в свое устройство Raspberry Pi:



## Подключение для передачи файлов

Иногда мы против проблемы передачи данных и файлов на Raspberry Pi и с него. Для этого нельзя использовать SSH, поэтому в следующем разделе будут описаны способы, с помощью которых вы можете легко передавать файлы с вашего raspberry pi и на него.

## FileZilla

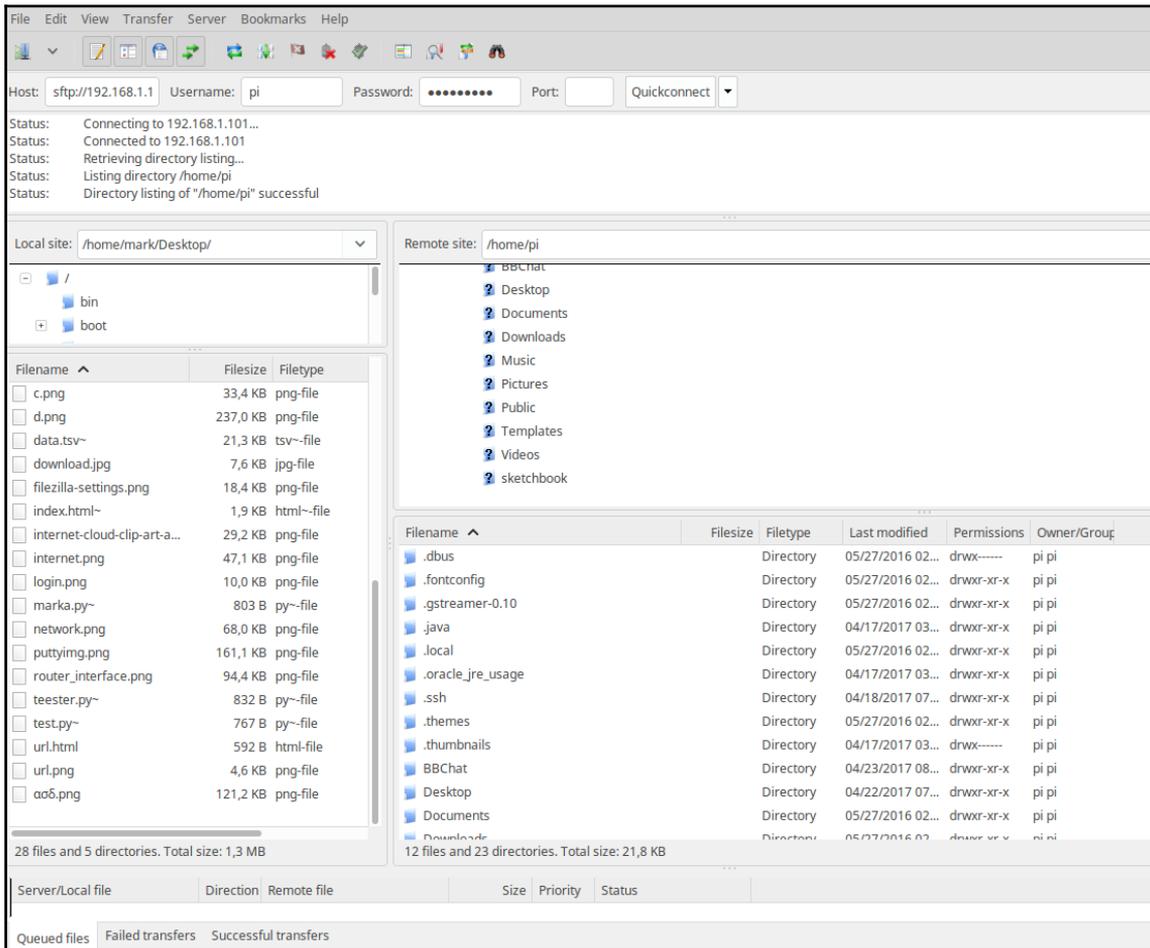
Программу FileZilla можно загрузить как в Windows, так и в Linux, и это довольно просто. Для Windows вы можете загрузить его с <https://filezilla-project.org/download.php>, а для Linux вы можете использовать ту же ссылку или ввести следующую команду:

```
sudo apt-get install filezilla
```

После установки этой программы все, что вам нужно сделать, это ввести IP-адрес Raspberry Pi, порт (по умолчанию 22) и имя пользователя/пароль, которые обычно являются pi/raspberry.

Нажмите « connect - Подключиться», и вы должны увидеть два раздела. Левый раздел — это локальные файлы на вашем компьютере, а правый раздел — удаленные файлы (Raspberry Pi Zero). Просто используйте перетаскивание файлов из одного раздела в другой.

После подключения к плате Raspberry Pi вы должны увидеть что-то вроде этого:

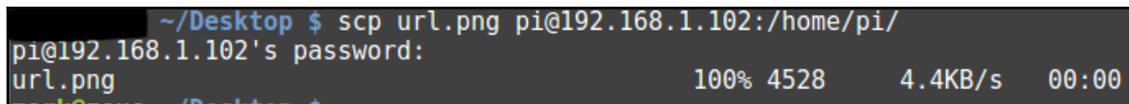


## SCP

В операционных системах Linux есть одна команда под названием `scp`, которая может передавать файлы с удаленного сервера и на него. Это полезно, потому что иногда у нас нет графического пользовательского интерфейса, и `scp` — наш единственный вариант. USB-накопители тоже работают, но мы предполагаем, что передаем файлы по сети. `scp` — это инструмент командной строки, с которым легко работать.

Чтобы передать файл с вашего компьютера на удаленный сервер (Raspberry Pi), введите следующую команду:

```
scp <file-transferred> <user>@<pi-address>:<where to put>
```

A terminal window showing the execution of the scp command. The prompt is ~/Desktop \$ and the command is scp url.png pi@192.168.1.102:/home/pi/. The terminal shows the password prompt, the file name url.png, and progress information: 100% 4528, 4.4KB/s, 00:00.

```
~/Desktop $ scp url.png pi@192.168.1.102:/home/pi/  
pi@192.168.1.102's password:  
url.png 100% 4528 4.4KB/s 00:00
```

Здесь `<user>` — это `pi`, а `<pi-address>` — это IP-адрес Raspberry Pi. Обратите внимание, что между IP-адресом и передаваемым файлом есть `:`. Если вы находитесь внутри папки и хотите скопировать все файлы, замените `<file-transfer>` на `*`.

С другой стороны, для переноса с удаленного сервера (Raspberry Pi) на локальный компьютер введите следующее:

```
scp username@remote:/file/to/send /where/to/put  
scp <user>@<pi-address>:<file-transferred> <where to put>
```

Вы всегда можете узнать больше о команде Linux, используя следующую команду:

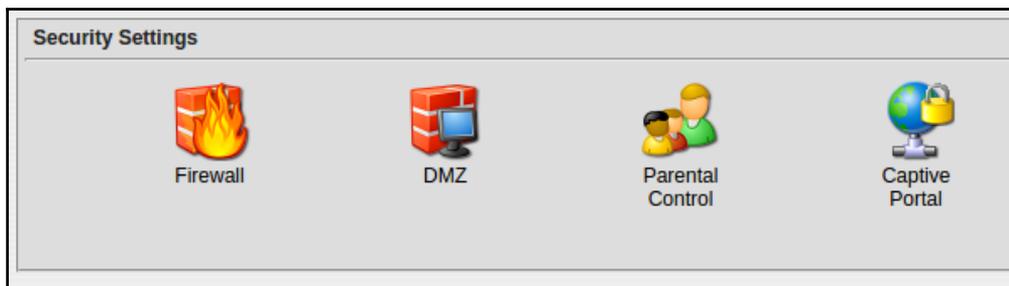
```
man <linux-command>
```

## Защите Raspberry Pi от воздействия

В последнем разделе этой главы мы поговорим о безопасности. Raspberry Pi Zero W великолепен, но никто не хочет просто подключить его к вашему роутеру и позволить другим управлять вашим освещением, вашей кофеваркой и всем остальным, что вы к нему подключили. Начнем с того, что правда в том, что вы не можете быть в безопасности ни от кого. Это означает, что существуют уровни безопасности, которые следует учитывать. На этом этапе вы должны подумать, от кого вы хотите защититься. Например, существуют методы защиты вашей Raspberry от вашей семьи, ваших друзей, кого-либо в вашем доме, обычных пользователей в Интернете, опытных пользователей в Интернете и АНБ.

## Firewall - Брандмауэр

Почти каждый роутер имеет опцию брандмауэра. Брандмауэр — это система, способная фильтровать все входящие соединения и выбирать, подходит ли каждое из них для пропуска в вашу сеть или нет. На следующем снимке экрана вы можете увидеть параметр «Firewall» в настройках моего роутера. Он может отличаться от вашего роутера, но обычно находится на вкладке «Security - Безопасность» в меню:



Кроме того, если мы нажмем «Firewall - Брандмауэр», появится возможность активировать или деактивировать его. Существует также вариант того, насколько сильным будет наш брандмауэр. LOW брандмауэр означает, что он будет почти все пропускать через вашу сеть. HIGH брандмауэр означает, что он почти предупредит и предотвратит входящее соединение через вашу сеть. Если вы думаете, что HIGH всегда лучший вариант, то это не так. Иногда, когда вы играете в онлайн-игры или используете приложения по сети, вам необходимо отключать брандмауэр. Так что имейте в виду, что иногда, в зависимости от вашей сети, вам нужно отключить брандмауэр, и это означает, что да, вы более уязвимы, чем раньше. Так что никогда не показывайте свой IP и не делайте гадостей:



В Windows есть брандмауэр по умолчанию, который вам необходимо отключить при создании программы, связанной с вашей машиной. В заключение никогда ничего не отключайте, если все работает нормально. Я просто упоминаю их в случае каких-либо проблем с сетью и сбоями подключения.

И последнее, но не менее важное: при подключении Raspberry Pi Zero к Интернету начните использовать пароли, состоящие не менее чем из восьми символов, включая специальный символ, цифру и заглавную букву.

## **Резюме**

В этой главе мы рассмотрели некоторые основы работы в сети и подключение к ней своего Raspberry Pi. Мы также говорили о доступе через Интернет и о протоколах IoT.

В следующей главе мы начнем создавать наш первый проект с новой платой Raspberry Pi ZeroW и рассмотрим более практичные вещи, используя все знания из предыдущих глав.

# 3

## Чат-бот

В настоящее время чат-боты очень популярны и используются крупными компаниями, исследователями и всеми, кто хочет учиться и использовать автоматические ответы. Чат-бот — это система, которая может автоматически отвечать на ваши вопросы. Есть умные чат-боты и дамповые. Разница между ними заключается в том, что умный чат-бот хорошо обучен и способен понимать язык и синтаксис, который вы вводите, и отвечает соответствующим образом, в то время как дамповый робот отвечает почти определенным образом в соответствии с вопросом. В этой главе мы создадим простой и удобный чат-бот с помощью Raspberry Pi Zero W. В этой главе мы рассмотрим следующие темы:

- Пользовательский ввод
- Установка сервера
- и сетевые сервисы
- Security - Безопасность

### Пользовательский ввод

В этом разделе мы увидим несколько основных способов, с помощью которых пользователь может отправлять сообщения или разговаривать с ботом. Есть несколько способов, которыми каждый может поговорить со своим ботом, в зависимости от используемой операционной системы. Raspberry Pi Zero W можно использовать как бота или как пользовательское устройство ввода. Raspberry Pi 3 была хорошей платой даже для создания бота Facebook, но эту новую плату Raspberry Pi Zero W не рекомендуется использовать для сложных целей, таких как запуск огромного бота. Итак, то, что мы будем делать в этом проекте, — это простая демонстрация того, как все работает и как мы можем разработать бота с нуля.

В этом разделе мы разработаем веб-страницу для использования в качестве пользовательского устройства ввода. Благодаря этому вы можете легко разработать приложение для Android или систему распознавания речи. Все будет работать нормально, пока программа имеет строку того, что сказал пользователь в качестве вывода. Веб-страница будет разработана в HTML и немного CSS.

## Установка сервера

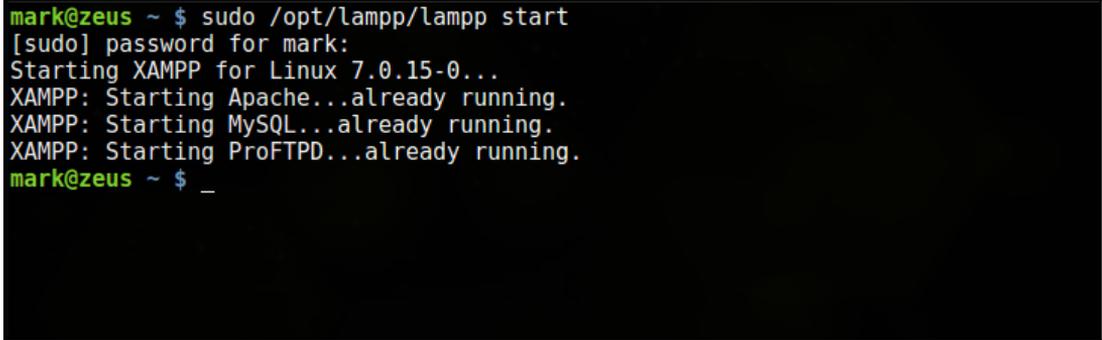
Прежде чем создать сайт, мы должны решить, где мы будем разрабатывать сайт и с помощью каких инструментов. Я рекомендую использовать ваш настольный компьютер или ноутбук, а не Raspberry Pi в качестве основного компьютера для работы. Кроме того, попробуйте использовать текстовый редактор, например Sublime Text, так как они вам очень помогут. Это мой предпочтительный стиль; не стесняйтесь выбирать свой.

Для разработки веб-сайта нам нужен сервер, и мы можем легко использовать сервер XAMPP или LAMPP в системе Windows, Linux или Mac. Загрузите пакет с <https://www.apachefriends.org/download.html>.

После этого установите пакет в свою систему и запустите панель управления. В системах Linux вы можете запустить службы, открыв Terminal и введя следующую команду:

```
sudo /opt/lampp/lampp start
```

Как мы видим на следующем снимке экрана, Apache, MySQL и ProFTPD запустились, так как они ответили уже работающими сообщениями:



```
mark@zeus ~ $ sudo /opt/lampp/lampp start
[sudo] password for mark:
Starting XAMPP for Linux 7.0.15-0...
XAMPP: Starting Apache...already running.
XAMPP: Starting MySQL...already running.
XAMPP: Starting ProFTPD...already running.
mark@zeus ~ $ _
```

Используйте этот код, если вы хотите закрыть тип служб:

```
sudo /opt/lampp/lampp stop
```

На следующем снимке экрана показано, что вы увидите в результате:

```
mark@zeus ~ $ sudo /opt/lampp/lampp stop
Stopping XAMPP for Linux 7.0.15-0...
XAMPP: Stopping Apache...ok.
XAMPP: Stopping MySQL...ok.
XAMPP: Stopping ProFTPD...ok.
mark@zeus ~ $ _
```

Если вы хотите перезапустить, введите следующую команду:

```
sudo /opt/lampp/lampp restart
```

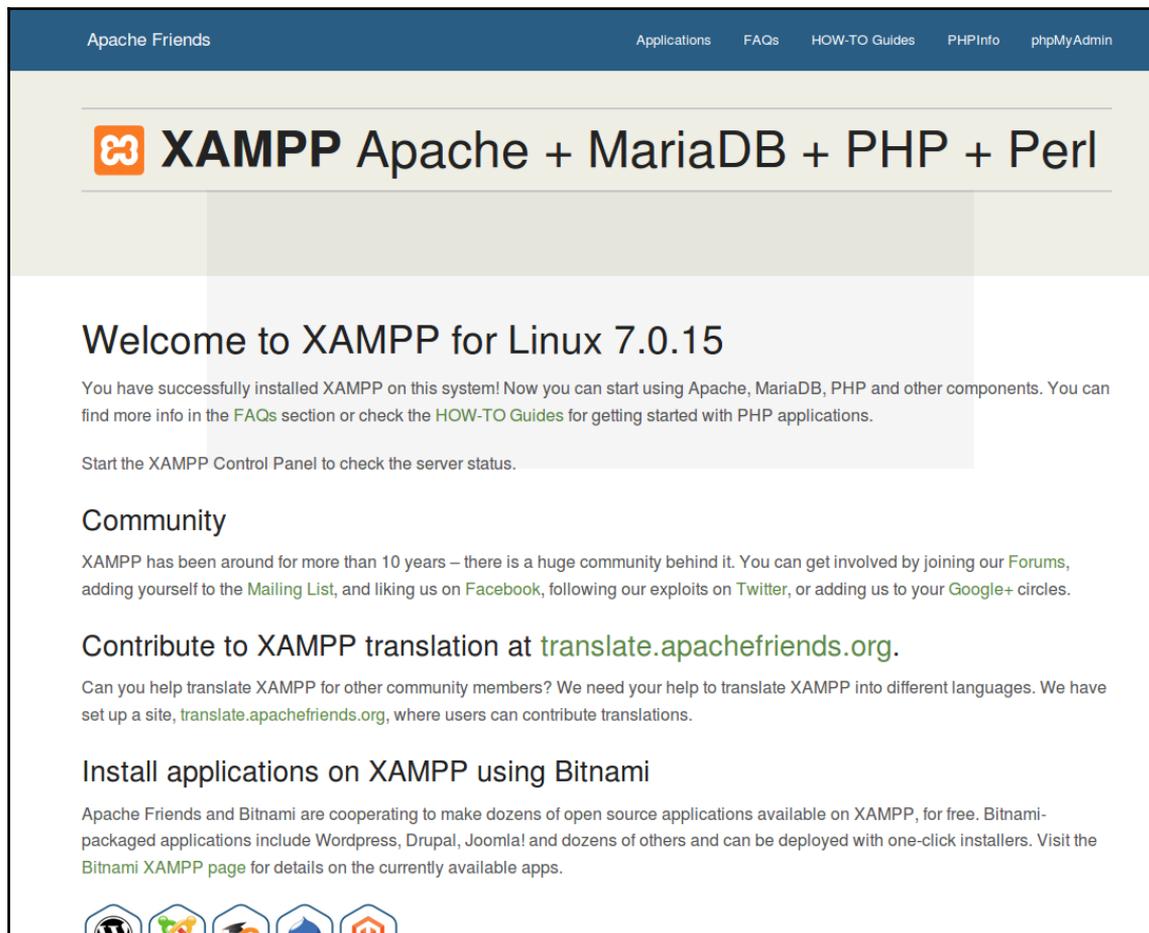
На следующем снимке экрана мы видим, что службы не были запущены, но запустились. Если бы они работали, а не не работали, мы бы получили сообщение ок:

```
mark@zeus ~ $ sudo /opt/lampp/lampp restart
Restarting XAMPP for Linux 7.0.15-0...
XAMPP: Stopping Apache...not running.
XAMPP: Stopping MySQL...not running.
XAMPP: Stopping ProFTPD...not running.
XAMPP: Starting Apache...ok.
XAMPP: Starting MySQL...ok.
XAMPP: Starting ProFTPD...ok.
mark@zeus ~ $ _
```

Обратите внимание, что XAMPP будет использовать порт 80, поэтому рекомендуется закрыть Skype или любую другую программу, работающую на этих портах. После запуска служб XAMPP вы можете без проблем снова открыть Skype или любую другую программу. Общая проблема заключается в том, что они конфликтуют с портами, которые используют тот же порт, что и Skype, или когда мы выполняем потоковую передачу с любой открытой аналогичной программой и используем порт, необходимый для XAMPP.

Если все в порядке, единственное, что вам нужно сделать, это открыть браузер, например Firefox, и ввести <http://localhost/dashboard/>.

Вы увидите веб-сайт, подобный следующему, который является панелью управления сервером XAMPP:



На данный момент нам ясно, что сервер Apache — это то, что нам нужно для написания веб-сайта с использованием кода HTML, CSS и PHP. С другой стороны, служба MySQL — это то, что нам нужно для создания базы данных.

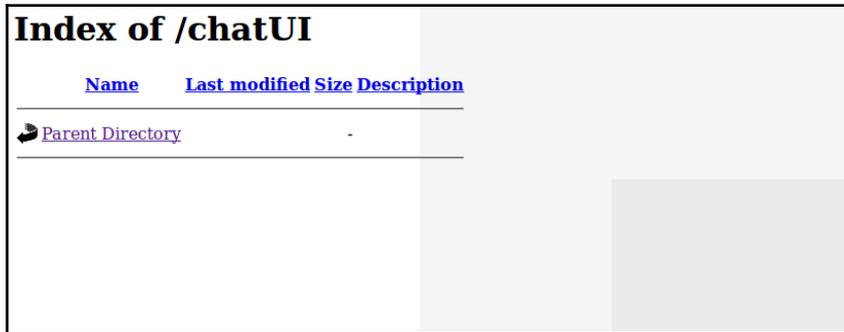
Папка, в которой мы сохраним наш сайт, называется `/opt/lampp/htdocs`. Перейдите в эту папку с помощью следующей команды:

```
cd /opt/lampp/htdocs/
```

Создайте новый каталог с именем chatUI:

```
sudo mkdir chatUI
```

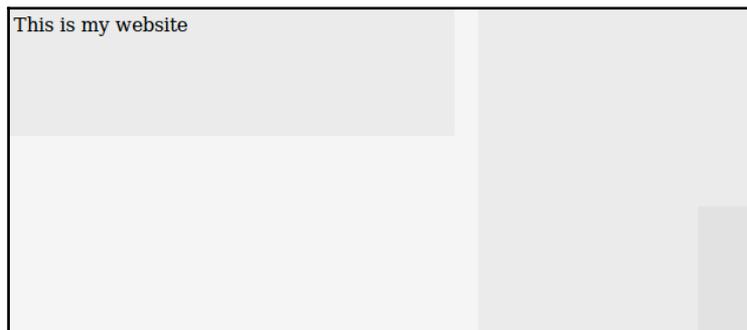
Теперь запустите ls, и у вас должен быть каталог в папке htdocs.  
Подтвердите это, перейдя в браузер и набрав <http://localhost/chatUI/>. Вы должны увидеть что-то похожее на следующий скриншот:



Теперь вы можете видеть, что в каталоге ничего нет, так как в нашем проекте нет файлов. Создайте файл с именем index.php в своем любимом редакторе и используйте следующий код:

```
<html><head></head><body>  
This is my website</body>  
</html>
```

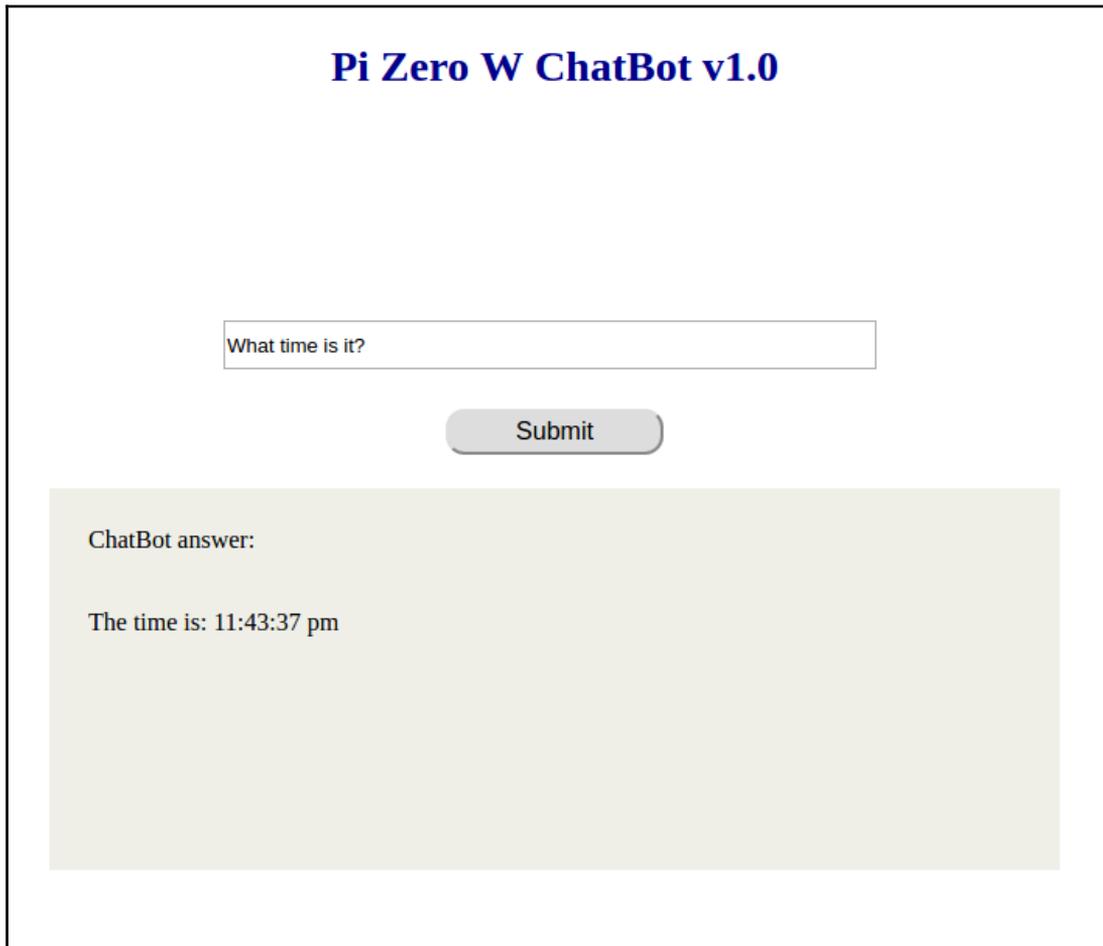
Результат, который вы должны увидеть после перезагрузки сайта, выглядит следующим образом:



Теперь, поскольку эта глава не о веб-разработке, я написал некоторый код, чтобы пользователь мог задать вопрос и получить ответ от сервера.

## Разработка чат-бота

Чат-бот — это бот, который взаимодействует с чем-либо; например, пользователь может отправить запрос, и бот ответит в соответствии с запросом:



**Pi Zero W ChatBot v1.0**

What time is it?

Submit

ChatBot answer:

The time is: 11:43:37 pm

Далее вы найдете код, демонстрирующий этот скрипт. Это веб-сайт нашего чат-бота, использующего язык HTML, CSS и PHP. Давайте пройдемся по коду строка за строкой.

Следующие строки создают заголовок страницы и связывают файл с файлом таблицы стилей CSS. Название можно найти на вкладке браузера на сайте чат-бота. Кроме того, style.css должен находиться в той же папке, что и файл index.php. Если нет, вам нужно добавить полный путь к файлу thestyle.css следующим образом:

```
<title>PI Zero W - Chatbot</title>
<link rel="stylesheet" href="style.css" />
```

Далее нам нужно написать некоторый PHP-код. Сначала мы проверяем, установлен ли пользовательский текст. Это означает, что если пользователь что-то набрал и нажал кнопку, то устанавливается поле usertext. Мы получаем данные с помощью метода \$\_POST, а затем проверяем, является ли пользовательский ввод одним из известных вопросов:

```
<?php
if( isset($_POST['usertext'])) {
    $input = $_POST['usertext'];
    switch($input) {
        case "hello":
            $outdata = "Hello there! What can I do for you?";
            break;
        case "hi":
            $outdata = "Hey! What' up?";
            break;
        case "how are you?":
            $outdata = "I am fine! You?";
            break;
        case "what time is it?":
            $timezone = date_default_timezone_get();
            date_default_timezone_set($timezone);
            $date = date('h:i:s a', time());
            $outdata = "The time is: " . $date;
            break;
        case "who are you?":
            $outdata = "I am your lovely chatbot!";
            break; default:
            $outdata = "Oup's I didn' t get that!";
            break;
    }
}
?>
```

Обратите внимание, что с помощью этого кода мы не создали умного чат-бота или бота с искусственным интеллектом. Для этого нам нужно написать еще много строк кода. В этом разделе мы увидим, как создать простой чат-бот, чтобы вы были знакомы с этой идеей.

Если нам нужен продвинутый чат-бот, мы должны использовать здесь базу данных. Вместо использования оператора `switch` мы должны создать запрос к нашей базе данных и задать вопрос, который задал пользователь. Вы можете прочитать больше о базах данных в следующем разделе этой главы.

Наша часть тела состоит из двух разделов или разделов: заголовочного блока и основного блока. Раздел заголовка — это основной заголовок нашей веб-страницы, а основной раздел — это HTML-форма, которая позволяет пользователю задать вопрос чат-боту. Внутри основного блока у нас есть блок ответов, который на самом деле является невидимым полем, в котором будет отображаться ответ бота. Ничего не печатаем, если вопрос не задан, и печатаем ответ бота, если страница перезагружается и вопрос задан:

```
<body>
  <div id="header">  Pi Zero W ChatBot v1.0
</div>
  <div id="main">
    <form method="POST" action="">
      <input id="userinput" type="text" name="usertext" value="Type
here...">
      <br><br>
      <input id="submitbutton" type="submit" value="Submit">
    </form>
    <div id="answer">
      ChatBot answer:<br><br><br>
      <?php
        if(!isset($outdata)) {
          echo "";
        }else {
          echo $outdata;
        }
      ?>
    </div>
  </div>
</body>
```

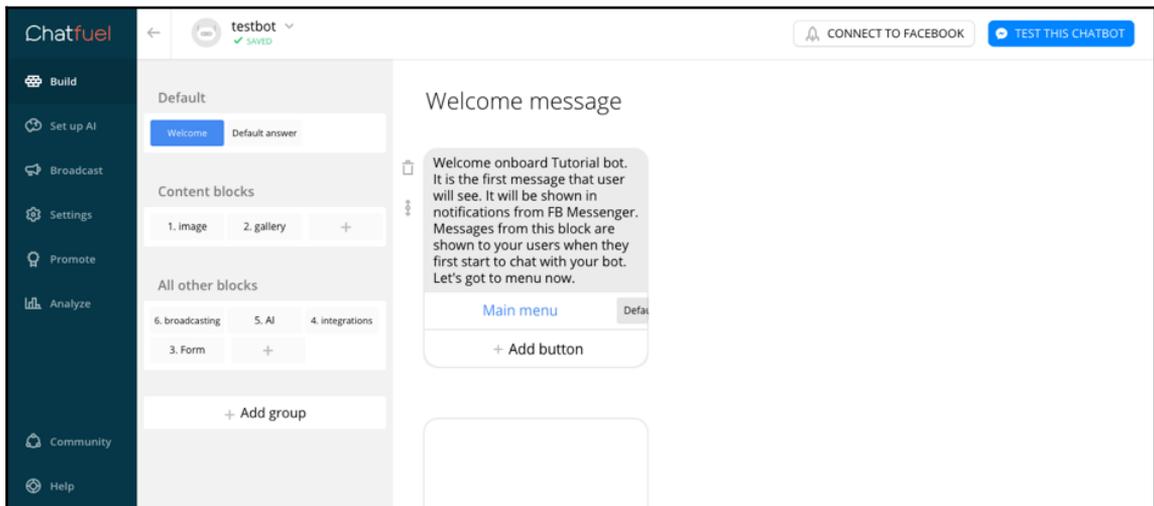
Как вы уже видели, мы создали веб-сайт и загрузили файлы в папку `www` или `htdocs` платы Raspberry Pi. Наш сайт доступен для всех локальных сетей. Это означает, что если вы хотите получить к нему доступ откуда-либо еще, вы не сможете этого сделать. Вы должны создать правило в настройках вашего роутера и перенаправить внешний порт на ваш внутренний порт. Внешний порт может быть любым, например, 2823, но внутренний порт вашего Raspberry Pi должен быть портом 80.

## Услуги чата

В Интернете есть несколько доступных сервисов чат-ботов. Вы можете легко создать чат-бот, зарегистрировавшись на одном из следующих сервисов или скачав любые необходимые библиотеки. Давайте рассмотрим несколько очень интересных чат-ботов, о которых вы должны помнить и держать их в уме для будущих проектов.

## Chatfuel - Чатфуэль

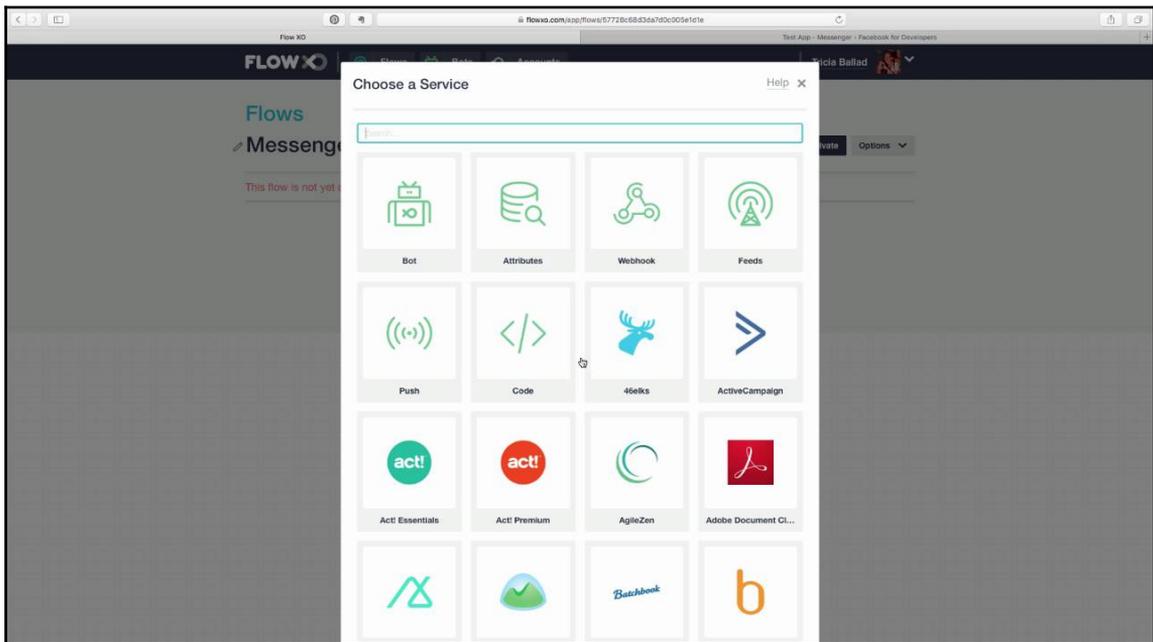
Chatfuel хорош, если у вас есть страница на Facebook. На самом деле это бот для Facebook Messenger и Telegram. Этапы установки очень просты и помогут вам шаг за шагом создать собственного чат-бота без необходимости кодирования. Вы можете отредактировать ответ на любой заданный вопрос, и самое приятное в этом то, что это бесплатно, если вы используете менее 1 00 000 сообщений в месяц. На следующем снимке экрана мы видим интерфейс службы Chatfuel:



Источник: <https://bbvaopen4u.com/>

## Frow XO

Flow XO, с другой стороны, предлагает более широкий набор функций и интеграций. У вас есть визуальный конструктор ботов с графическим пользовательским интерфейсом, такой как Chatfuel (упомянутый ранее), по стандартному плану за 19 долларов в месяц за 15 ботов или активных потоков и 2500 взаимодействий. Он поддерживает множество дополнительных функций, таких как многопользовательский режим, для более чем одного пользователя. На следующем скриншоте мы видим интерфейс, где пользователю предлагается выбрать услугу:



Source: <https://i.ytimg.com>

## Converse

Converse — новая услуга с удивительными функциями, доступная каждому. Это перспективная услуга, с которой вы сможете разговаривать и платить. Эта платформа бесплатна для всех и имеет больше возможностей, чем Facebook Messenger.

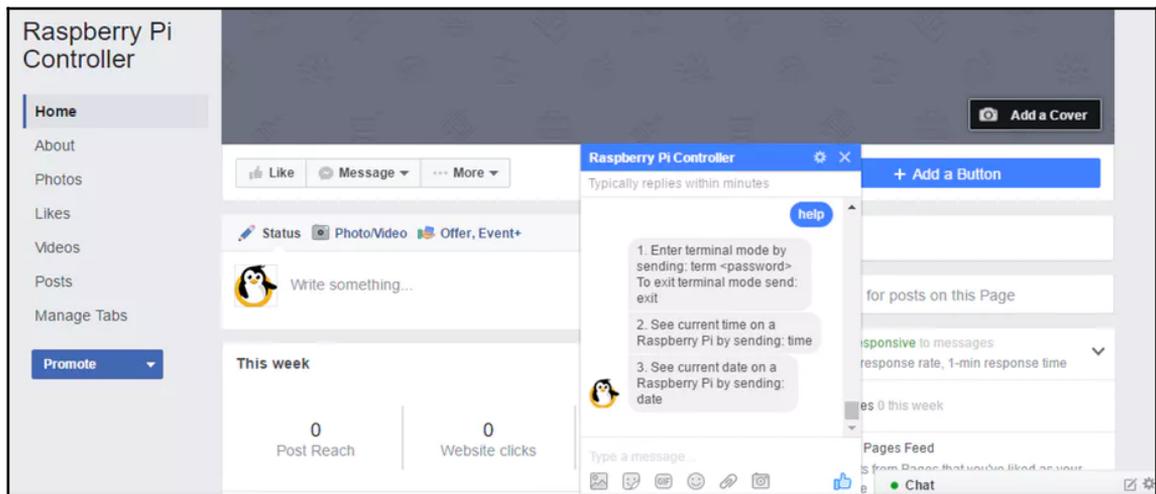
## Мессенджер Facebook

Другой вариант — управлять платой Raspberry Pi с помощью Facebook Messenger. Facebook позволяет вам подключить платформу мессенджера к Raspberry и просто вводить команды или все, что вы хотите, и выполнять их на плате Raspberry Pi. В этом разделе мы упомянем всю идею предыдущего сценария в качестве альтернативного варианта. Пользователь общается через страницу Facebook, созданную для Raspberry Pi, используя Facebook API, с фактической платой Raspberry Pi, которая должна быть подключена к Интернету. В результате у нас есть два требования к этому проекту:

- Плата Raspberry Pi, подключенная к Интернету
- Удаленный терминал Dataplicity с включенной wormhole

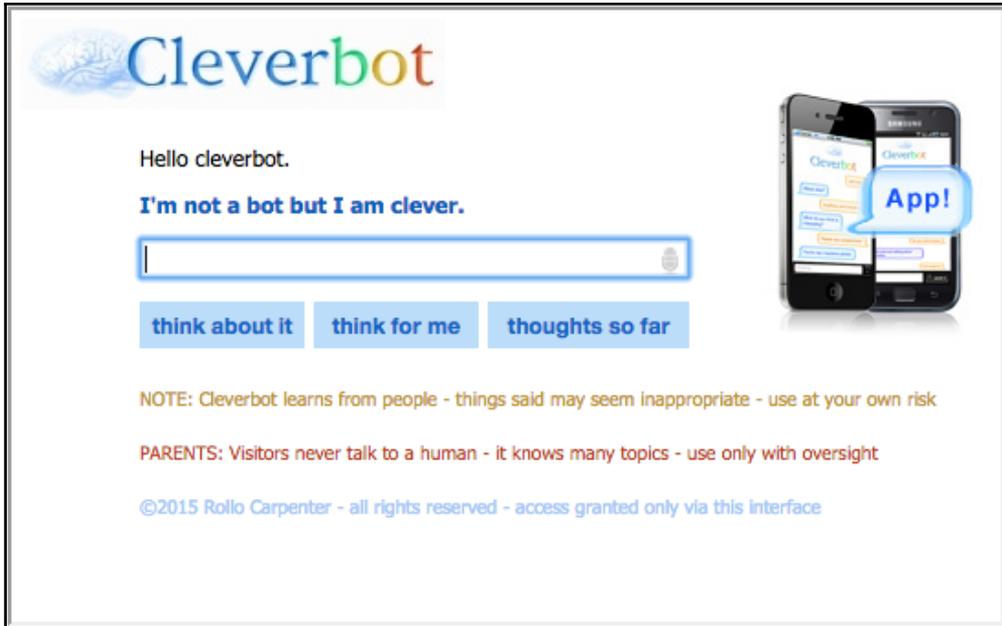
Сначала вам нужно создать аккаунт Facebook и страницу Facebook. Затем перейдите на <https://developers.facebook.com/>, войдите в систему, используя аккаунт администратора, и создайте приложение. Затем вам нужно загрузить соответствующий код на плату Raspberry Pi и подключить ее к WebHook.

Поскольку этот проект не является основной целью главы, если он вам интересен, вы можете следовать инструкциям на странице <https://www.hackster.io/dataplicity/use-facebook-to-control-your-raspberry-pi-terminal-2be038>.



## Умный бот Google

Google в 2012 году разработал библиотеку, обеспечивающую привязку для Cleverbot. Он поддерживает открытие сеанса чата, вопросы и получение ответов. В 2012 году библиотек было не так много, так что это отличная возможность пройтись по старым вещам, над которыми работали разработчики. Дополнительную информацию можно найти на <https://code.google.com/archive/p/pycleverbot/>, как показано на следующем снимке экрана:



Пример использования показан здесь:

```
import cleverbot
mycb=cleverbot.Session()
mycb.Ask("Hello there") 'Hello.'
mycb.Ask('Whats going on?')
'Not much. What is your name?' ````
```

К сожалению, как нетрудно заметить, работа над этим проектом остановилась в 2012 году. Никто не знает, что будет дальше и готовит ли Google что-то еще, но это Google, и почти весь софт Google великолепен. Стоит взглянуть на эту библиотеку.

## Безопасная передача данных

Создавая чат-бот самостоятельно и не используя ни один из сотен доступных сервисов, вы должны помнить о безопасности. Во-первых, при создании программы, в зависимости от того, какую информацию кто-то может извлечь из бота, вам может потребоваться добавить несколько дополнительных уровней безопасности, таких как аутентификация пользователя. Вы можете легко создать или найти в Интернете некоторый код со страницей входа перед доступом к веб-странице ввода пользователя. Во-вторых, рекомендуется создавать больше проверки безопасности в вашем коде. Если у вас есть только оператор `if`, который проверяет, указан ли пароль пользователя, то пользователь может отправить некоторый код и выполнить SQL-инъекцию. SQL-инъекция выполняется на веб-сайтах, которые работают с базой данных. Злоумышленник может легко отправить оператор `SQL` и изменить или получить доступ к таблицам базы данных, даже если у него не должно быть такой возможности. Меры безопасности, такие как предыдущие, имеют решающее значение для обеспечения того, чтобы вы и только вы имели доступ к своему боту.

По сети вы можете использовать программу Wireshark для захвата пакетов и просмотра содержимого каждого пакета. Доступно больше программ, похожего на Wireshark, и все они относятся к категории программ безопасности, называемого сетевыми снифферами. Любой в вашей сети может прослушивать вашу сеть и перехватывать пакеты, которые вы отправляете на плату Raspberry Pi. В результате всегда рекомендуется использовать HTTPS, а не простой HTTP-протокол для своих сервисов, особенно если вы разрабатываете чат-бот и делаете его доступным в Интернете для всех.

## Резюме

В этой главе вы узнали о чат-ботах, о том, как они работают и как взаимодействуют с пользователем. Вы также увидели несколько новых сервисов чат-ботов, доступных для всех, которые вы можете использовать и создавать отличные проекты. В следующей главе мы изменим нашу тему и войдем в мир робототехники. К концу следующей главы вы сможете разработать собственного мобильного робота и использовать Facebook Messenger или любой другой сервис для управления им.

# 4

## Мобильный робот

В этой главе мы создадим и научимся управлять с помощью приложения для Android простым мобильным роботом. Мобильный робот сможет двигаться во всех направлениях и давать обратную связь Android-устройству или компьютеру, к которому он будет подключен.

Почти все роботы следуют одному и тому же принципу. В результате, если вы понимаете, как работает робот, вы можете легко настроить своего робота на все, что захотите: создать еще одного, развить старого и построить целый мир с помощью роботов. В этой главе мы рассмотрим все необходимые основы для создания вашего первого беспроводного робота с Raspberry Pi Zero W. В заключение мы рассмотрим следующие темы:

- Основы робототехники
- Обзор компонентов
- Программирование контроллера
- Будущие идеи

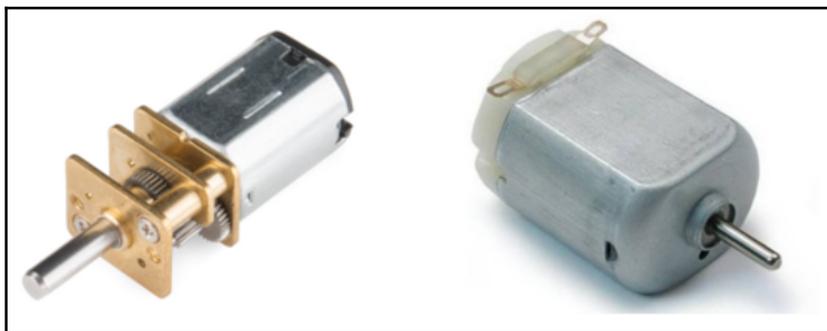
### Основы робототехники

Мобильный робот относится к категории колесных роботов и обычно оснащен колесами или гусеницами. Для простоты в этой главе мы разработаем двухколесного мобильного робота. Фактически у нашего робота будет три колеса, но третье колесо будет нейтральным и будет использоваться для общего баланса робота. Таким образом, фактическими приводами являются два передних двигателя, поэтому мы называем его двухколесным роботом.

Любой механизм робота состоит из приводов и звеньев. Актуатор — это двигатель или что-то еще, что создает тягу и крутящий момент. Связь, с другой стороны, является аппаратной частью, которая связывает привод с шасси. Основной принцип очень прост: внутри каждого робота есть моторы и звенья. В этом проекте мы построим двухколесного мобильного робота, состоящего из двух приводов. Другими словами, наши актуаторы представляют собой простые двигатели постоянного тока, которые создают необходимый крутящий момент для вращения колеса и перемещения робота. Каждый простой привод соединяется винтами или двойной лентой с шасси, описанным ниже. Наше третье колесо является нейтральным и повторяет общую горизонтальную тягу и вертикальный поворот приводов.

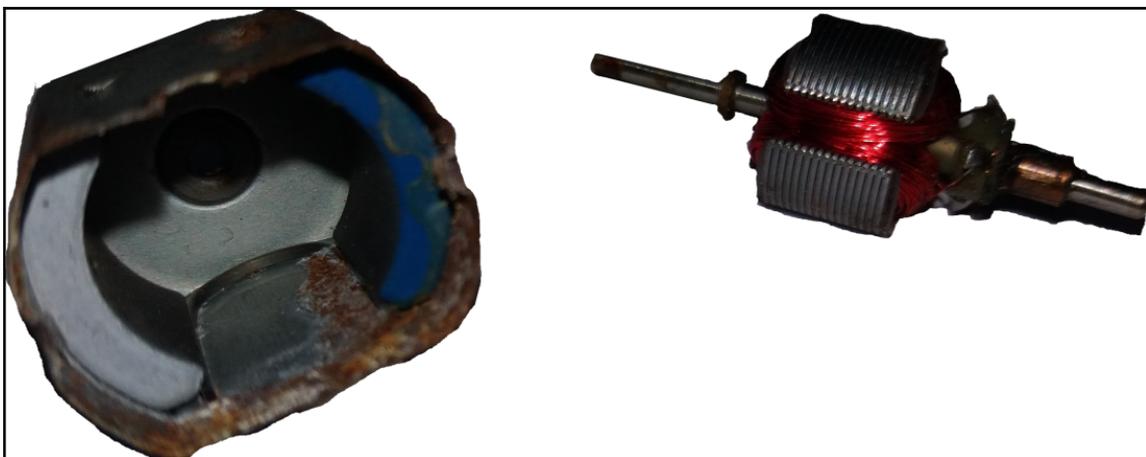
При сборке робота своими руками важно определить причину. Если вы хотите просто построить робота и поиграть с ним, что здорово и весело, вам не нужно заботиться о багах и ошибках. Однако, если вы хотите построить робота для исследовательских целей, например, для вашего университета, вам нужно тщательно выбирать оборудование. Мы поговорим об этом позже, но важно иметь в виду общий бюджет и конечную цель. Для простоты в этом проекте мы рассмотрим простого и дешевого мобильного робота, который будет выдавать ошибки, как вы увидите позже. Компоненты профессиональной робототехники дороги, но минимизируют общую системную ошибку.

Кроме того, если вы хотите построить робота для исследовательских целей, имейте в виду, что вам может понадобиться купить оборудование с максимально возможным количеством спецификаций. Покупка двигателя, о напряжении, частоте вращения, внутреннем сопротивлении и всех других характеристиках которого вы не знаете, только усложнит вашу работу. Всегда есть возможность протестировать и найти то, что вы хотите, но зачем беспокоиться, когда вы можете выбрать другой двигатель со всей необходимой информацией. С учетом сказанного пример с двумя двигателями развеет все ваши опасения. Правый двигатель постоянного тока дешевый и стоит 2,50 доллара, а левый двигатель постоянного тока немного дороже и стоит 14 долларов.



## Двигатели постоянного тока

Двигатель постоянного тока в основном состоит из катушек и шестерен. Но чтобы понять, что на самом деле представляет собой двигатель постоянного тока и как он работает, нам нужно заглянуть внутрь него и понять лежащую в его основе физику. Два магнита с противоположной полярностью притягиваются друг к другу, но с одинаковой полярностью противодействуют друг другу. На рынке есть различные двигатели постоянного тока, но чтобы понять идею создания магического крутящего момента, мы рассмотрим простой двигатель постоянного тока, такой как двигатель, который мы будем использовать позже. На следующем изображении показан простой двигатель постоянного тока в разобранном виде:

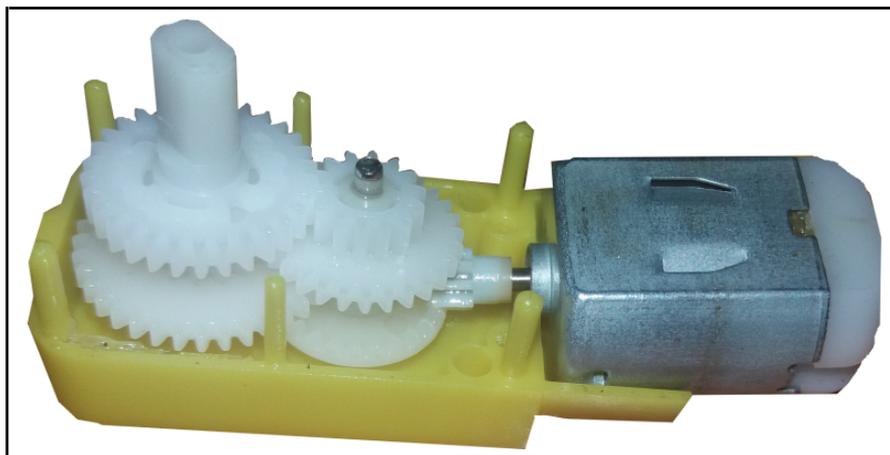


В левой части изображения мы видим внешнюю часть двигателя, где есть два магнита с разной полярностью. В правой части изображения мы видим вал, обмотанный катушкой. Подача напряжения на катушку заставляет ее действовать как небольшой магнит. Этот магнит, созданный катушкой, противостоит или притягивает фиксированный магнит на внешней части. В итоге имеем финальное вращение вала. Обычно мы добавляем к валу какие-то шестерни или подобные компоненты и контролируем точное количество оборотов в минуту (об/мин), которое нам нужно.

Чем больше ток на входе катушки, тем сильнее катушка-магнит. Чем мощнее катушка-магнит, тем больше число оборотов на валу.

## Крутящий момент

Говоря о двигателях постоянного тока и мобильных роботах, обычно нам нужно управлять роботом с низким числом оборотов в минуту. Другими словами, нам не нужен вал, который вращается со скоростью 10 000 об/мин. Обратите внимание, что реальный двигатель может вращаться со скоростью 10 000 об/мин, но нам нужен способ уменьшить это число оборотов в последней точке крепления колеса. Итак, когда мы поместим колеса или любой другой подобный компонент в двигатель, у нас будет нормальная и контролируемая скорость. Это отличается от бесколлекторных двигателей, которые используют квадрокоптеры. В квадрокоптерах и других летательных аппаратах нам нужен не крутящий момент, а скорость. Один простой способ уменьшить фактическую скорость вращения двигателя — использовать шестерни. На следующем изображении показана внутренняя часть двигателя, который будет использоваться в этом проекте:



Мы не будем проходить через геометрию и все теории о зубчатых колесах, которые открыли древние греки. Однако важно понимать, что шестерня снижает скорость вращения двигателя. Например, когда вы едете на велосипеде или ведете машину, у вас есть возможность изменить скорость с помощью соответствующего компонента. Вы на самом деле переключаете передачи, и это легко увидеть на вашем велосипеде. Простой двигатель постоянного тока, подобный показанному на предыдущем изображении, работает по тому же принципу.

## Колесо

Колесо или колеса вашего мобильного робота очень важны. Колесо обычно состоит из резины, чтобы иметь большое сцепление с землей и вращаться, не скользя по земле. На рынке есть два качества каучука. Обычно дешевые колеса сделаны из резины, которая довольно трудно гнется и немного скользит. Другой резиновый материал - более настоящая резина, и он неплохо подходит для такого рода или роботов. Она не скользит по земле, даже если материал - стекло или плитка. К сожалению, невозможно на глаз понять, насколько хороша резина. Так что имейте это в виду, когда будете покупать колеса. Оба колеса, описанные ранее, имеют отверстие в форме буквы В, и оно используется для соединения колеса с двигателем. Вы можете увидеть это на следующем изображении:



Дешевый вариант здесь — использовать пластиковые мотор-редукторы, пластиковые колеса и пластиковые соединения. Несложно понять, что пластиковые вещи дешевле железных. Конечно, они сделают свою работу, но если у вас хороший бюджет, выбирайте железо. Пример показан на следующем изображении:



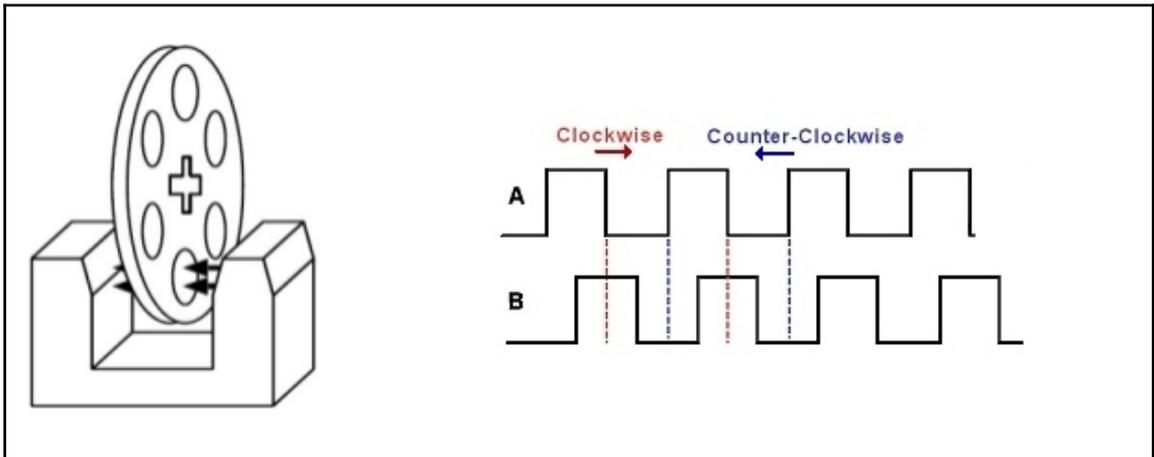
Очевидно, что вы не увидите огромной разницы в общем поведении мобильного робота. Тем не менее, он будет более стабильным, без большого количества ошибок или необходимости балансировать ваши мотор-колеса и тому подобное, что на самом деле является дополнительной проблемой в общем проекте.

## **Енкодеры**

Как упоминалось ранее, важно иметь скорость двигателя в каждый момент времени. Таким образом, нам нужен компонент, который фактически информирует нас об этом. Этот компонент называется энкодером. Энкодер может быть внешним или внутренним по отношению к двигателю. Энкодер — это система, которая знает, сколько оборотов в минуту сделал двигатель, и передает эти данные контроллеру. Важно иметь как можно больше подробностей. Есть энкодеры с 500, 1000 или 2000 и более отверстиями в диске и естественно чем больше отверстий тем лучше результат и меньше системных ошибок. Энкодер — это черный диск, установленный с другой стороны двигателя, как вы можете видеть на следующем изображении:



По мере того, как вращается двигатель, вращается диск, и единственное, чего не хватает, — это системы, способной понять, сколько отверстий прошло с начальной точки. Эта система представляет собой микросхему энкодера, которая показана на следующем рисунке, и мы вскоре увидим, как она работает:



Луч света переносится с одной стороны на другую, как показано стрелками. Правый край является передатчиком светового луча, а левый - приемником. Вы, очевидно, понимаете, что только когда между ними проходит отверстие, приемник может принять луч. В результате мы получаем что-то вроде меандра в правой части предыдущего рисунка. Приемник принимает прямоугольный сигнал. Вот как работает кодировщик, и в разделе программирования главы мы увидим, как преобразовать эти 0 и 1 во что-то более полезное. На следующем изображении показан двигатель с внутренним энкодером:



К настоящему времени мы рассмотрели достаточно двигателей постоянного тока и то, как работают приводы. Далее мы рассмотрим компоненты, необходимые для создания простого двухколесного мобильного робота. Мы пройдемся по каждому компоненту и опишем, что это такое и как оно работает.

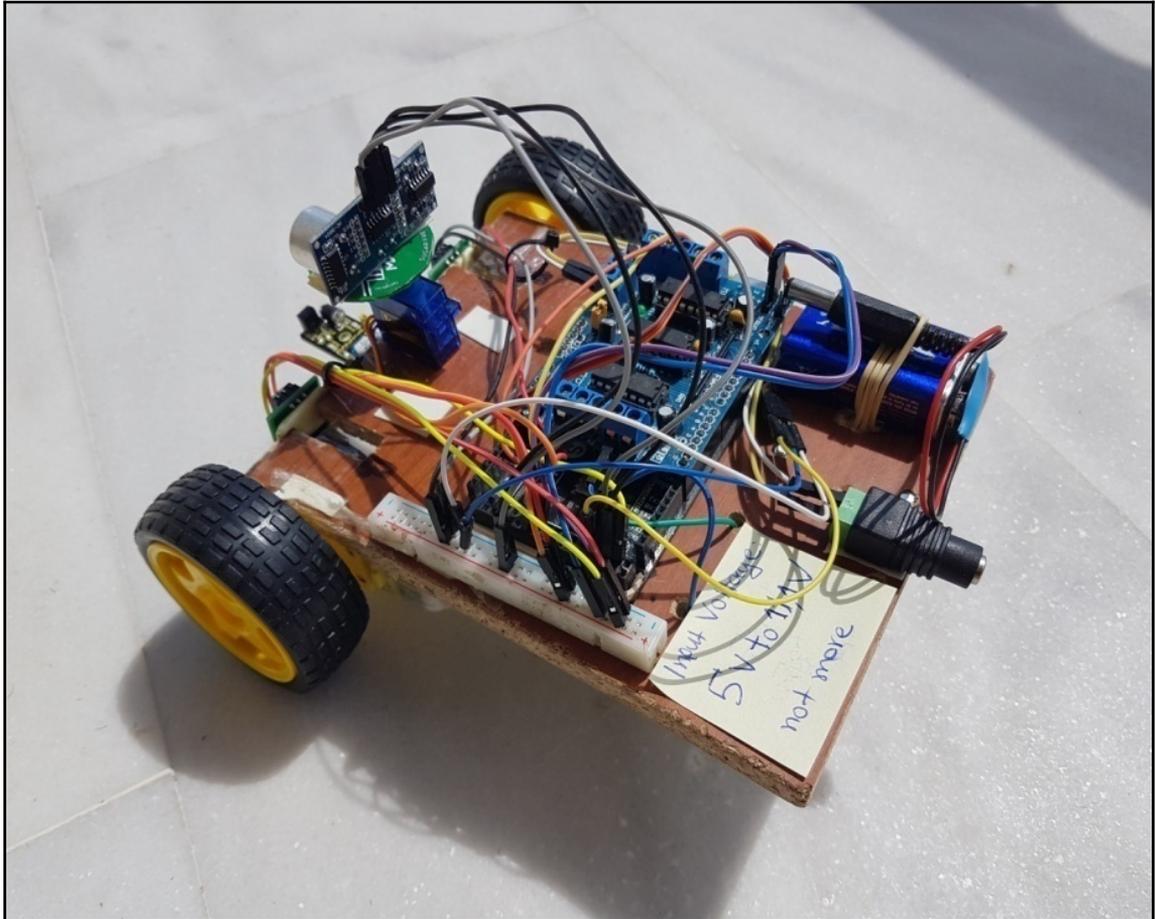
## **Обзор компонентов**

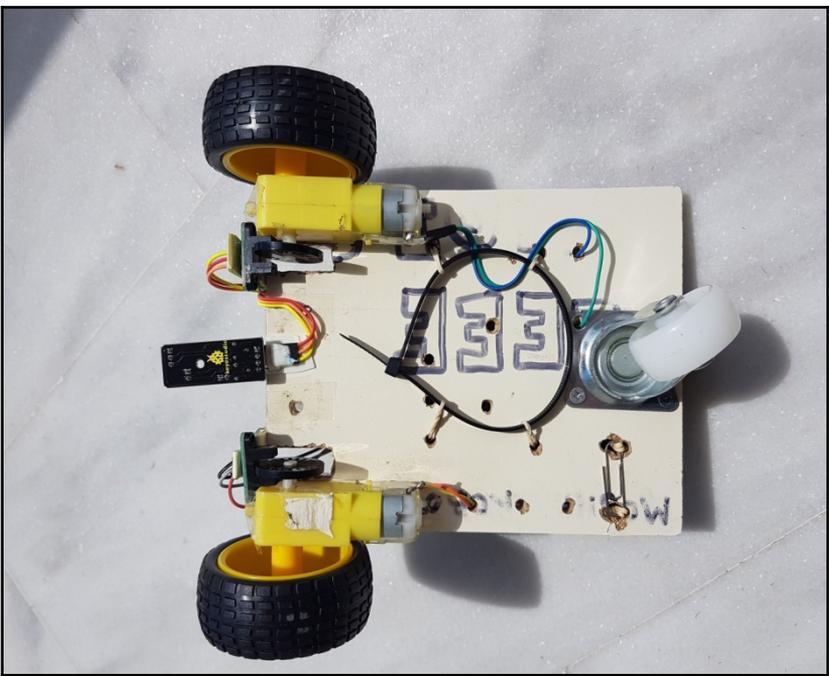
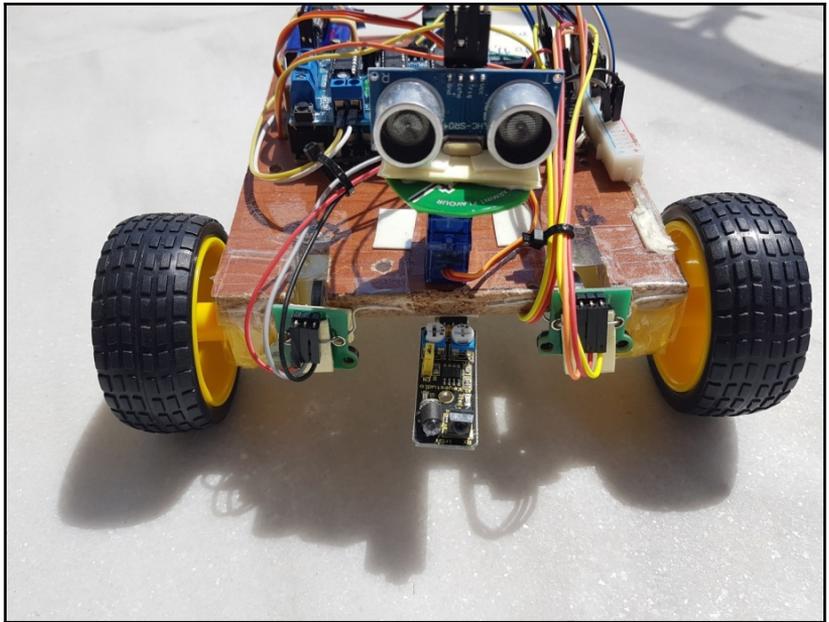
В этом разделе мы рассмотрим все необходимые компоненты. Конечно, вы можете использовать все, что угодно, вместо упомянутых здесь, но я думаю, что этот список достаточно дешев для создания вашего первого робота. Подводя итог, мы создадим двухколесного робота с энкодерами, используя Arduino Mega в качестве платы контроллера и, наконец, Bluetooth для подключения к нашему приложению для Android:

- Двигатель постоянного тока и колеса
- Датчики двигателя
- Arduino
- Мотор шилд
- Сервопривод
- Ультразвуковой датчик расстояния
- Bluetooth

- Макетная плата
- Батарея
- Другие компоненты

Последний робот, которого мы сделаем, выглядит следующим образом:





## Двигатель и колеса

Как мы уже описывали, двигатель — это исполнительный механизм, компонент, который обеспечивает скорость транспортного средства. Мы будем использовать простые двигатели постоянного тока, которые достаточно дешевы и их легко найти на рынке. Используемые двигатели постоянного тока управляются от 3 В до 11 В и выглядят следующим образом:



У них есть ось, в которой одна сторона для колеса, а другая для энкодера. Если вы найдете двигатели с уже припаянными выводами, отдайте предпочтение им, потому что выводы двигателя можно легко повредить. Нам понадобятся два двигателя, как показано на предыдущем изображении.

Колеса, используемые для этого проекта поставляются с вилкой Dupont:



Диаметр колеса 66 мм \* 26 мм. Нам понадобится по одному колесу на каждый мотор.

## Енкодер

Далее кодировщик, который мы будем использовать, довольно распространен и его можно найти по следующей ссылке:

<http://www.ebay.com/itm/Wheel-Encoder-Kit-For-Robot-Car-/221245494078?hash=item338342533e>. Энкодер поставляется с диском и чипом, который выполняет работу со световым лучом, описанную ранее.



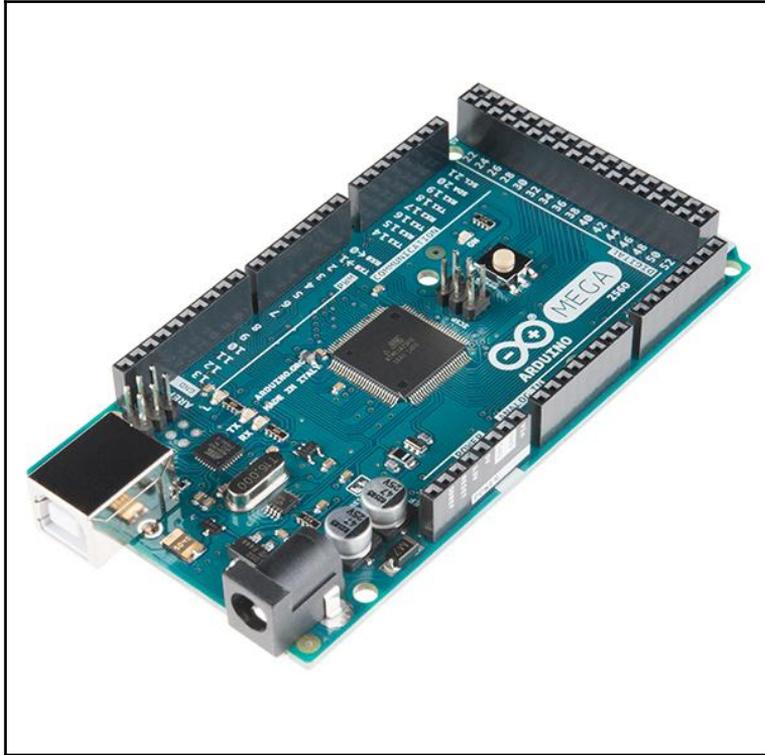
Далее вы можете увидеть электронную часть энкодера. Этот черный пластик содержит датчик, который активирует луч от одного края до другого. Если между лучом нет преграды, он достигает места назначения, и мы называем его ВЫСОКИМ или 1, при преграде - мы называем его НИЗКИМ или 0. По мере движения мобильного робота вырабатывается прямоугольный сигнал - меандр:



## **Ардуино микроконтроллер**

Следующим компонентом, который нам понадобится, является микроконтроллер Arduino. Обычно люди используют Arduino UNO для такого рода проектов, но мы будем использовать Arduino Mega, потому что позже нам нужно будет использовать наш ультразвуковой датчик расстояния и другие датчики. Так что найдите Arduino Mega на <http://www.ebay.com/> или в любом другом магазине, который вы предпочитаете.

Ваш Arduino должен выглядеть так:



## Моторшилд

Так как Arduino Mega работает от 5В, она не может управлять нашими двигателями и развить их максимальную скорость. Таким образом, нам нужен шилд под названием Motor Shield от Adafruit, который мы подключим к нашему Arduino и будем использовать его библиотеки для управления моторами, как мы увидим в теме программирования в этой главе. Шилд, который мы будем использовать, выглядит следующим образом:



При необходимости он может управлять 4 двигателями DC и двумя сервоприводами. Для этого проекта мы будем управлять двумя двигателями, а сервопривод будет управляться напрямую с мегалплаты Arduino.

## Сервопривод

Сервомеханизм — это что-то вроде мотора, но вместо того, чтобы вращаться, он может поворачиваться только на 180 градусов. Представьте себе этот механизм как систему, которая поворачивается от 0 до 180 градусов, а затем обратно до 0. Он также имеет возможность оставаться под определенным углом, где простой двигатель постоянного тока не может остановиться, он продолжает вращаться, пока мы не отключим питание. Простой сервопривод, подобный тому, который мы будем использовать в нашем проекте, показан на следующем снимке экрана:



Он имеет три провода, первый для питания 5В, второй для земли и последний для сигнала, который определяет положение сервопривода. Следующий код поворачивает сервопривод от 0 до 180 градусов и обратно до 0. Чтобы использовать следующий код с микроконтроллером Arduino, нам нужна библиотека Servo.h.

Во-первых, нам нужно включить библиотеку в наш скетч:

```
#include <Servo.h>
```

Затем определите переменную Servo:

```
Servo myServo;
```

А затем в функции `setup()` иницилируйте сервопривод и определите контакт, к которому подключен наш сервопривод. Мы можем сделать это с помощью следующего кода:

```
myServo.attach(4); // Предположим, что мы подключили сервопривод в 4 цифровых пинга.
```

Затем в функции `loop()` мы должны повернуть сервопривод от 0 до 180 и обратно до 0. Для этого мы будем использовать цикл `for` и записывать каждую позицию от 0 до 180:

```
for(int i=0; i<180; i++) {  
    myServo.write(i);  
}  
for(int i=180; i>0; i--) {  
    myServo.write(i);  
}
```

## Ультразвуковой датчик

Мы использовали ультразвуковой датчик расстояния для измерения расстояния между мобильным роботом и всеми объектами в диапазоне 180 градусов перед роботом. Ультразвуковой датчик используется с сервомеханизмом, как описано далее, что помогает сканировать большую площадь. Используемый датчик показан на следующем снимке экрана:



Как видите, у него есть два круга, которые мы называем триггерным кругом и эхо-кругом соответственно. Триггерный круг отбрасывает ультразвуковой луч и, как только он попадает в какой-либо твердый предмет, возвращается и захватывается эхо-кругом. Мы не будем вдаваться в детали и геометрию этого процесса. Однако важно понимать, что существует период времени, в течение которого ультразвук проходит от датчиков и возвращается. Эта продолжительность делится на число, и мы можем получить расстояние до объекта. Далее вы найдете код, который нам нужен для всего этого процесса с использованием контроллера Arduino.

Сначала мы должны определить контакты, к которым подключены триггерный и эхо-вывод. Давайте предположим, что мы подключили вывод эха к цифровому выводу 7 Arduino, а вывод триггера — к выводу 8 Arduino:

```
#define echoPin 7 // Echo Pin
#define trigPin 8 // Trigger Pin
```

В функции `setup()` мы должны определить, что это за пины. Итак, мы определяем триггерный контакт как выходной контакт, а эхо-контакт — как входной. Причина этого проста. Триггерный контакт, как объяснялось ранее, посылает ультразвуковой сигнал от датчика в окружающую среду; таким образом, это выход, где эхо получило (или не получило) звук, поэтому мы можем назвать его нашим входом:

```
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
```

В цикле вы можете создать функцию с именем `getDistance()` и добавить следующую реализацию. Мы должны создать импульс с двухсекундным НИЗКИМ, а затем ВЫСОКИМ в течение 10 секунд. С помощью этого импульса мы можем получить продолжительность, и если мы разделим ее на 58,2, мы сможем получить расстояние от ультразвукового датчика до ближайшего найденного объекта или обнулить, если впереди ничего нет:

```
digitalWrite(trigPin, LOW);delayMicroseconds(2);digitalWrite(trigPin,
HIGH);delayMicroseconds(10);digitalWrite(trigPin, LOW);duration = pulseIn
(echoPin, HIGH);
```

```
//Рассчитайте расстояние (в см) на основе скорости звука.distance = duration/58.2;
```

Наконец, в вашей функции вы можете либо иметь глобальную переменную с именем расстояния, либо возвращать расстояние с помощью следующей строки кода:

```
return distance;
```

Теперь у вас должна быть функция, которая отлично работает с вашим сонаром и определяет расстояние. Далее вы найдете полный код для проверки ваших подключений. Возможно, вам придется изменить контакты, определенные в начальных строках:

```
#define echoPin 24 // Эхо Pin
#define trigPin 26 // Trigger Pin
#define LEDPin 13 // встроенный LED

int maximumRange = 200; // Необходим максимальный диапазон
int minimumRange = 0; // Необходим минимальный диапазон
long duration, distance; // Длительность, используемая для расчета расстояния
void setup() { Serial.begin (9600); pinMode(trigPin, OUTPUT); pinMode
(echoPin, INPUT);
  pinMode(LEDPin, OUTPUT); // Use LED indicator (if required)
}

void loop() {
/* Следующий цикл trigPin/echoPin используется для определения расстояния до
ближайшего объекта путем отражения от него звуковых волн.*/

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);

  /digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);

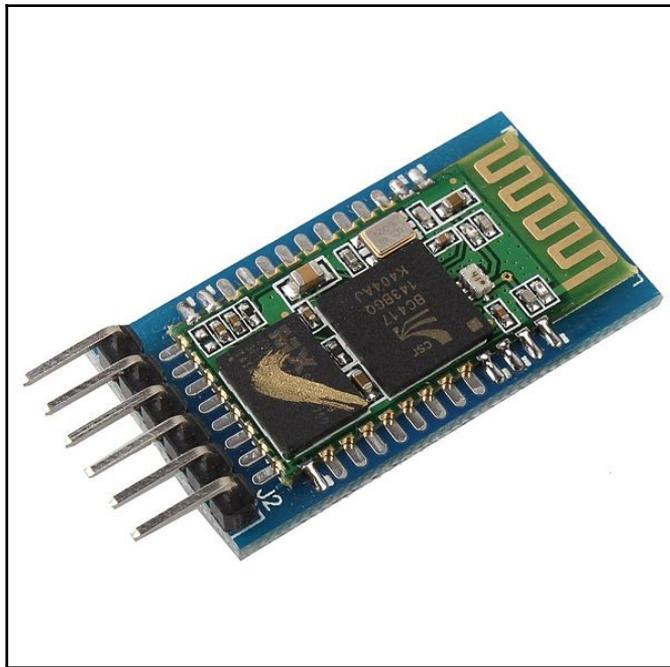
  //Рассчитайте расстояние (в см) на основе скорости звука.
  distance = duration/58.2;

  if (distance >= maximumRange || distance <= minimumRange){
  /* Отправьте отрицательное число на компьютер и включите светодиод, чтобы указать «вне
диапазона»*/
  Serial.println("-1");
  digitalWrite(LEDPin, HIGH);
  }
  else {
  /* Отправьте расстояние на компьютер, используя последовательный протокол, и
выключите светодиод, чтобы указать на успешное чтение. */
  Serial.println(distance);
  digitalWrite(LEDPin, LOW);
  }
  //Задержка 50 мс перед следующим чтением.

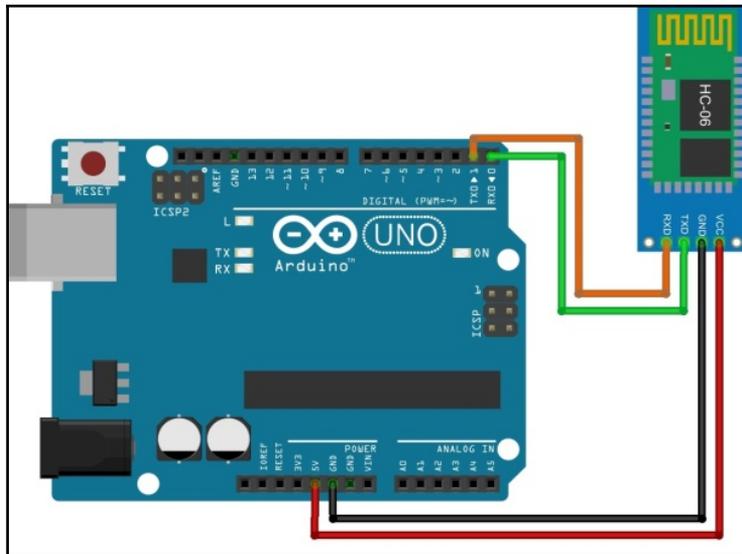
  delay(50);
}
```

## Bluetooth

Датчик Bluetooth — это устройство, которое взаимодействует с контактами TX/RX Arduino и, конечно же, получает данные с использованием протокола связи Bluetooth. В противовес всем предыдущим датчикам, очень легко обжечься, просто подключив не те провода. Датчик Bluetooth, который мы будем использовать, называется HC06 и выглядит следующим образом:



Обычно он состоит из шести пинов, но мы используем только четыре из них. Если вы посмотрите ближе на распиновку, вы увидите, что настоящие контакты (контакты по краям не используются) — это 3,3 В, GND, TX и RX. Обратите внимание, что если вы подключите GND к питанию Arduino, он мгновенно сгорит. Кроме того, если по какой-либо причине контакт 5V или контакт GND коснется области Arduino, которая не должна касаться, он также может сгореть. Так что разместите его как можно дальше от Arduino, не проверяйте распиновку соединений и, если возможно, подключите его к макетной плате, чтобы он был в безопасности. Очевидно, что питание составляет 3,3 В или 5 В. На следующем снимке экрана показано, как подключить модуль Bluetooth к Arduino:



Для тех, кто задается вопросом, почему вы должны использовать модуль Bluetooth с Arduino, а не просто использовать Bluetooth платы Raspberry Pi, ну, если вы используете Bluetooth платы Raspberry Pi, то вы не сможете полностью использовать контакты по умолчанию. .

Следующий код можно использовать для проверки ваших подключений; вы можете прочитать из Arduino, что получает серийный номер:

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(0,1);
String Data = "";

void setup()
{
  Serial.begin(9600);
  mySerial.begin(9600);
  //Serial.println("Hello world");
}

void loop() // бегать снова и снова
{
  while (mySerial.available() > 0)
  {
    //Serial.println("Something is available");
    char character = mySerial.read(); // Получить один символ из монитора порта программы

    Data.concat(character); // Добавить полученный символ в приемный буфер
  }
}
```

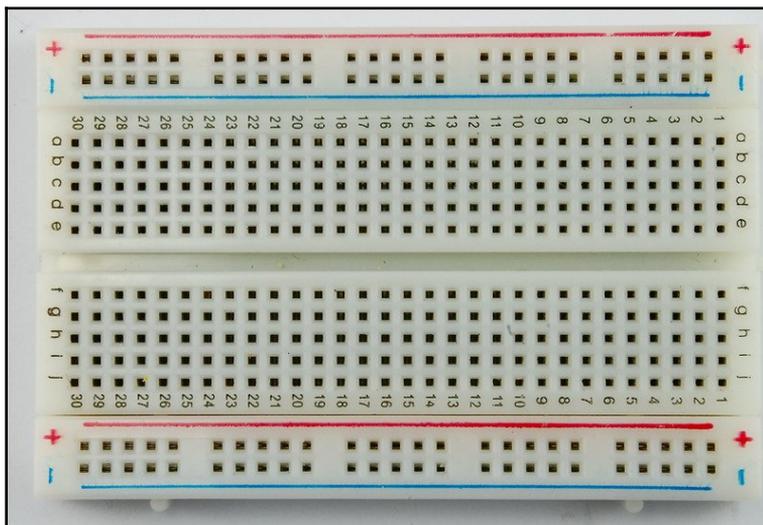
```
if (character == '\n')
{
    //Serial.print("Received: ");
    Serial.println(Data);

    // Добавьте свой код для разбора полученной строки здесь...

    // Очистите буфер приема, чтобы мы были готовы принять следующий
line
    Data = "";
}
}
```

## Макетная плата

Макетная плата — это простой компонент, который позволяет нам отлаживать вашего робота и подключать больше датчиков. Альтернативный способ — припаять провода, но я предпочитаю их втыкать, подключать к контроллеру, а затем тестировать робота. Как только я дохожу до своего финального робота, то снимаю все соединения и спаиваю все вместе. Простая макетная плата показана на следующем снимке экрана:



## Батарея

Поскольку мы создаем мобильного робота, у нас есть много вариантов. Мы можем использовать LiPo, NiHm или даже NiCD аккумуляторы. Преимущество LiPo аккумулятора перед другими технологиями заключается в его весе. Это причина того, что мы выбрали этот тип батареи. Вес является решающим фактором всего проекта, поскольку он влияет на продолжительность работы нашего робота. Аккумулятор LiPo определяется следующим:

- Емкость (мАч)
- Напряжение (В)
- Элементы

Емкость — это количество энергии, которое аккумулятор может хранить. Представьте это как количество воды в бутылке. Кроме того, чем выше напряжение, тем больший крутящий момент может иметь двигатель постоянного тока. Батарея зависит от вашего приложения. Для простоты мы выберем обычный аккумулятор, как показано на следующем рисунке:

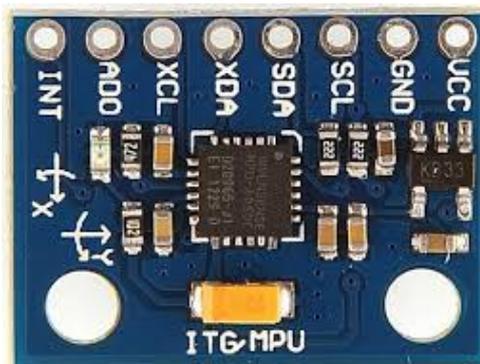


Обратите внимание, что важно подключить все наши батареи в соответствующем порядке. Есть два способа подключения батареи, последовательный и параллельный. При последовательном соединении батарей общее напряжение аккумуляторной батареи увеличивается до суммы напряжений каждой батареи. С другой стороны, при параллельном подключении у вас будет то же напряжение, что и у каждой батареи, но увеличится отдаваемый ток. Таким образом, вы увеличиваете емкость батареи при том же напряжении. Для этого проекта мы соединим наши 5 батареек последовательно и увеличим напряжение до 7,5 В. Каждая имеет 1,5 напряжения, поэтому, если мы соединим пять батарей последовательно, мы получим напряжение  $1,5В \times 5 = 7,5 В$ .

**Примечание** Аккумуляторная батарейка имеет напряжение 1,2 В, поэтому мы получим напряжение  $1,2В \times 5 = 7В$

## Другие компоненты

Мы уже видели некоторые основные компоненты нашего мобильного робота. Мы всегда можем добавить модуль камеры в наш мобильный телефон и иметь глаза. Многие подобные проекты добавляют датчик MPU6050, который имеет три гироскопа и три акселерометра. Это дает нам дополнительную функциональность. Ниже приведено изображение датчика MPU6050:



## Пайка двигателя

Мы увидели некоторые основные компоненты двухколесного мобильного робота. Это основы робототехники, и почти у каждого робота есть некоторые из них. В этом разделе мы рассмотрим все необходимые действия, которые вам необходимо выполнить, чтобы собрать все предыдущие компоненты в рабочий мобильный робот.

На следующем изображении вы можете увидеть некоторые полезные инструменты, облегчающие пайку. Очевидно, что всегда предпочтительнее паяльная станция. Нам осталось только припаять моторчики:



Мультиметр полезен для измерения напряжения двигателей и для отладки нашего робота.

Поместите паяльник на двигатель и соедините провода. Удержание паяльника над проводами более 10 секунд может привести к расплавлению пластиковой крышки двигателя. Конечно, это зависит от вашего паяльника или паяльной станции.

□езультат при хорошей пайке должен быть примерно таким:



## Программирование контроллера

Микроконтроллером является Arduino UNO R3 и наш Raspberry Pi Zero W. Двигатели и датчик расстояния будут управляться с помощью Arduino, который будет обмениваться данными с платой Raspberry Pi Zero W. В зависимости от моторного шилда, который вы будете использовать, вам может потребоваться изменить часть кода, но основная идея останется прежней. Как упоминалось в разделе обзора оборудования, для моторного шилда, который мы будем использовать, требуется определенная библиотека, которая находится в прилагаемой папке с кодами.

Библиотека AFMotor: <https://github.com/adafruit/Adafruit-Motor-Shield-library>

После загрузки всех необходимых файлов вам нужно поместить их в домашнюю папку | Arduino | библиотеки и извлеките файл. Затем откройте IDE Arduino, перейдите в File | Examples и найдите новую установленную библиотеку. Там есть несколько примеров, и вы можете легко использовать их для тестирования ваших двигателей. Предполагая, что вы провели некоторые тесты, теперь мы напишем полный контроллер.

## Основная концепция

Основная идея состоит в том, чтобы разработать контроллер, используя нашу плату Arduino и моторный шилд для управления двумя двигателями. Есть несколько способов сделать это. Это означает, что вы можете использовать свое устройство Android с приложением для управления его движением и перемещением. Вы также можете ничего не использовать и сделать его полностью автономным. Наконец, есть возможность управлять им через компьютер с помощью Wi-Fi.

В этой главе мы увидим автономного мобильного робота и рассмотрим, как настроить соединение с компьютером с помощью Raspberry Pi Zero Wboard. Кодекс состоит из трех основных частей. Первая часть — это код, который создает необходимые переменные и константы, вторая часть — это функция настройки Arduino, которая инициализирует переменные, используемые в части 1, а последняя, часть 3, — это основной цикл. Часть 3 на самом деле является контроллером.

## Разработка контроллера

Следующий код является примером того, как вы можете работать и создавать свой собственный контроллер. Мы будем использовать AFMotor, Servo и математические библиотеки:

```
#include <AFMotor.h>
#include <Servo.h>
#include <math.h>

#define echoPin 24 // Эхо - Pin
#define trigPin 26 // Триггер пин
#define LEDPin 13 // Встроенный LED
#define objAvoidPin 22
#define leftEncPin 40
#define rightEncPin 43

AF_DCMotor leftMotor(4); // M4 - ЛЕВЫЙ двигатель
AF_DCMotor rightMotor(2); // M2 - ПРАВЫЙ двигатель
```

... and more

На этом этапе вам нужно инициализировать некоторые переменные. Определите каждый вывод, будь то INPUT или OUTPUT:

```
void setup() {
  Serial.begin(9600);           // настроить библиотеку на 9600 бит/с
  Serial.println("Мобильный робот up and ready..."); //Мобильный робот
  собран и
  ГОТОВ..
  pinMode(objAvoidPin, INPUT);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  delay(3000);
}
```

На этом этапе вам нужно создать некоторые функции, чтобы упростить кодирование. Например, вы можете увидеть некоторые из них в следующем коде и понять, что нам нужна функция для получения расстояния до переднего ультразвукового датчика. Еще один, чтобы двигаться вперед, двигаться назад, поворачивать влево и вправо.

Функция `updateFrontDistance` считывает ультразвуковой датчик расстояния и сохраняет результат в глобальной переменной. Следующие функции возвращают двигатель вперед или назад:

```
void moveForward(){
  Serial.println("\tMoving forward...");
  leftMotor.run(FORWARD);
  rightMotor.run(FORWARD);

  leftMotor.setSpeed(180);
  rightMotor.setSpeed(180);
}

void posStop(){
  Serial.println("\tStopping...");
  leftMotor.run(RELEASE);
  rightMotor.run(RELEASE);
}

void moveLeft(){
  Serial.print("moving left");
  leftMotor.setSpeed(0);
  rightMotor.setSpeed(150);
  delay(1200);
  rightMotor.setSpeed(0);
}

void turnRight(){
```

```
Serial.println("\tTurning right...");
leftMotor.setSpeed(200);
rightMotor.setSpeed(0);
}

// Turns the servo 180 degrees and finds the position that the mobile can
go forward
void findEmptyRoute() {

    for(int i=10; i<160; i++) {
        Serial.println("Increasing servo ");
        servo.write(i);
        if(distance <=25) {
            updateFrontDistance();
            Serial.print("Distance: ");
            Serial.println(distance);
        }
    }
}
```

Функция `FDSController` используется для обнаружения объекта и поворота мобильного робота:

```
void FDSController() {
    if ( (distance >= 35) && (distance <= 100)) { // Объект обнаружен
        moveForward();
        delay(1000);
    }else {
        turnRight();
        delay(500);
    }
}
```

Наконец, в функции цикла вы можете раскомментировать часть кода в зависимости от вашей цели. Теперь контроллер проверяет расстояние до переднего датчика и движется вперед или поворачивает вправо, если объект найден.

Дополнительную информацию можно найти в репозитории [github](#).

Теперь пришло время использовать нашу плату `Raspberry Pi Zero W` для подключения `Arduino` к устройству. Вы можете сделать это с помощью модуля `Bluetooth` или даже `Wi-Fi`. Например, один из способов — настроить веб-страницу с четырьмя кнопками и подключиться к веб-странице с помощью мобильного телефона или компьютера. При нажатии на кнопку вы сможете отправлять последовательные данные с `Raspberry Pi` на `Arduino`.

Другой способ связи — использование только модуля Bluetooth и приложения для Android. С помощью приложения для Android вы можете управлять движением мобильного робота. Двигайтесь вперед или назад или поворачивайтесь вправо или влево. В PlayStore есть различные приложения, и если у вас есть какие-то знания о разработке для Android, вы всегда можете создать свое собственное приложение.

При разработке контроллера важны две вещи. Прежде всего, вы должны решить, какие сообщения вы будете отправлять с контроллера на плату Raspberry Pi, а затем какие сообщения вы будете отправлять с платы Raspberry Pi на контроллер Arduino. На вашей плате Raspberry Pi вы можете начать последовательную связь с платой Arduino, используя библиотеку Serial. Введите Python, чтобы открыть оболочку

```
python
```

Затем импортируйте последовательную библиотеку, введя следующее:

```
import Serial
```

Затем создайте последовательную связь с вашим Arduino, набрав следующее:

```
ser = serial.Serial(port, baudrate);
```

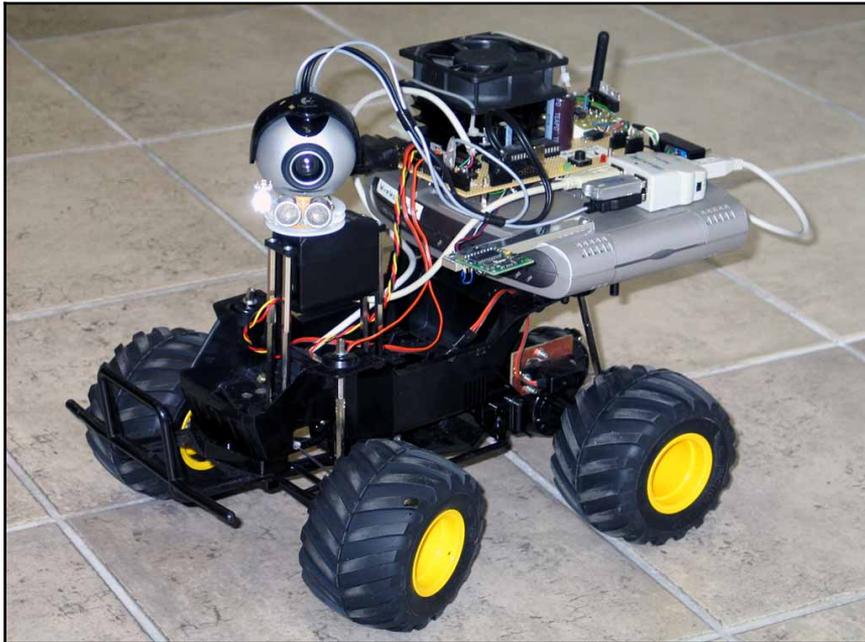
Здесь порт — это порт, к которому подключен ваш Arduino, а скорость передачи зависит от вас. Обычно скорость передачи данных по умолчанию равна 9600. Помните, что в Linux порт — это что-то вроде /dev/tty\*\*\*, тогда как в Windows это что-то вроде COM\*. Заменяя их, вы можете начать связь, и теперь вы сможете разговаривать с Raspberry на плате Arduino.

## Будущие идеи

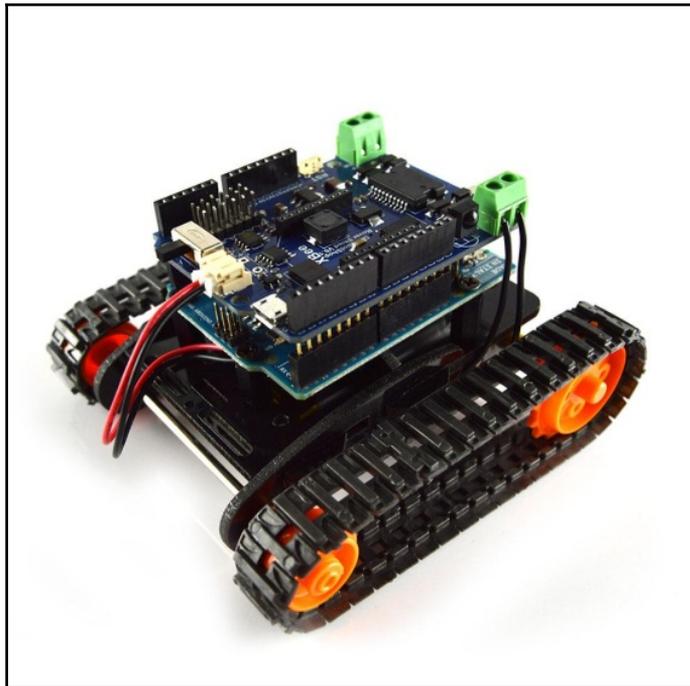
Этот проект был о создании двухколесного мобильного робота. □огика и базовые знания, лежащие в основе этого, имеют решающее значение для создания любого подобного робота. Мы можем добавить компоненты и чипы в мобильного робота и сделать его лучше, добавив больше функций.

## Четырехмоторные мобильные роботы

Вы можете легко модифицировать робота, добавив еще два мотора и еще два колеса. Тогда ваш мобильный робот будет иметь четыре двигателя. Очевидно, вам нужно изменить код и добавить то, что необходимо, но это не должно быть проблемой, так как ядро `ROS` уже закодировано. При изменении способа движения мобильного робота нам нужно подумать, на что это изменение повлияет. Это означает, что центр масс может быть изменен, и орбита движения может быть другой. Например, вот четырехколесный мобильный робот:



Что касается колес и двигателей, вы можете выполнить еще одно простое изменение. Добавьте гусениц к своему роботу и сделайте так, чтобы он мог передвигаться по пыльной или грязной местности. Далее вы можете видеть, что гусеницы весьма полезны для такого типа местности, но, конечно, они довольно дороги, тогда как простые колеса могут стоить всего 2 доллара:



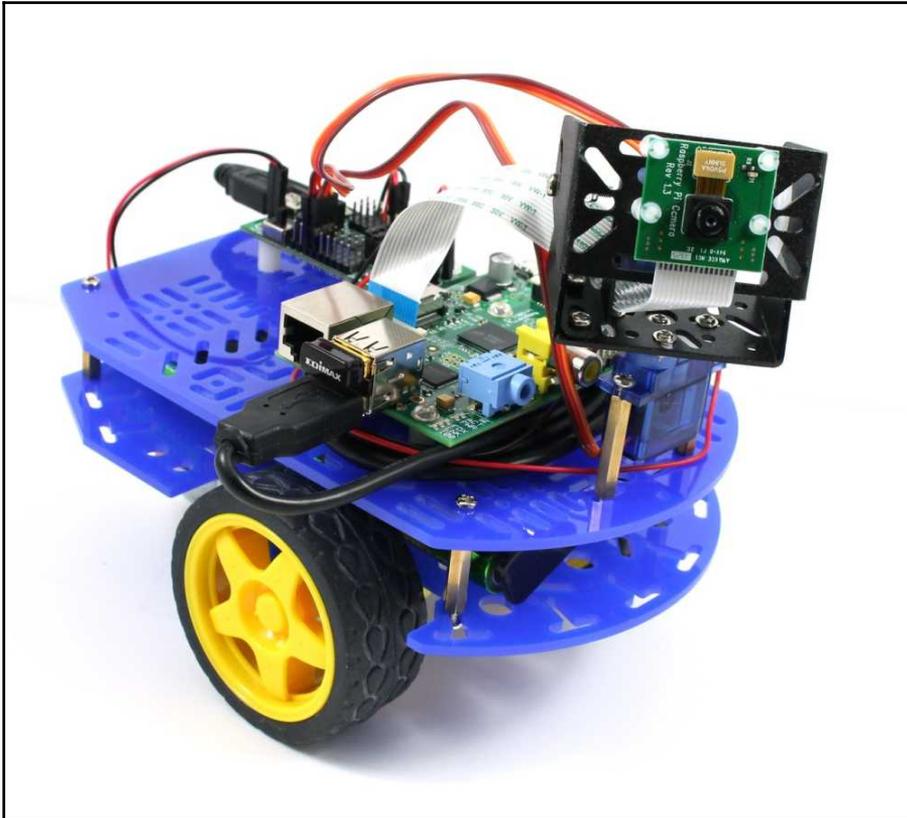
Кроме того, вы можете добавить Grapple с рукой, чтобы держать его. Захват — отличный способ захватывать предметы и перемещать их или менять их положение. Есть несколько проектов, использующих Grapple, таких как робот, который вы можете видеть на следующем изображении. Обычно Grapple управляется сервомеханизмом, но не таким дешевым, как используемый для ультразвукового датчика расстояния в этом проекте. Обычно это сервопривод с металлическими шестернями, которые могут выдерживать большее давление и создавать больший крутящий момент:



Может быть хорошей идеей начать новый проект и научиться управлять Grapple с помощью руки, прежде чем добавлять его к своему мобильному роботу. Таким образом, вы можете быть уверены, что ваше внешнее дополнение Grapple работает правильно. Это более простой способ отладки будущих проблем.

Что касается датчиков, есть несколько решений, в зависимости от того, что вы хотите сделать. Многие разработчики добавили камеру на плату Raspberry Pi и используют алгоритмы для отслеживания движущегося объекта, других движущихся мобильных устройств, идентификации пользователя или даже полного управления роботом с помощью визуального обслуживания на основе изображений (IBVS). Дополнительную информацию об этих дополнительных темах, как всегда, можно найти в поиске Google. Важно и очень полезно, если у вас есть базовые знания и вы читаете соответствующие статьи.

□обот, использующий упомянутую ранее камеру, можно найти на следующем изображении:



Нет никаких ограничений на то, что вы можете делать с мобильным роботом. Единственным ограничением является ваша фантазия, поэтому поищите в Google, посмотрите видео на YouTube и подумайте, что вы хотите сделать. Возможно, это уже было сделано кем-то.

## **Резюме**

В этой главе мы рассмотрели основы робототехники и, конечно же, разработку простого мобильного робота. Может показаться, что создать такого мобильного робота довольно просто, но, например, автономный пылесос не так уж и отличается от него. Вам просто нужно создать еще один проект, который не имеет ничего общего с навигацией робота. Надеюсь, с этими знаниями вы сможете создать своего собственного продвинутого мобильного робота.

В следующей главе мы увидим, как мы можем создать домашнего бота и взаимодействовать с нашим домом.

# 5

## Домашний бот

В предыдущей главе мы шаг за шагом видели, как легко можно создать простого двухколесного мобильного робота. Дальнейшее развитие можно осуществить, внедрив домашнего бота, таким образом, ваш мобильный робот сможет общаться с вашим домашним ботом и выполнять некоторые запланированные задачи. Итак, в этой главе мы увидим, как мы можем создать и разработать автоматизированного домашнего бота-помощника. Домашние боты в настоящее время довольно популярны, поскольку они автоматизируют вещи и действия в нашем доме, месте, где мы проводим большую часть своей жизни. Сценарий, который мы реализуем в этой главе, прост. Наш Raspberry Pi Zero W будет подключен к датчику и отправит уведомление на наш мобильный телефон, если что-то произойдет. Мы рассмотрим список различных тем, а именно:

- Знакомство с домашними ботами
- Программирование сокетов
- Домашняя автоматика

### Знакомство с домашними ботами

В настоящее время боты довольно популярны, и в некоторых случаях они действительно полезны; например, как мы видели в главе 3, чат-бот, представитель службы поддержки клиентов, может быть полностью заменен чат-ботом. Что касается дома, то многие компании, такие как LG, разработали автономные пылесосы; другие разработали умные холодильники.

Есть много идей и реализаций, которые стремятся автоматизировать нашу жизнь и повысить ее качество. Ниже представлено изображение автономного пылесоса, разработанного LG:



Далее мы рассмотрим программирование сокетов, чтобы увидеть несколько примеров того, как мы можем легко общаться с нашим сервером. После этого мы реализуем сервер и клиент со Slack API и другими библиотеками.

## Программирование сокетов

Каждое устройство, подключенное к Интернету, использует сокет для связи с другим устройством или сервером. Основная идея довольно проста; клиент, у которого обычно есть некоторые запросы, подключается к серверу, способному обрабатывать такие запросы, и связь между этими двумя устанавливается с использованием сокетов. В настоящее время существует множество фреймворков, с помощью которых можно легко настроить канал связи, но чтобы максимально понять этот сценарий, мы пройдемся по программированию сокетов, и я все объясню. Подводя итог, клиент запрашивает некоторые данные с сервера, и сервер отвечает клиенту ответом.

Ниже приведено простое изображение, представляющее этот сценарий связи:



На самом деле программирование сокетов — это две стрелки на предыдущем изображении. Она показывает, как реализована связь и как клиент может идентифицировать сервер и отправить некоторые данные, чтобы наконец получить ответ.

В качестве языка программирования мы будем использовать Python. Если вы не знакомы с ним, найдите время и пройдите онлайн-курс, так как очень важно понимать некоторые основы программирования. Python — это простой язык со множеством доступных фреймворков. Однако мы проанализируем реальное программирование, которое происходит под фреймворком. Таким образом, вы сможете расширить этот проект, узнать больше, получить общее представление о том, что мы делаем, и отладить свои проблемы.

## Простое взаимодействие клиент-сервер

Обратите внимание, что эта коммуникационная часть будет работать нормально, но вам может понадобиться использовать некоторые библиотеки, а не чистый код, например, простой сервер и клиент, как описано далее. Сценарий клиент-сервер происходит каждый день. Представьте, если вы идете в магазин обслуживания клиентов и задаете вопрос, очевидно, сотрудник ответит ответом или вопросом, а затем вы скажете что-то похожее на спасибо или перефразируете свой вопрос. Клиент (вы) запрашивает некоторые данные с сервера (<https://www.facebook.com> или сотрудника), а затем сервер проверяет, доступны ли данные и имеете ли вы право на получение данных. Если ошибок не возникает, он отвечает вам запрошенными данными, и вы получаете несколько байтов с сервера. Поскольку сервер хочет быть надежным, он ждет вашего подтверждения с сообщением Hello! Я клиент, и я получил ваши данные. Спасибо. После этого соединение закрывается.

## Простой сервер

Простой сервер состоит из IP-адреса и порта. IP-адрес чем-то похож на 38.24.64.23, состоящий из четырех частей; каждый из них имеет число от 1 до 254. Это похоже на адрес вашего дома, но вместо букв у нас четыре набора цифр. Порт может быть любым числом от 0 до 65535. Некоторые из них являются портами по умолчанию в приложении; например, когда вы выходите в Интернет, вы используете порт 80; когда вы подключаетесь по SSH к удаленному серверу (или к вашему Pi), вы используете порт 22. Таким образом, для простой реализации сервера мы будем использовать IP-адрес и порт. Эта комбинация определяет сервер уникальным образом во всем мире. В следующем примере мы будем использовать нашу локальную машину для демонстрации. Итак, вместо IP-адреса мы будем использовать 127.0.0.1 или имя localhost, что означает, что клиент и сервер находятся на одной машине. Вы также можете следовать этому руководству, используя эти настройки. Давайте выберем номер для нашего порта. Всегда рекомендуется выбирать число далеко от 0 до 100, так как там много портов по умолчанию. Итак, мы будем использовать номер 8000 для нашего порта. Сервер будет работать со следующими настройками:

```
IP address: 127.0.0.1 (or localhost)
Port number: 8000
```

Далее нам нужно настроить сервер и запустить его. Мы можем сделать это, используя очень простой код Python. Мы можем использовать HTTP-сервер из библиотек Python или чистое программирование сокетов. Дополнительную информацию о библиотеке HTTP-сервера можно найти по адресу <https://docs.python.org/2/library/simplehttpserver.html>.

С помощью следующего кода мы можем запустить HTTP-сервер в текущем каталоге:

```
import SimpleHTTPServer
import SocketServer

PORT = 8000
Handler = SimpleHTTPServer.SimpleHTTPRequestHandler
httpd = SocketServer.TCPServer("", PORT), Handler

print "serving at port", PORT
httpd.serve_forever()
```

С другой стороны, мы можем легко запустить HTTP-сервер с помощью одной строки кода:

```
python -m SimpleHTTPServer 8000
```

Если вы откроете браузер и наберете localhost или 127.0.0.1, вы увидите результат. Это должно выглядеть следующим образом:



Этот список содержит некоторые из моих домашних папок. Ваш сервер будет зависеть от каталога, из которого вы запускаете HTTP-сервер. Измените текущий каталог и проверьте это снова. Теперь начинается хардкорное программирование. С такими библиотеками, как `httpssimpleserver`, которые предоставляет Python, все было просто. Мы снова будем использовать Python, но создадим новое серверное приложение. Откройте новый файл в своем любимом редакторе и давайте начнем писать код. Сначала импортируем необходимые библиотеки:

```
import socket  
import time
```

Затем создайте новый сокет со следующими строками кода. Далее мы использовали порт 8888, поэтому измените его на любой, какой хотите, или оставьте его таким, как показано в этой демонстрации. Метод `listen` фактически определяет максимальное количество подключений. Как вы можете видеть дальше, в следующих строках мы разрешаем только до пяти одновременных подключений. Важно увеличить масштаб вашего сервера, так как у нас явно не будет стойки или серверной комнаты. Обычный компьютер не может обрабатывать тысячи подключений.

```
# Create socket  
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  
sock.bind(('', 8888))  
sock.listen(5)
```

Далее попробуем прочитать сообщение от клиента. Метод `accept()` зависает навсегда, пока не появится новое соединение:

```
try:
    # Echo loop
    while True:
        try:
            newSocket, address = sock.accept()
        except socket.error, (value, message):
            print "Socket error while waiting to accept socket: " +
                // "Ошибка сокета при ожидании принятия сокета:" +
message
            except:
                print "Unexpected error while waiting to accept"
                // "Непредвиденная ошибка при ожидании принятия"
                break
```

Далее метод `recv()` получает все данные, которые отправляет клиент. Если все в порядке, то сервер отвечает сообщением: I received your data. All ok! - Я получил ваши данные. Все в порядке!

```
try:
    # Echo loop
    while True:
        try:
            newSocket, address = sock.accept()
        except socket.error, (value, message):
            print "Socket error while waiting to accept socket: " +
message
            except:
                print "Unexpected error while waiting to accept"
                break
        print "Connected from echo client ", address
        while True:
            try:
                receivedData = newSocket.recv(1024)
            except socket.error as (value, message):
                print "Socket error while waiting to receive:
(value:", value, ") " + message
                break
            except:
                print "Unexpected error while waiting to receive"
                break

            if not receivedData:
                break
            else:
                print "Server received: " + receivedData

            newSocket.sendall("I received your data. All ok!")
```

```
        # Done transacting with client
        newSocket.close()
        print "Disconnected from", address

except:
    print "Unexpected error in echo loop"
finally:
    sock.close()
```

В итоге приходится закрывать соединение, так как с открытым соединением могут возникнуть проблемы при повторной попытке выполнить его. Это хорошая техника программирования, чтобы закрывать то, что нам больше не нужно.

## Простой клиент

Мы должны разработать клиентское приложение, которое будет намного проще. Следующий код создаст клиент, который подключается к существующему серверу и отправляет сообщение. После этого клиент получает ответ от сервера и распечатывает его. Как клиент, мы также можем иметь наше приложение Slack для мобильного телефона, как я опишу позже в расширенном клиенте. Это важно, потому что для связи клиент-сервер обе части должны иметь возможность отправлять и получать данные.

Сначала нам как всегда нужно импортировать все необходимые библиотеки:

```
import socket
import time
```

Далее нам нужно подключиться к серверу, расположенному на локальном хосте:

```
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(('localhost', 8888))
print "Connected to echo server"
```

Далее отправляем сообщение на сервер:

```
try:
    sock.sendall("Hello there!")
```

Теперь пытаемся получить ответ или ошибку:

```
try:
    response = sock.recv(1024)
except socket.error as (value, message):
    print "Socket error while waiting to receive: (value:", value, ") "
+ message
except:
    print "Unexpected error"
```

После этого мы просто печатаем ответ:

```
print "Received:", response
if not response:
    break
else:
    print "Message sended!"
```

Перед запуском клиента необходимо запустить серверный код. Вы должны убедиться, что сервер работает, а затем попытаться запустить клиентское программу. Полный код выглядит следующим образом:

```
import time
import socket

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(('localhost', 8888))
print "Connected to echo server"
try:
    sock.sendall("Hello there!")
    try:
        response = sock.recv(1024)
    except socket.error as (value, message):
        print "Socket error while waiting to receive: (value:", value, ") "
+ message
    except:
        print "Unexpected error"
    print "Received:", response
    if not response:
        break
    else:
        print "Message sended!"
except:
    print "Unexpected error in echo loop"

sock.close()
```

## Расширенная связь клиент-сервер

После простого взаимодействия клиент-сервер пришло время использовать некоторые вызовы API и другие библиотеки для разработки более простого и удобного в использовании взаимодействия между нашим сервером и клиентским приложением. Давайте начнем с реализации сервера дальше. Во-первых, мы реализуем серверную часть, которая будет нашей платой Raspberry Pi Zero W.

## Расширенный сервер

Теперь мы будем использовать API сервиса по вашему выбору, чтобы получать наши уведомления. Есть несколько способов сделать это, но для простоты мы продемонстрируем slack API. Slack API предоставляет нам функции и обработчики, чтобы легко оповещать наше настольное или мобильное приложение. Он также предоставляет функцию, позволяющую выполнять различные взаимодействия, такие как отправка сообщений обратно на наш сервер и обратно к нам домой. Код, который мы будем использовать, прост, и мы не будем углубляться в анализ того, как и почему он работает. Конечно, первым шагом здесь является создание slack учетной записи. Перейдите на официальный сайт slack, <https://slack.com/>, чтобы создать новую команду, а также загрузить и установить приложение slack на свой мобильный телефон. Когда все будет готово, мы снова воспользуемся Python, чтобы написать еще несколько строк кода и соединить нашу программу со slack приложением на нашем настольном или мобильном ПК. Для начала создайте новую папку с именем slackapi:

```
mkdir slackapi
```

Теперь создайте новый `virtualenv`, чтобы изолировать зависимости вашего приложения от других проектов, над которыми вы, возможно, работаете:

```
virtualenv venv
```

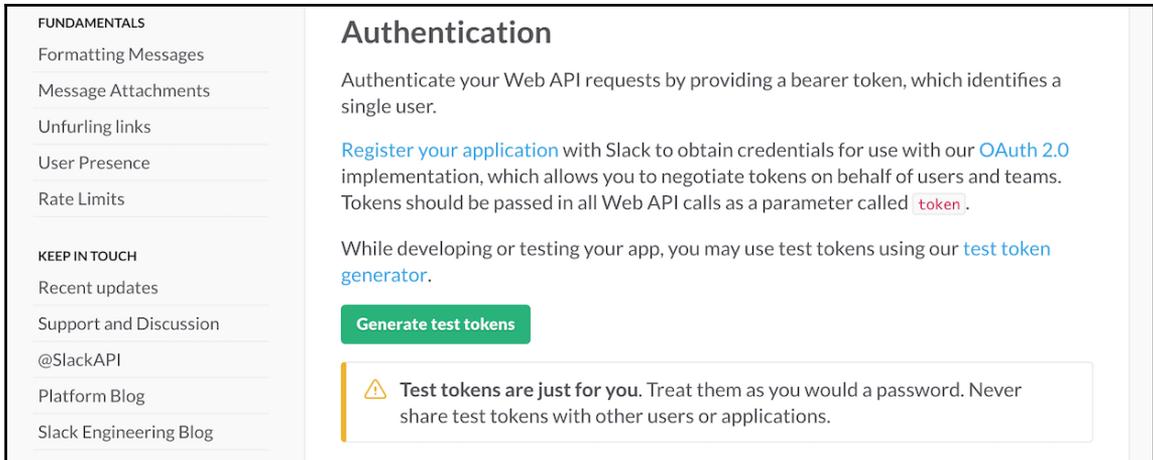
Обратите внимание, что в предыдущем шаге нет необходимости, если вы знаете, что делаете, и хотите настроить это в своей системе. После этого вам нужно активировать `virtualenv` с помощью следующего кода:

```
source venv/bin/activate
```

Если все пойдет хорошо, вы должны увидеть, что ваше приглашение изменилось на что-то похожее на `(venv) $`, но оно также может быть другим. У вас есть много вариантов, так как в Python много библиотек, но для простоты мы будем использовать `slackclient`. Итак, установите этот клиент с помощью следующего кода:

```
pip install slackclient==1.0.0
```

Далее перейдите на официальный сайт slack по адресу <https://api.slack.com/web> и, если вы не вошли в систему, войдите под своей учетной записью. Затем прокрутите вниз страницу веб-API, где вы увидите кнопку для создания тестовых токенов:



**FUNDAMENTALS**

- Formatting Messages
- Message Attachments
- Unfurling links
- User Presence
- Rate Limits

**KEEP IN TOUCH**

- Recent updates
- Support and Discussion

@SlackAPI

- Platform Blog
- Slack Engineering Blog

## Authentication

Authenticate your Web API requests by providing a bearer token, which identifies a single user.

Register your application with Slack to obtain credentials for use with our [OAuth 2.0](#) implementation, which allows you to negotiate tokens on behalf of users and teams. Tokens should be passed in all Web API calls as a parameter called `token`.

While developing or testing your app, you may use test tokens using our [test token generator](#).

[Generate test tokens](#)

⚠ Test tokens are just for you. Treat them as you would a password. Never share test tokens with other users or applications.

Обратите внимание, что у вас должны быть права администратора, чтобы создать тестовый токен для вашей рабочей группы. Если вы это сделали, сгенерируйте токен. Этот токен отлично подойдет для этого примера. Для получения дополнительной информации и расширенного использования посетите <https://api.slack.com/custom-integrations/legacy-tokens>, где другие пользователи могут создавать токены для аутентификации через свои собственные учетные записи.

Не закрывая вкладку, переключитесь в свою среду Python, чтобы настроить и опробовать API. Запустите оболочку Python со следующим кодом:

```
(venv) $ python
```

Проверьте свой токен API с помощью тестового вызова, введя следующий код:

```
>>> from slackclient import SlackClient
>>> slack_client = SlackClient('your test token here')
>>> slack_client.api_call("api.tet")
```

Если все пойдет хорошо, вы должны получить что-то похожее на следующий результат:

```
{'ok': False, 'error': u'invalid_auth'}
```

Проверьте еще раз, правильно ли вы скопировали токен. Для получения дополнительных тестов выполните следующую строку кода столько раз, сколько хотите:

```
>> slack_client.api_call("auth.test")
```

Вы всегда должны получить ответ, аналогичный предыдущему с другими номерами. На данный момент все подключено через slack API и мы не можем получать slack данные и обрабатывать сообщения! Экспортируйте свой слабый токен со следующим кодом:

```
export SLACK_TOKEN='your slack token pasted here'
```

На этом этапе используйте свой любимый редактор, такой как vim, nano, gedit или любой другой, чтобы написать код Python. Создайте новый файл с именем app.py и импортируйте следующие библиотеки:

```
import os
from slackclient import SlackClient
```

После этого получите токен, который вы ранее экспортировали в ОС, и создайте slack клиент, как мы делали в нашем предыдущем примере:

```
SLACK_TOKEN = os.environ.get('SLACK_TOKEN')
slack_client = SlackClient(SLACK_TOKEN)
```

Теперь давайте создадим новую функцию для вывода списка каналов через вызов API. Slack возвращает результаты в словаре с двумя ключами: ok и channels.list. Первый, ok, позволяет нам узнать, был ли вызов API успешным или нет, и, таким образом, если его значение истинно, то каналы будут содержать все необходимые нам данные. Если у нас нет канала, он вернет false.

```
def list_channels():
    channels_call = slack_client.api_call("channels.list")
    if channels_call.get('ok'):
        return channels_call['channels']
    return None
```

Наконец, добавьте свою основную функцию, которая позволит нам печатать все каналы:

```
if __name__ == '__main__':
    channels = list_channels()
    if channels:
        print("Channels: ")
        for c in channels:
            print(c['name'] + " (" + c['id'] + ")")
    else:
        print("Unable to authenticate.")
```

После этого получите токен, который вы ранее экспортировали в ОС, и создайте slack клиент, как мы делали в нашем предыдущем примере:

```
python app.py
```

Вы получите вывод, аналогичный следующему:

```
Channels:
  general (C0S82S5RS)
  python (C0S8HABL3)
  random (C0S8F4432)
```

Обратите внимание, что API slack нуждается в уникальной ссылке для каналов. Это означает, что мы можем легко использовать идентификатор, а не имя для ссылки на канал. На этом этапе вы должны понять, что мы пытаемся сделать. Наш следующий и последний шаг здесь — отправить некоторые данные обратно в приложение, используя различные методы Slack API. Вы можете комбинировать весь продемонстрированный код для разработки собственной программы.

Обновленный код выглядит следующим образом:

```
import os
from slackclient import SlackClient

SLACK_TOKEN = os.environ.get('SLACK_TOKEN', None)
slack_client = SlackClient(SLACK_TOKEN)

def list_channels():
    channels_call = slack_client.api_call("channels.list")
    if channels_call['ok']:
        return channels_call['channels']
    return None

def channel_info(channel_id):
    channel_info = slack_client.api_call("channels.info",
channel=channel_id)
    if channel_info:
        return channel_info['channel']
    return None

if __name__ == '__main__':
    channels = list_channels()
    if channels:
        print("Channels: ")
        for c in channels:
            print(c['name'] + " (" + c['id'] + ")")
            detailed_info = channel_info(c['id'])
            if detailed_info:
                print(detailed_info['latest']['text'])
    else:
        print("Unable to authenticate.")
```

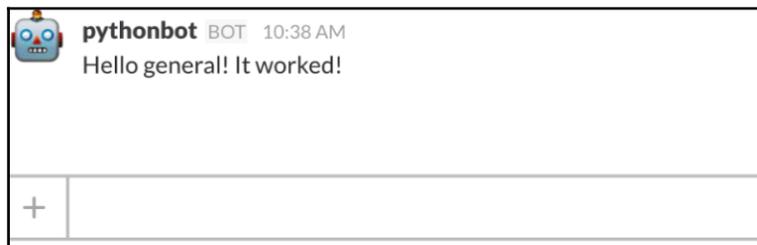
Чтобы отправлять сообщения обратно в slack через вызовы API, мы должны добавить новый метод следующим образом:

```
def send_message(channel_id, message):
    slack_client.api_call(
        "chat.postMessage",
        channel=channel_id,
        text=message,
        username='pythonbot',
        icon_emoji=':robot_face:'
    )
```

Таким образом, мы можем отправлять сообщения на канал #general. Ваша основная функция должна выглядеть следующим образом:

```
if __name__ == '__main__':
    channels = list_channels()
    if channels:
        print("Channels: ")
        for channel in channels:
            print(channel['name'] + " (" + channel['id'] + ")")
            detailed_info = channel_info(channel['id'])
            if detailed_info:
                print('Latest text from ' + channel['name'] + ":")
                print(detailed_info['latest']['text'])
            if channel['name'] == 'general':
                send_message(channel['id'], "Hello " +
                    channel['name'] + "! It worked!")
        print('-----')
    else:
        print("Unable to authenticate.")
```

Обратите внимание, что для всех каналов мы ищем канал с общим именем. Как только мы находим его, мы отправляем сообщение с идентификатором канала и словом Hello, затем название канала, а затем сообщение! Это сработало!. Если вы сохраните файл и снова выполните код, вы должны получить уведомление о своем slack приложении в общем канале. Попробуйте изменить сообщение на то, что вам нравится, и проверьте его снова:



К этому моменту вы узнали, как создать уведомление с вызовами Slack API. Довольно легко объединить наш предыдущий код с датчиком температуры и новый код с медленным API и отправить уведомление, когда температура превысит 25 градусов Цельсия.

## Расширенный клиент

Клиент здесь на самом деле является приложением на нашем мобильном устройстве или на нашем настольном компьютере. Для простоты и для этого проекта клиент не будет отправлять никаких сообщений обратно на сервер. Клиент будет получать уведомления только в том случае, если температура превысит 25 градусов по Цельсию. Однако расширенный клиент можно разработать, добавив две функции. Фактическая функция, которая отправляет сообщение:

```
def send_message(channel_id, message):
    slack_client.api_call(
        "chat.postMessage",
        channel=channel_id,
        text=message,
        username='pythonbot',
        icon_emoji=':robot_face:'
    )
```

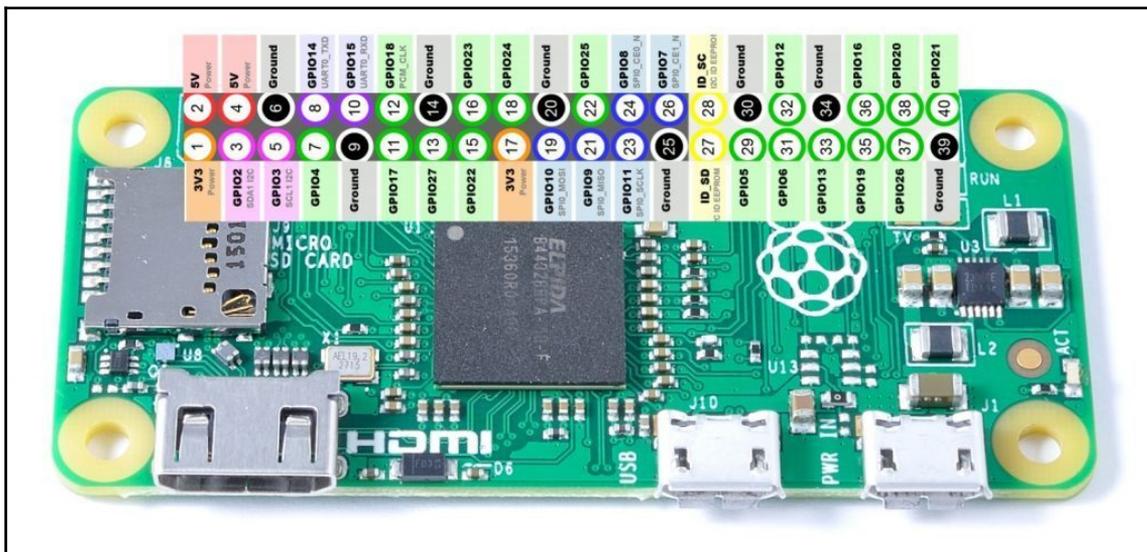
В предыдущем коде вы выбираете канал, в который хотите отправить сообщение и, конечно же, основную функцию:

```
if __name__ == '__main__':
    channels = list_channels()
    if channels:
        print("Channels: ")
        for channel in channels:
            print(channel['name'] + " (" + channel['id'] + ")")
            detailed_info = channel_info(channel['id'])
            if detailed_info:
                print('Latest text from ' + channel['name'] + ":")
                print(detailed_info['latest']['text'])
            if channel['name'] == 'general':
                send_message(channel['id'], "Hello " +
                    channel['name'] + "! It worked!")
        print('-----')
    else:
        print("Unable to authenticate.")
```

Мы не будем анализировать этот код, но вы можете попробовать отправить сообщения обратно на сервер; например, вы можете просто получить уведомление и отправить обратно сообщение ok.

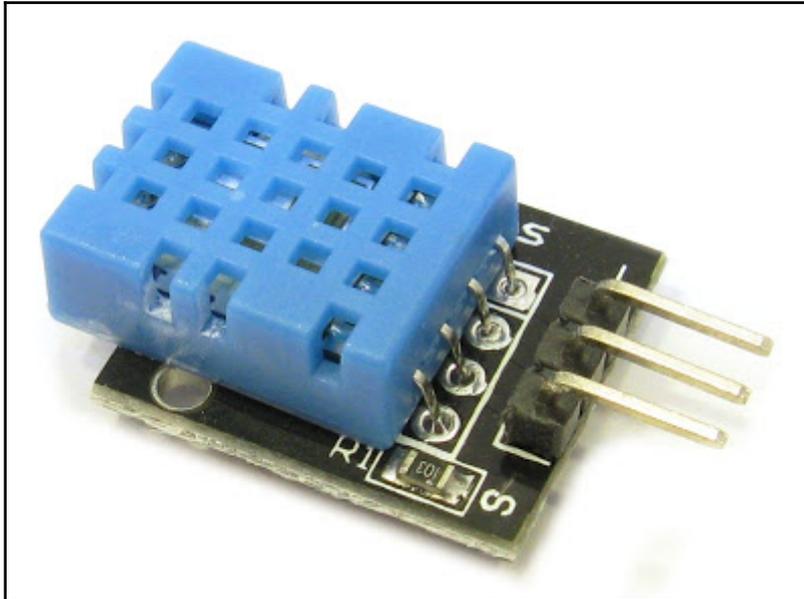
## Домашняя автоматика

Для полной домашней автоматике мы будем использовать нашу плату Raspberry Pi Zero W и датчик температуры и влажности DHT11. Датчик DHT11 может измерять как температуру, так и влажность. Его довольно легко настроить, и мы сделаем это с помощью библиотеки. Датчик DHT11 состоит всего из трех контактов: один для земли, один для 5 В и последний для сигнала. Мы должны аккуратно подключить их к соответствующим контактам на нашей плате Raspberry Pi Zero W. На следующем изображении вы можете найти распиновку платы:



Если этот проект новый, а не дополнение, все пины должны быть доступны; Итак, подключите контакт заземления датчика к контакту 9. Затем подключите контакт 5V датчика к контакту 2 и, наконец, подключите сигнальный контакт датчика к контакту 7. Важно, чтобы он был подключен к контакту 7, так как позже мы свяжем этот вывод с нашим кодом Python.

Ниже приведено изображение датчика температуры и влажности, который мы использовали:



Прежде всего, обновите свою плату Pi до последней версии:

```
sudo apt-get update
```

Затем, если git не установлен на вашей плате Pi, установите его, чтобы мы могли позже клонировать соответствующий репозиторий с GitHub:

```
sudo apt-get install git-core
```

Далее вам необходимо установить библиотеку Adafruit DHT11. Мы будем использовать эту библиотеку для подключения нашего датчика Pi к коду Python и считывания температуры и влажности:

```
git clone
```

Измените текущий каталог на каталог, который вы только что загрузили:

```
cd Adafruit_Python_DHT
```

Чтобы написать наш код и закончить эту часть, если они у вас не установлены, установите все необходимые библиотеки:

```
sudo apt-get install build-essential python-dev
```

Наконец, установите библиотеку с помощью следующего кода:

```
sudo python setup.py install
```

Теперь мы должны запустить наш код и вывести температуру и влажность. Если vim не установлен в вашем Pi, вы можете пропустить этот шаг и использовать ваши любимые редакторы, такие как nano или gedit с графическим интерфейсом:

```
sudo apt-get install vim
```

После установки откройте редактор vim, набрав команду (вы также можете пропустить этот шаг):

```
vim
```

После нажатия i для перехода в режим вставки введите следующий код:

```
#!/usr/bin/python
import sys
import Adafruit_DHT
import time

while True:
    humidity, temperature = Adafruit_DHT.read_retry(11, 4)
    print "Humidity: " + str(humidity)
    print "Temperature: " + str(temperature) + "\n"
    time.sleep(1)
```

Код на самом деле довольно прост; мы создали цикл while, а это значит, что блок внутри него будет работать вечно; затем вы вызываете функцию Adafruit\_DHT.read\_retry (11, 4), чтобы вернуть влажность и температуру на контакт 7, которые мы прикрепили к нашему датчику. Наконец, мы печатаем данные и засыпаем на одну секунду.

Чтобы закрыть и сохранить файл, нажмите Esc, затем введите: wq! example.py и нажмите Enter. Вы должны увидеть файл с именем example.py. Для получения дополнительной информации взгляните на [http://vim.wikia.com/wiki/Saving\\_a\\_file](http://vim.wikia.com/wiki/Saving_a_file). Чтобы запустить код, просто введите:

```
python example.py
```

Вы должны увидеть следующий вывод:

```
pi@raspberrypi:~/ch5-temp-project$ python temphum.py
Humidity: 91.0
Temperature: 25.0

Humidity: 91.0
Temperature: 26.0

Humidity: 91.0
Temperature: 25.0
```

Теперь можно использовать оператор **if** внутри кода Python и проверить, превышает ли температура 25 градусов по Цельсию, а затем выполнить и вызвать резервный код, который мы разработали ранее. Вы можете легко добавить больше автоматике и сделать своего бота более умным и способным понимать больше вещей в вашей среде.

## Резюме

В этой главе мы рассмотрели проект клиент-серверного приложения для связи через сокеты. Мы рассмотрели некоторые основные функции программирования сокетов на очень низком уровне, но также и на более высоких уровнях, используя вызовы slack API и простые библиотеки HTTP Python. Вы должны иметь возможность добавлять дополнительные датчики на плату Raspberry Pi Zero W и настраивать свой код, чтобы автоматизировать больше вещей и действий. Ваш домашний бот должен сделать ваш дом более безопасным и умным! В следующей главе мы поиграем с камерами и распознаванием лиц.

# 6

## Камера безопасности

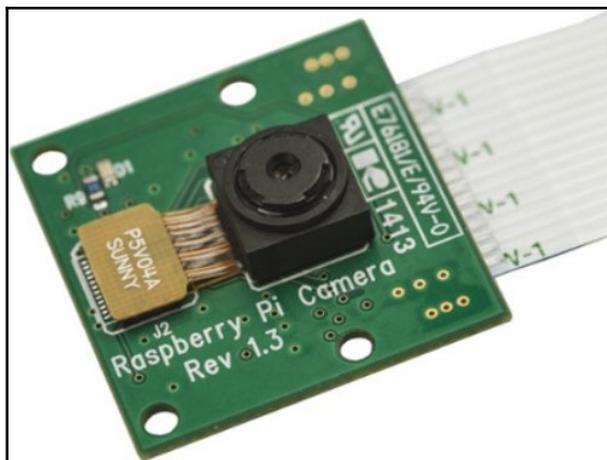
В предыдущей главе мы исследовали домашних ботов. Мы рассмотрели, как домашний бот может быть автономным и отправлять вам уведомления. В этой главе мы рассмотрим разработку камеры видеонаблюдения. Мы продемонстрируем расширенный способ подключения камеры к плате Raspberry Pi с помощью MotionPie. Позже мы обсудим библиотеку OpenCV на случай, если вы захотите расширить этот проект. Подводя итог, вот темы, которые мы рассмотрим в этой главе:

- Установка камеры
- Установка MotionPie
- Настройка параметров MotionPie
- Установка OpenCV
- Распознавание лица

### Установка камеры

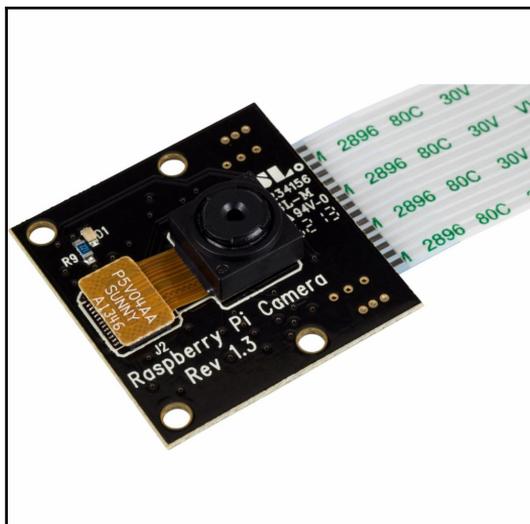
Как и все платы Raspberry Pi, новая плата Raspberry Pi Zero W имеет слот для подключения камеры. Однако, прежде всего, нам нужно купить камеру Raspberry Pi на случай, если у нас ее нет. Есть два типа камер, которые вы можете купить: обычная ИК-камера или камера без ИК-подсветки. Обычная ИК-камера не позволяет хорошо видеть в темных местах. Это означает, что как только солнце зайдет, будет трудно увидеть какие-либо объекты. С другой стороны, камеры без инфракрасного излучения отлично подходят для такого рода проектов. В зависимости от проекта купите соответствующую камеру. Здесь вы можете найти две ссылки для обоих типов камер.

Обычная камера выглядит следующим образом:

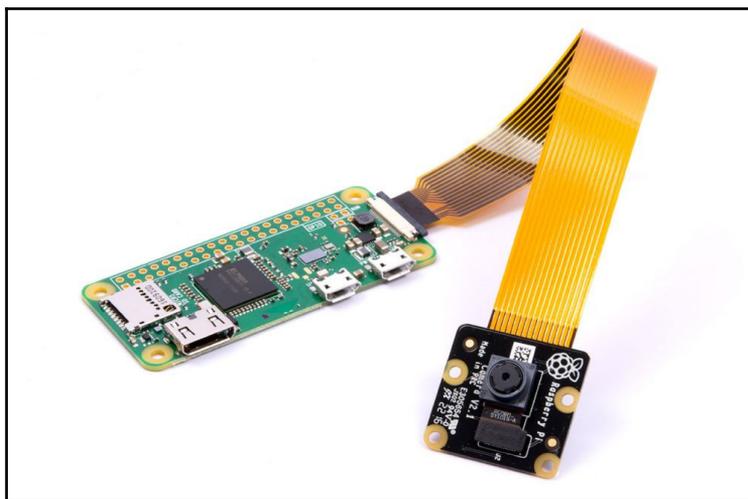


Модуль ночного видения Raspberry Pi 5MP 1080P Camera NoIR (без ИК-фильтра):

Расширенный модуль камеры выглядит следующим образом. Он не имеет ИК-фильтра и прекрасно подходит для ночного видения:



Чтобы подключить камеру к плате Raspberry Pi Zero W, просто поместите ее вручную и потяните за обе стороны разъема. Как только разъем будет подтянут, попробуйте поместить ленточный кабель внутрь и аккуратно нажмите на разъем платы, который вы подтянули ранее. Теперь камера должна быть заблокирована на плате, вот и все:



## Установка MotionPie

MotionPie — отличное решение для множества разных проектов, когда задействована камера. Чтобы установить MotionPie, выполните следующие действия:

1. Во-первых, мы должны загрузить и отформатировать нашу SD-карту с помощью соответствующей программы обеспечения:



Загрузите последнюю версию из репозитория GitHub и отформатируйте SD-карту. Ссылка на репозиторий

<https://github.com/ccrisan/motioneyeos/releases>. Если вы не помните, как форматировать SD-карту, обратитесь к Главе 1, **Введение в Raspberry Pi Zero W**, где была представлена Raspberry Pi Zero W, и повторите инструкции еще раз.

2. Затем вставьте SD-карту и загрузите плату.
3. Теперь нам нужно найти IP-адрес или имя хоста платы Raspberry Pi. Если вы опытный пользователь, пропустите этот шаг, иначе, следуйте инструкциям:
  - Для пользователей Windows зайдите в сеть, затем в проводник, и вы должны увидеть имя компьютера, например MP-E28D9CE5. Скопируйте это имя и откройте браузер. Введите следующее:

```
http://MP-E28D9CE5
```

- Теперь, если все в порядке, у вас должен быть запущен интерфейс Motion Pie.
- Другой способ узнать IP-адрес платы Raspberry Pi — перейти к маршрутизатору и узнать, какое устройство, подключенное по беспроводной сети, называется Raspberry Pi. Чтобы войти в систему как администратор, перейдите к символу ключа в верхнем левом углу. Обратите внимание, что имя пользователя — **admin**, а пароль следует оставить пустым. Конечно, позже мы их изменим, а пока просто авторизуйтесь. Если все пройдет нормально, вы сможете увидеть интерфейс камеры и получить доступ ко всем настройкам потока камеры.

## Настройка нескольких сетевых камер

Если вы хотите масштабировать это и подключить больше сетевых камер с помощью программы, это довольно просто. Таким образом, вы можете иметь более одной камеры в одном окне. Для этого выполните следующие действия:

1. Нажмите на три линии в верхнем левом углу программы.
2. В верхнем левом углу должно появиться раскрывающееся окно; нажмите на него и выберите `add camera` - добавить камеру.

Теперь нам нужно реализовать все четыре необходимые настройки:

1. Во-первых, `Device` позволяет вам выбрать, где будет располагаться камера и какого она типа; например, она может быть расположена в локальной сети и может быть камерой типа MJPEG.
2. Затем URL-адрес — это URL-адрес другой сетевой камеры. Каждая камера имеет IP-адрес или имя хоста в локальной сети, например, `http://othercamera:8080`.
3. Далее, `Username` - имя пользователя и `Password` - Пароль являются полями устройства камеры. Для большей безопасности в вашей системе рекомендуется добавить имя пользователя и пароль, а не оставлять это поле пустым.
4. Наконец, поле «Camera» фактически позволяет вам выбрать камеру, которую вы хотите добавить в систему, и все. Теперь все необходимые настройки сделаны.

## Подключение камеры за пределами вашей локальной сети

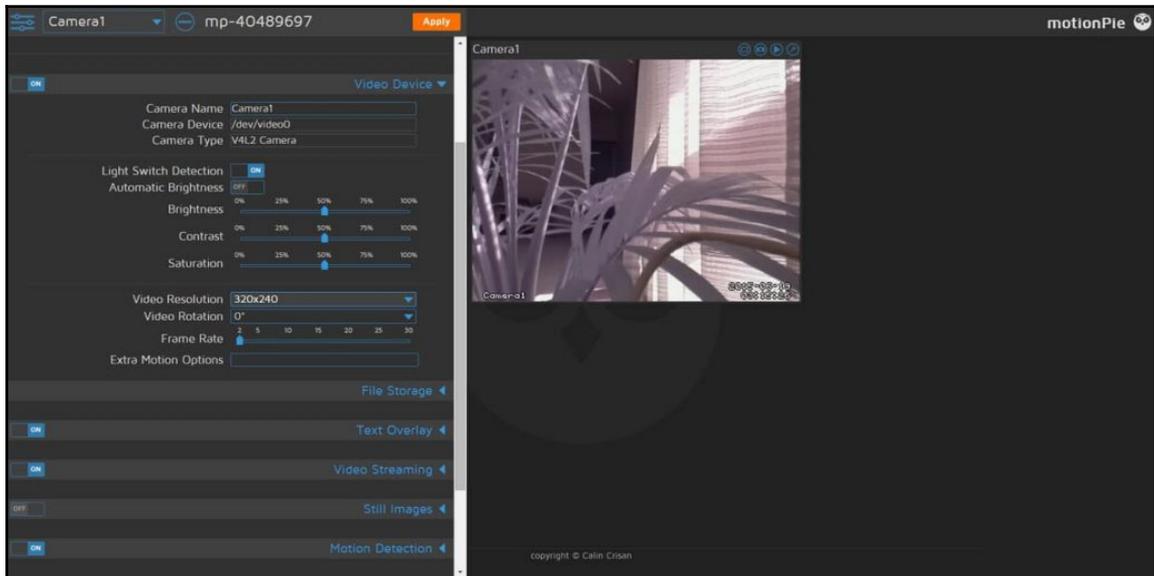
На этом этапе вам следует рассмотреть возможность подключения платы Raspberry Pi Zero W к роутеру и получения доступа из любой точки мира. Для этого вам необходимо перенаправить входящие соединения на плату Raspberry Pi Zero W, как мы уже видели в предыдущих главах, когда говорили о сети. По сути, вам нужно найти свой IP-адрес Raspberry Pi Zero W и после входа в интерфейс роутера создать правило переадресации портов для всех или определенных входящих подключений к этому IP-адресу и порту 80. Затем, чтобы защитить свой локальный камеры безопасности, вы можете рассмотреть возможность настройки имени пользователя и пароля.

## Камера безопасности

Таким образом, вы затрудняете доступ нежелательных пользователей к вашей локальной системе безопасности. После этого вы можете открыть браузер и перейти по URL-адресу <X.X.X.X>:80, где X.X.X.X — ваш домашний IP-адрес/роутер от вашего интернет-провайдера. Вы можете найти этот IP-адрес, просто зайдя в Google и выполнив поиск *What is my IP address?* - Какой у меня IP-адрес?. Рассмотрите возможность изменения порта 80 на какой-либо другой порт, который должны знать только вы. Это усложняет доступ к вашей системе для посторонних. На самом деле им нужно будет сканировать вашу сеть на наличие открытых портов и найти, какой из них вы используете. Помните, что мы можем создать столько уровней безопасности, сколько сможем, и вы не сможете быть на 100% защищены от кого бы то ни было.

## Настройка параметров MotionPie

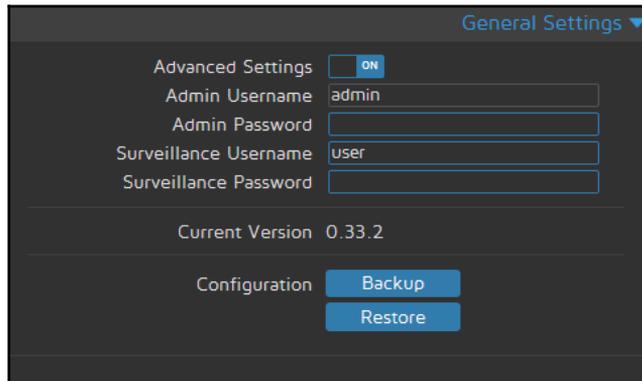
Теперь пришло время настроить параметры MotionPie. Предполагая, что вы открыли интерфейс MotionPie, вы должны увидеть интерфейс MotionPie следующим образом:



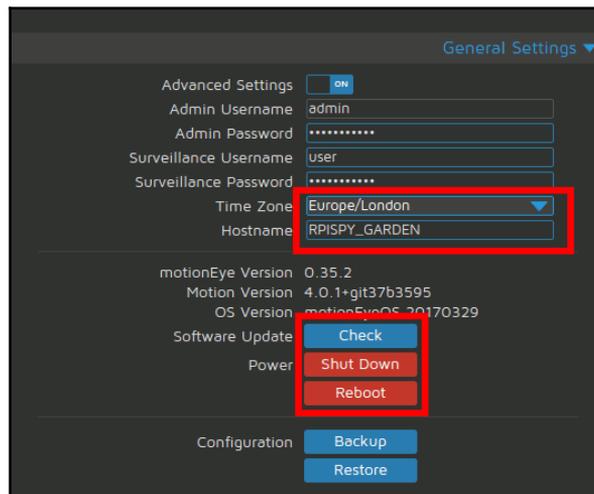
В левой части экрана есть несколько разделов с разными настройками. Сначала пройдемся по общим настройкам.

## General settings - Общие настройки

В этом разделе вы сможете установить имя пользователя и пароль администратора, как мы обсуждали ранее. Это важный шаг, и вы не должны его пропускать. Эта учетная запись (аккаунт) будет иметь доступ ко всем настройкам, которые вы видите в данный момент. Здесь также можно установить имя пользователя и пароль для наблюдения — их можно использовать только для доступа к интерфейсу камеры. Общие настройки следующие:



Чтобы просмотреть все доступные настройки, вы должны включить Advanced Settings - Дополнительные настройки. Вы можете увидеть все настройки в расширенном меню, как показано ниже:

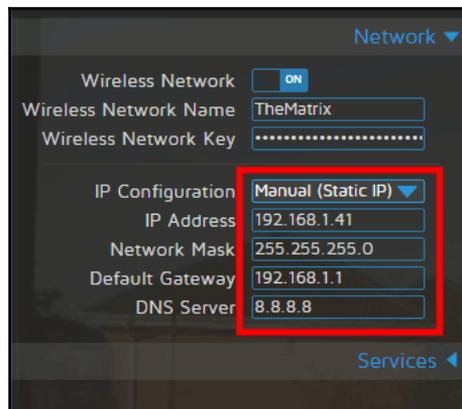


## Беспроводная сеть

В этом разделе вы можете изменить все настройки, относящиеся к беспроводной сети. Включите этот параметр, если вы планируете подключаться к сети через беспроводной ключ. Raspberry Pi 3 Zero W имеет встроенный Wi-Fi, так что вы, вероятно, сможете его использовать. Таким образом, вам нужно будет заполнить две вещи:

- **Network name:** Имя сети: введите имя сети или SSID, к которому вы хотите подключиться
- **Network key:** Сетевой ключ: введите сетевой пароль или сетевой ключ для сети, к которой вы планируете подключиться.

Вы можете увидеть меню беспроводной сети и сети следующим образом:



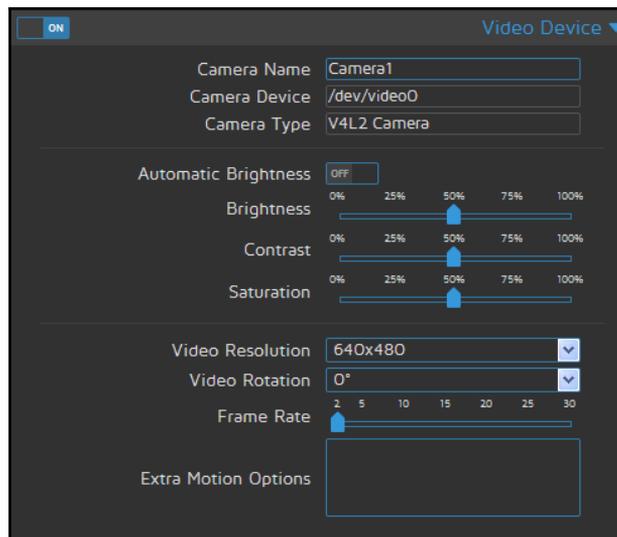
## Видеоустройство

В этом меню вы сможете установить определенные параметры камеры Raspberry Pi. Это означает, что вы можете установить имя камеры и некоторые основные характеристики, относящиеся к среде, в которой вы планируете установить эту систему безопасности. Итак, вот список параметров, которые вы можете изменить.

- **Camera Name** - Имя камеры: Это поле на самом деле просто имя для вас. Установите это на то, что представляет эту камеру; например, если камера находится в саду, вы можете назвать ее Garden Camera.

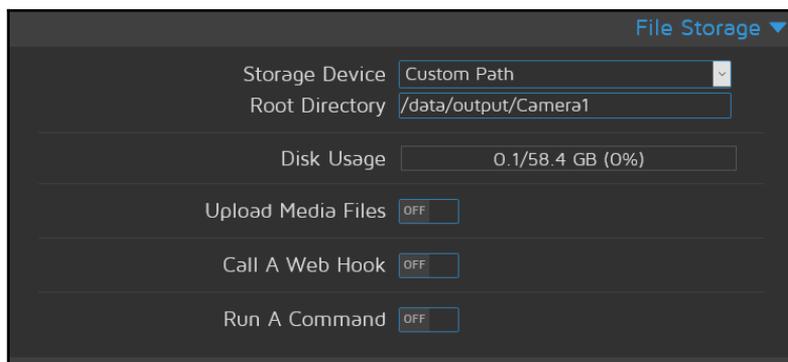
- **Camera Device:** Это поле является именем устройства камеры и, к сожалению, его нельзя редактировать.
- **Light Switch Detection:** Обнаружение выключателя света: вы можете включить это, если хотите внезапных изменений; например, камера в вашей комнате будет воспринимать включение и выключение света как движение. Чтобы этого не произошло, используйте эту опцию.
- **Automatic Brightness:** Она включит программную автоматическую яркость; в результате эта камера автоматически регулирует яркость в зависимости от количества солнечного света, точно так же, как новые телефоны Android регулируют свою яркость. Обратите внимание, что некоторые камеры уже поддерживают эту функцию, поэтому в этом случае вам не нужно активировать ее, если ваша камера уже поддерживает ее.
- **Video Resolution:** Здесь вы можете установить разрешение видео камеры. Чем выше разрешение, тем больше места вам потребуется на SD-карте и тем больше пропускной способности потребуется для потоковой передачи.
- **Video Rotation:** Вы всегда можете повернуть свое видео с платы Raspberry Pi Zero W, если по какой-либо причине вы обнаружите перевернутые кадры потока.
- **Frame Rate:** И последнее, но не менее важное: вы можете установить количество кадров, которые будут отправляться каждую секунду. Опять же, имейте в виду, что чем выше это значение, тем плавнее видео, но также это увеличит объем необходимой памяти и, конечно же, пропускную способность.

Теперь мы рассмотрим настройки меню, как они появляются в Video Device - меню видеодустройства:



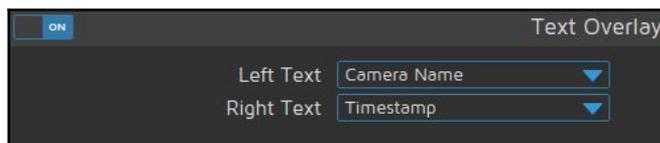
## Файловое хранилище

В этом меню, как вы понимаете, вы можете указать, где вы хотите хранить файлы на камере безопасности Raspberry Pi Zero W. Вы можете указать этот путь на SD-карте, установленной на плате Raspberry Pi, или установить его на внешний жесткий диск / твердотельный накопитель, подключенный через USB. Наконец, имейте в виду, что вы всегда можете установить этот путь в соответствии с диском, подключенным к сети. Вы можете увидеть параметры, доступные в этом разделе меню, следующим образом:



## Наложение текста

Здесь вы можете установить наложение текста на выходе камеры. Если вы оставите все как есть, левый текст читает имя камеры, а правый текст читает текущую временную метку. На следующем снимке экрана показаны две опции, доступные в меню «Text Overlay - Наложение текста»:

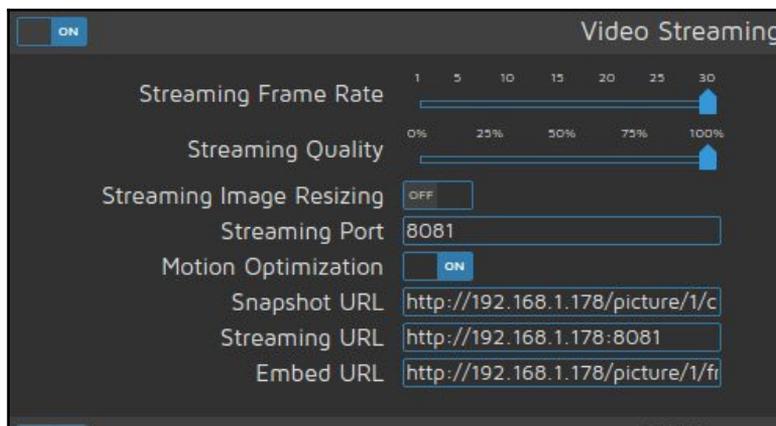


## Видео трансляция

В этом меню вы сможете установить параметры потокового видео. Параметры связаны с видео, которое вы увидите в браузере при подключении к плате Raspberry Pi Zero W. Ниже приведен список параметров, которые необходимы и важны, когда вам нужно снизить качество видео или когда у вас есть другие проблемы с потоковой передачей:

- Streaming Frame Rate - Частота кадров потоковой передачи: точно такая же, как упоминалось ранее в разделе «Video device - Видеоустройство».
- Streaming Quality - Качество потоковой передачи: здесь вам, возможно, придется увеличить или уменьшить качество вашего видео. Опять же, помните, что если у вас есть доступ к потоковому видео с вашей камеры через соединение с низкой пропускной способностью, всегда рекомендуется снижать качество потоковой передачи. Однако, если вам просто нужно сохранить качество видео на жесткий диск, вы можете рассмотреть возможность максимизации качества, поскольку вам не нужно заботиться о проблемах с потоковой передачей.
- Streaming Image Resizing: Изменение размера потокового изображения. Вы можете включить эту функцию, если хотите, чтобы MotionPie изменял размер изображений перед их отправкой в браузер. Но, поскольку мы используем программу MotionPie с нашей платой Raspberry Pi Zero W, делать это не рекомендуется. Итак, пропустите этот вариант и перейдите к следующему.
- Streaming Port - Порт потоковой передачи: это порт, использующийся в своем браузере для подключения к потоковой передаче видео с камеры. Например, вы можете ввести `http://motionpie:8081` в браузере, если ваш потоковый порт — 8081.
- Motion Optimization - Оптимизация движения: эта опция уменьшает частоту кадров, если движение не обнаружено, что может сэкономить много трафика. В некоторых случаях нам нужно максимально уменьшить пропускную способность, поэтому помните об этом параметре.

На следующем изображении показаны все параметры, которые можно настроить и изменить в меню «Video Streaming - Поток видео».

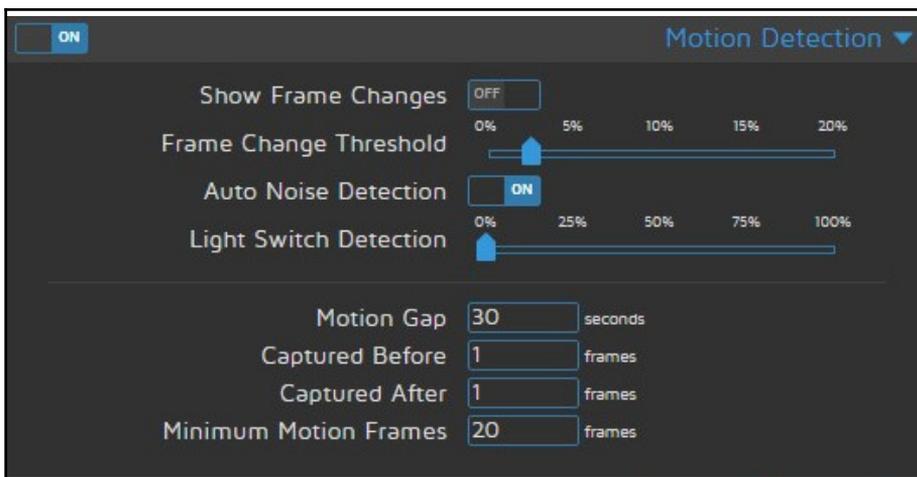


## Неподвижные изображения

В этом меню вы можете настроить камеру безопасности Raspberry Pi Zero W на съемку неподвижных изображений всякий раз, когда срабатывает движение, в определенные промежутки времени или все время; например, вы можете настроить это и снимать изображения каждые 5 минут. В некоторых случаях нам может понадобиться эта опция.

## Обнаружение движения

Это меню очень важно. Здесь вы можете активировать обнаружение движения камеры безопасности Raspberry Pi Zero W, которое включено в программу. С помощью этой программы вы можете обнаруживать движение в области и использовать Slack API, как мы видели в предыдущей главе, для для отправки уведомлений обратно пользователю — вы можете вести файлы журнала о том, что происходит, и многое другое. Внесите все изменения в показанные здесь настройки, чтобы улучшить обнаружение движения:

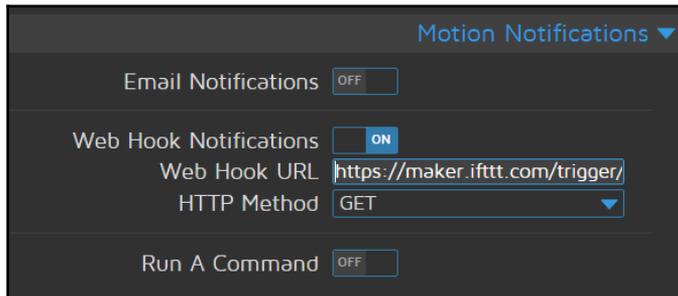


## Кинофильмы

Здесь вы можете настроить плату Raspberry Pi Zero W на запись фильмов при обнаружении движения. Поскольку плата Raspberry Pi Zero W достаточно мала, чтобы ее можно было носить с собой и является портативной, вы можете сделать свой личный самодельный видеомэгнитофон.

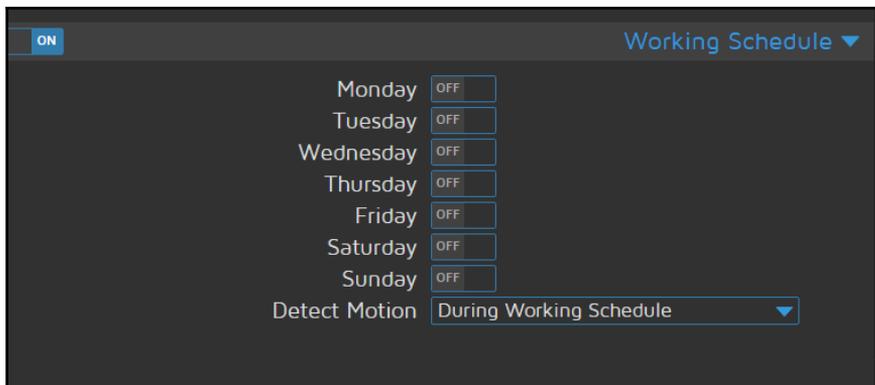
## Уведомления о движении

Помимо [Slack API](#), который предоставляет вам уведомления, это программа позволяет включать и настраивать уведомления по электронной почте, уведомления Webhook или даже запускать команду при обнаружении движения. Это самая важная опция меню, так как она является началом домашней автоматике. Это позволит вам получать уведомления всякий раз, когда на камере обнаруживается активность. Ниже приведено меню, которое отображается в меню программы MotionPie:



## Рабочее расписание

Последний вариант — Working Schedule - рабочий график. Здесь вы можете установить дни и часы работы, которые система должна контролировать; например, вам может понадобиться оставить его на выходные. По умолчанию это 24/7. Этот вариант идеален, если вам нужно, чтобы он работал только в определенные часы. Ниже приведены все параметры, доступные для этой функции в меню MotionPie:



К этому моменту вы должны знать все, что необходимо для настройки программы MotionPie и максимально упростить ее, чтобы максимально использовать ее возможности. Далее мы пройдем установку OpenCV и посмотрим, что еще мы можем сделать.

## Установка OpenCV

OpenCV — это библиотека для компьютерного зрения. Он хорошо известен благодаря распознаванию лиц, отслеживанию видео и другим целям обработки изображений и видео. На данный момент я должен сказать, что если вы хотите использовать и играть с распознаванием лиц, это одна из лучших библиотек, которую вы должны использовать, и ее довольно сложно настроить. Итак, в этой главе мы рассмотрим простой пример того, как вы можете запустить программу для распознавания лиц на изображении. Мы не будем использовать программу MotionPie, поэтому скачайте и установите на SD-карту понравившийся вам дистрибутив. Предполагая, что мы можем включить нашу плату Raspberry Pi Zero W с загруженным дистрибутивом, мы можем начать установку библиотеки, выполнив следующие шаги:

1. Прежде всего, подключите SSH к вашему Raspberry Pi Zero W.
2. Измените текущий каталог и перейдите в свою домашнюю папку. Затем создайте новую папку с именем `opencv_src` и снова измените текущий каталог на папку, которую вы только что создали, следующим образом:

```
cd ~
mkdir opencv_src
cd opencv_src
```

3. Теперь нам нужно получить соответствующий код с GitHub. Мы можем сделать это, клонировав соответствующий репозиторий с помощью следующей команды:

```
git clone
```

4. Затем измените текущий каталог на тот, который вы только что клонировали, с помощью следующей команды:

```
cd opencv/
```

5. Создайте новый каталог с именем `release` и измените текущий каталог с помощью следующей команды:

```
mkdir release
cd release/
```

6. Теперь нам нужно создать и построить библиотеку, которую мы загрузили. Это может занять некоторое время, так что наберитесь терпения. Мы можем сделать это с помощью следующей команды:

```
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local ..
```

7. Теперь выполните команду `make` следующим образом:

```
make
```

8. Установите загруженную нами библиотеку с помощью следующей команды:

```
sudo make install
```



Дополнительную информацию можно найти на официальном сайте OpenCV: [http://docs.opencv.org/2.4/doc/tutorials/introduction/linux\\_instvce/linux\\_install.html](http://docs.opencv.org/2.4/doc/tutorials/introduction/linux_instvce/linux_install.html)

## Распознавание лица

Предполагая, что вы успешно установили OpenCV, мы можем проверить это, просто импортировав `cv` с помощью Python. Откройте терминал и введите `python`, чтобы запустить оболочку, а затем введите следующую команду:

```
import cv2
```

Если на экране нет ошибок, все в порядке. В противном случае вернитесь и переустановите OpenCV, как описано ранее. Теперь давайте разберем фактический код, который вы можете найти и скачать по следующей ссылке или выбрать из прилагаемой папки с кодами:

<https://github.com/shantnu/FaceDetect/>

В сценарии `face_detect.py` изображение `abba.png` и файл `thehaarcascade_frontalface_default.xml` являются следующими. При выполнении файла вы должны передать эти два аргумента:

```
# Get user supplied values
imagePath = sys.argv[1]
cascPath = sys.argv[2]
```

В результате вы сначала передаете изображение, а затем каскадные имена. В следующем примере мы будем использовать изображение музыкальной группы ABBA в качестве исходного входного изображения и каскада по умолчанию для обнаружения почти всех лиц, предоставляемых OpenCV. С помощью следующего кода мы создадим каскад и инициализируем его нашим каскадом лиц по умолчанию:

```
# Create the haar cascade
faceCascade = cv2.CascadeClassifier(cascPath)
```

Этой командой мы просто загружаем в память каскад лиц, чтобы он был готов к использованию в дальнейшем. На этом этапе важно помнить, что каскад — это просто XML-файл, содержащий данные для обнаружения лиц и ничего более. Если у вас есть знания об обработке изображений, вы можете пойти и изменить файл или создать новый:

```
# Read the image
image = cv2.imread(imagePath)
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

С помощью команды `imread` мы можем просто прочитать данное изображение, а затем преобразовать его в оттенки серого. Это важно, потому что многие операции в OpenCV выполняются в оттенках серого, а не в изображениях RGB. Следующая функция определяет лицо и является ключевой частью программы. У нас есть следующие параметры, которые мы можем настроить здесь:

- Во-первых, функция `detectMultiScale` — это общая функция, которая обнаруживает объекты. В зависимости от каскада, который мы предоставляем в качестве входных данных, он обнаружит объект, который описывает каскад:
  - Первый вариант здесь — изображение в градациях серого.
  - Второй — масштабный фактор, и это важно, так как одни лица находятся глубже на изображении, а другие — ближе. Важно настроить масштабный коэффициент, чтобы программа знала, насколько велика искомая грань.
  - Далее важно знать, что алгоритм обнаружения использует движущееся окно для обнаружения объектов:
    - `minNeighbors` определяет, сколько объектов будет обнаружено рядом с текущим, прежде чем он выведет и объявит, что лицо было найдено.

- Кроме того, `minSize` на самом деле является размером каждого используемого окна. Вы можете поэкспериментировать с разными значениями и выбрать то, что лучше всего подходит для вас. Таким образом, функция возвращает список прямоугольников, как вы можете видеть на следующем изображении, где она считает, что нашла лицо. Могут быть некоторые ложные срабатывания, но мы можем исправить их с помощью параметра `ScaleFactor`. Далее мы пройдемся по тому месту, где, по его мнению, он что-то нашел.

```
print "Found {0} faces!".format(len(faces))

# Draw a rectangle around the faces
for (x, y, w, h) in faces:
    cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2)
```

В результате у нас возвращаются четыре значения. `X` и `Y` — это расположение прямоугольника, а `W` и `H` — ширина и высота прямоугольника. Мы можем использовать эти четыре значения, чтобы просто создать прямоугольник, используя функцию построения библиотеки. Итак, делаем это с помощью следующей команды:

```
cv2.imshow("Faces found" , image)
cv2.waitKey(0)
```

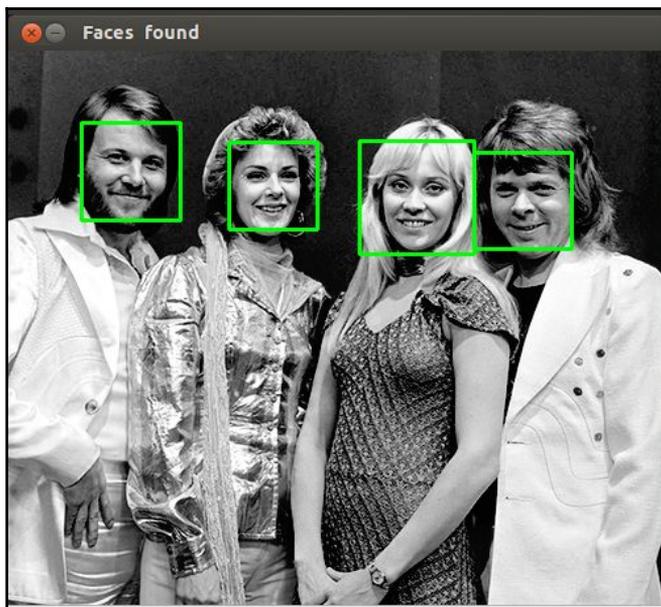
Наконец, мы отображаем окончательное изображение после вставки прямоугольников. Чтобы проверить это, выполните следующую команду:

```
python face_detect.py abba.png haarcascade_frontalface_default.xml
```

Может возникнуть проблема с этой командой. Если это не работает, вам может потребоваться открыть файл `face_detect.py` и заменить строку следующим образом:

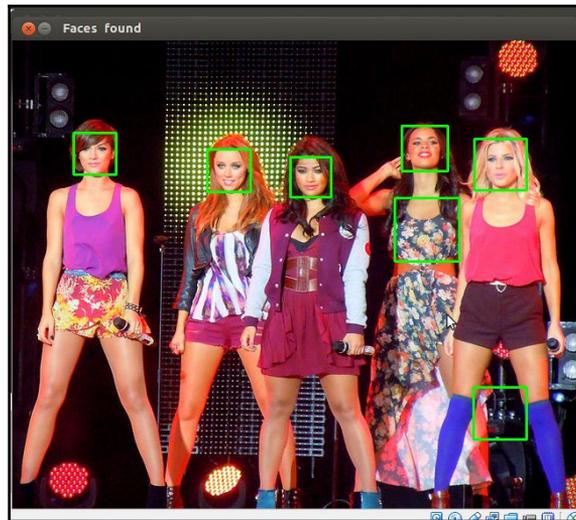
```
Flags = cv2.cv.CV_HAAR_SCALE_IMAGE
Flags = cv2.CASCADE_SCALE_IMAGE
```

В результате получится следующее изображение с добавленными зелеными прямоугольниками:



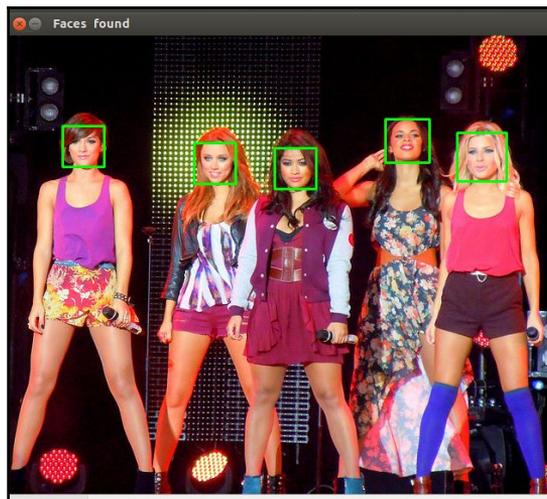
Источник: <https://github.com/shantnu/FaceDetect/>

Вы можете проверить это на своих собственных изображениях и посмотреть, что обнаружит программа. Качество изображения имеет решающее значение для правильной работы этой функции. Кроме того, правильный масштабный фактор должен быть изменен в соответствии с размером лиц на изображении. Вы всегда должны помнить, что результаты никогда не будут точными на 100%. Однако в большинстве случаев вы получите достаточно хорошие результаты! Если мы проверим это с другим изображением, как показано ниже, вы увидите, что оно обнаруживает несуществующие лица:



Источник: <https://github.com/shantnu/FaceDetect/>

Если вы измените масштабный коэффициент на 1,2, вы получите соответствующий ответ от программы, как вы можете видеть на следующем изображении:



Источник: <https://github.com/shantnu/FaceDetect/>

## Резюме

В этой главе мы рассмотрели установку программы MotionPie, которое легко установить и использовать. Вы должны иметь возможность подключить свою камеру и добавить Raspberry PiCamera в свои проекты. Далее мы шаг за шагом рассмотрим, как вы можете создать свои собственные портативные колонки с помощью новой платы Pi.

# 7

## Портативные колонки

В предыдущей главе мы шаг за шагом рассмотрели разработку камеры видеонаблюдения и подключение нескольких камер к нашей плате Raspberry Pi Zero W. В этой главе мы создадим что-то другое. Raspberry Pi Zero W полезен не только для проектов по обеспечению безопасности, но и для других решений DIY. Итак, здесь мы увидим, как создать свои персональные портативные колонки. После завершения этого проекта вы сможете подключить свои портативные колонки и слушать собственную музыку или расширить уже существующие проекты, которые вы должны были разработать. Наконец, мы добавим еще одну функцию, которая позволит вам воспроизводить музыку в зависимости от вашего настроения. Поэтому в этой главе мы рассмотрим следующие разделы:

- Рынок колонок
- Настройка программ
- Сеть
- Музыка на Pi
- Настройка колонки
- Выбор настроения

### Рынок колонок

Если вы будете искать колонки на рынке, вы найдете множество различных модулей, доступных для покупки. Raspberry Pi называет эти модули расширениями, и в зависимости от вашего проекта, вы можете купить соответствующее расширение и упростить ваш проект, поскольку многие вещи уже реализованы. Это означает, что вам не нужно кодировать и разрабатывать все с нуля. Иногда стоит потратить время на поиск модулей и подобных проектов. Динамики являются одним из них, и вы можете купить их непосредственно для своего Raspberry Pi.

К сожалению, Raspberry Pi Zero W не имеет аудиовхода/выхода. Итак, есть два способа воспроизведения музыки, а именно:

- From HDMI
- From Bluetooth or Wi-Fi

Есть несколько руководств по воспроизведению звука через порт HDMI, но в этом разделе мы рассмотрим воспроизведение звука через Bluetooth или Wi-Fi. Одной из популярных колонок на рынке являются портативные мини-колонки для Raspberry Pi, которые вы можете найти прямо в магазине Pi Hut. Это недорогие динамики, которые отлично работают с платами Raspberry Pi. Ниже приведено изображение того, как они выглядят:



Он работает от 5 В и может воспроизводить аудио около семи часов. Итак, с чем-то вроде этого мы можем создать вашу портативную колонку.



Более подробную информацию об этом спикере можно найти на <https://thepihut.com/products/mini-portable-speaker-for-the-raspberry-pi?variant=897680793>.

Конечно, имейте в виду, что им нужна батарея, поэтому либо возьмите с собой блок питания, либо купите какой-нибудь аккумулятор.

## Настройка программы

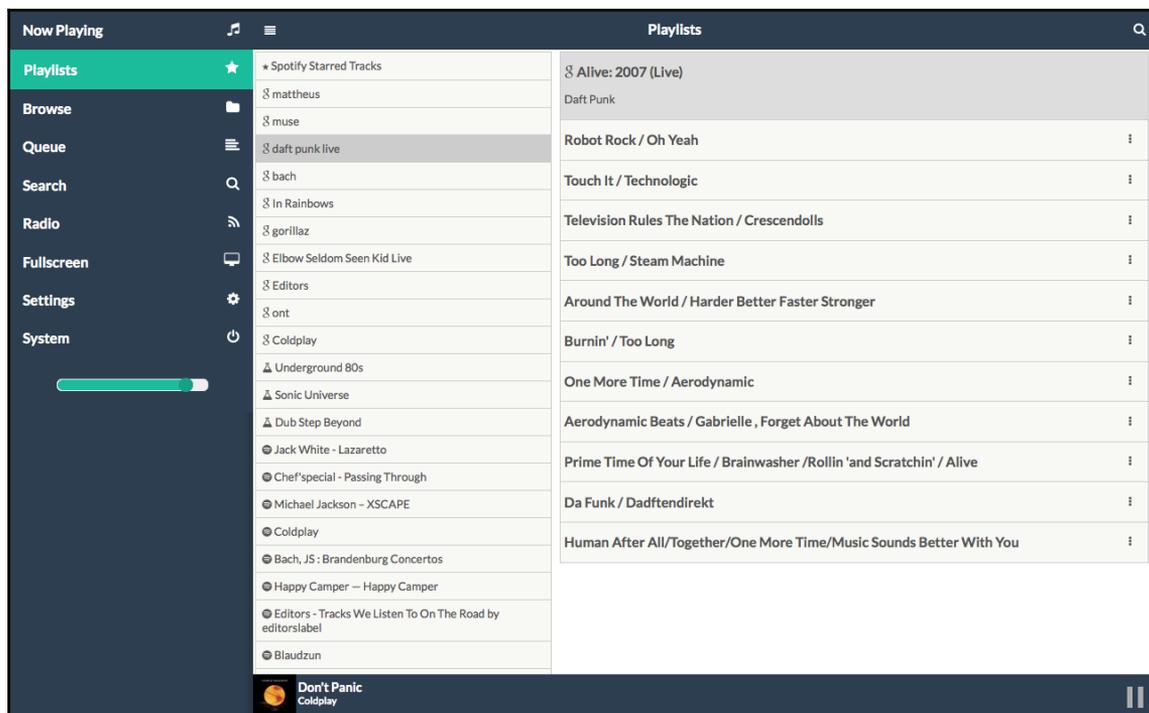
Прежде чем мы начнем соединять все вместе на нашей плате Raspberry Pi Zero W, нам нужно записать новую SD-карту с соответствующим образом. Итак, скачайте zip-архив Pi Music Box с официального сайта по следующей ссылке: <http://www.pimusicbox.com/>. Теперь вам нужно установить эту программу на свою SD-карту, поэтому используйте инструмент `dd` или что-то еще, с чем вы знакомы, в зависимости от вашей операционной системы, и выполните следующие шаги:

1. Вставьте SD-карту в плату Raspberry Pi Zero W. Если вы используете внешний ключ Wi-Fi (по какой-либо причине), вы можете отредактировать файл `config settings.ini`. Расположение файла находится на SD-карте, которую вы только что записали. Предполагая, что ваша плата оснащена Wi-Fi и Bluetooth, вы пропускаете предыдущий шаг, но для тех, кто предпочитает трудный путь или имеет плату Raspberry Pi Zero (которая очень похожа), ему придется отредактировать имя пользователя и пароль настроек Wi-Fi.
2. Затем вам необходимо предоставить доступ в Интернет для вашего Raspberry Pi Zero W.
3. Нам нужно найти IP-адрес платы Raspberry Pi Zero W. После этого нам нужно подключиться по `ssh` к этому IP-адресу через порт 22, и вам будет предложено ввести имя пользователя и пароль. Введите имя пользователя и пароль по умолчанию, так как вы их не меняли, и вы уже должны быть в Pi.
4. Предполагая, что все настроено, попробуйте открыть браузер и перейти по следующему адресу: <http://musicbox.local>.

Если по какой-либо причине у вас возникли проблемы с предыдущими шагами, вам нужно будет использовать IP-адрес Raspberry Pi. Помните, что вы узнали, как получить и найти этот IP-адрес в предыдущих главах. При отображении Pi, вы можете найти IP-адрес из командной строки, введя следующую команду:

```
ifconfig
```

После подключения в браузере вы должны увидеть экран, подобный показанному на следующем снимке экрана:



Теперь важно понимать, что мы будем транслировать это по локальной сети, а не через Интернет. Для потоковой передачи или подключения устройств через Интернет требуется больше действий для настройки. Мы не будем рассматривать все это в этой главе. Не забудьте обновить свою систему при установке последней версии дистрибутива Raspberry Pi. Вы можете обновить свою систему, введя следующую команду:

```
sudo apt-get update
```

Вы можете обновить все последние пакеты, используя следующую команду:

```
sudo apt-get upgrade
```

На этом этапе у вас должна быть новейшая система, которая может подключаться к динамике и воспроизводить вашу любимую музыку.

## Сеть

В этом разделе мы рассмотрим некоторые настройки, которые мы можем изменить, чтобы они работали, и настроим их в нашей локальной сети. После этого плеер MusicBox должен работать в вашей сети и все должно быть подключено. Посмотрим на следующие настройки:

- **Network name:** это имя на самом деле является SSID или именем вашей сети Wi-Fi. В большинстве случаев его можно изменить в интерфейсе роутера, посетив URL-адрес <http://192.168.1.1>. Если нет, проверьте в руководстве соответствующий URL-адрес вашего роутера
- **Wi-Fi password:** очевидно, это поле, в котором вы должны разместить пароль Wi-Fi вашей сети. Еще раз, пароль можно изменить, перейдя по ссылке, которую мы посетили ранее. По многим причинам важно всегда добавлять надежный пароль, который включает буквы, заглавные буквы, символы и цифры в случайном порядке.
- **Workgroup:** это поле представляет собой имя рабочей группы вашей локальной сети Windows. При подключении компьютера с Windows к сети вам будет предложено выбрать группу, к которой вы будете принадлежать.
- **SSD:** как мы уже знаем, `ssh` всегда полезен по многим причинам, поэтому рекомендуется включить его для последующих модификаций. С другой стороны, если вы хотите просто завершить проекты и ничего не менять в плате *Raspberry Pi Zero W*, вы всегда можете оставить все как есть.

## MusicBox

В этом разделе мы рассмотрим некоторые настройки и упомянем несколько советов для вашего MusicBox. Возможно, вам не нужно менять многое, но всегда полезно знать, какие варианты у нас есть:

- **Device name:** это имя вашей платы *Raspberry Pi Zero W* в вашей локальной сети. Имя по умолчанию — MusicBox, поэтому, если по какой-либо причине вы хотите изменить его, продолжайте. Обратите внимание, что URL-адрес, который мы использовали, имел имя MusicBox, поэтому, если вы измените его, вам нужно будет посетить другой URL-адрес в соответствии с именем.
- **Autoplay URL:** URL-адрес автозапуска: здесь вы можете вставить URL-адрес, чтобы плата *Raspberry Pi Zero W* могла начать воспроизведение радиостанции или потокового вещания при загрузке устройства. Учтите, что вам, возможно, придется немного увеличить время ожидания, чтобы процесс загрузки завершился до запуска MusicBox.

- **Wait time:** это время ожидания MusicBox перед началом воспроизведения URL-адреса. Как мы упоминали ранее, разумно немного увеличить это значение. Помните, что числа указаны в секундах. 20-секундная задержка будет хорошим выбором.
- **Root password:** Пароль root по умолчанию — MusicBox, но вам следует подумать о том, чтобы изменить его на что-то более безопасное. Сделать это практически необходимо, особенно если вы собираетесь включить доступ по SSH.
- **AirPlay streaming:** здесь мы можем выполнять прямую трансляцию с мобильного телефона на музыкальный сервер Raspberry Pi. Если вы включите эту опцию, она должна отобразиться в списке устройств Airplay.
- **DLNA/uPnP/OpenHome streaming:** поскольку нам может потребоваться потоковая передача MusicBox по некоторым протоколам, мы можем включить соответствующий из этого раздела.

## Аудио

Что касается звука, здесь есть несколько вариантов. Начальная громкость — это громкость устройства по умолчанию при его запуске. Затем аудиовыход является выходом аудио. Здесь вы можете вручную выбрать выход для звука. Наконец, можно включить USB-диск с загрузкой, если у вас есть какие-либо проблемы с качеством звука при воспроизведении. Обычно на старых платах может потребоваться включить это. Последние просто прекрасны и не нуждаются в этой опции.

## Музыкальные файлы

Здесь мы увидим вариант музыкальных файлов, которые мы можем воспроизводить, и то, как мы можем связать список воспроизведения с программой:

1. Во-первых, Scan Music Files можно включить для сканирования любых файлов при загрузке, но это замедлит процедуру.
2. Затем Network Drive - сетевой диск, который может быть здесь лучшим вариантом, требует, конечно, наличия настройки сетевого диска, на котором есть вся ваша музыка, но вы можете легко подключить его к MusicBox и воспроизводить музыку прямо с него. Одна вещь, которая обычно идет не так, - это настройка используемого имени пользователя и пароля. Итак, взгляните на это еще раз в случае каких-либо проблем.
3. Наконец, параметр «Resize Filesystem - Изменить размер файловой системы» очень важен, поскольку при автоматической перезагрузке он дает нам доступ ко всей SD-карте.

## Музыкальные онлайн-сервисы

Теперь давайте просто перейдем к другой теме музыкальных онлайн-сервисов, таких как Spotify или YouTube. Существуют сотни сервисов, таких как [Spotify](#), [YouTube](#) или даже [SoundCloud](#), которые можно подключить к [MusicBox](#), и каждый из них довольно прост. Мы не будем рассматривать их в этой главе, но, как всегда, если вы хотите расширить свой список воспроизведения и подключить одну из упомянутых ранее служб к вашему Raspberry Pi Zero W, вы можете получить дополнительную информацию в Google.

## Безопасность

Имейте в виду, что [MusicBox](#) может быть хорошим вариантом, но он недостаточно безопасен, чтобы работать вне брандмауэра. Поэтому держите локальный брандмауэр включенным и всегда защищайте себя надежным паролем. Наконец, есть еще один вариант, который мы можем рассмотреть для доступа. Итак, если вы хотите получить доступ к графическому интерфейсу [mopify](#) вместо графического интерфейса Pi [MusicBox](#), просто зайдите в свой браузер и введите:

```
musicbox.local/mopify
```

Снова замените [musicbox.local](#) на IP-адрес Raspberry Pi. [Mopify](#) — это графический интерфейс, разработанный только для [Spotify](#), поэтому, если вы хотите использовать другие сервисы, вам нужно будет использовать графический интерфейс музыкальной шкатулки Pi.

Я надеюсь, что теперь у вас есть полностью работающий музыкальный проигрыватель Raspberry Pi.

## Звуковые проигрыватели

Поскольку мы много раз подключались к нашей плате Raspberry Pi Zero W через ssh, важно знать, как управлять громкостью и любым возможным звуком с терминала. Когда нет графического пользовательского интерфейса или графического интерфейса, у нас не так много возможностей для управления громкостью звука. Хорошей идеей является использование команды под названием [alsamixer](#), которая позволяет нам легко изменять и иметь полный контроль над входными/выходными звуками на нашей плате Raspberry Pi Zero W или нашем устройстве Linux. Обычно он предустановлен в вашей системе, но если его нет, вы всегда можете найти его с помощью следующей команды:

```
aptitude search alsa
```

Таким образом, он будет перечислять ваши пакеты, которые вы можете установить аналогично alsa. Команда aptitude полезна для поиска похожих пакетов. Вот результат поиска alsa:

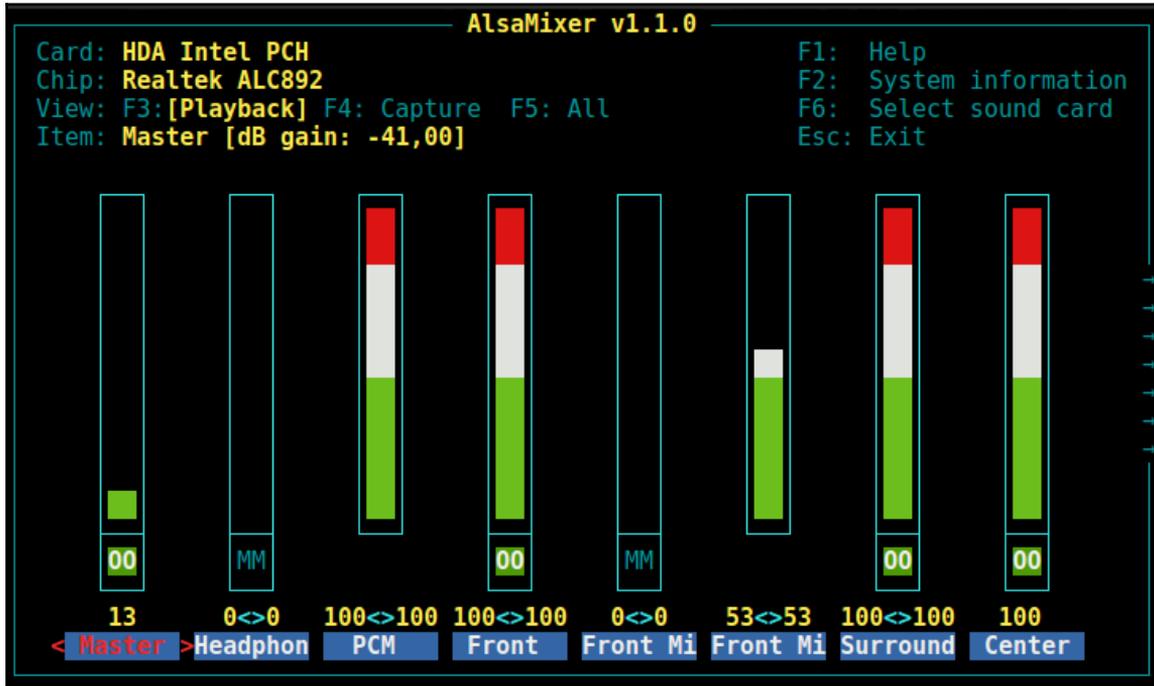
```
mark@zeus ~ $ aptitude search alsa
v   alsa -
v   alsa:i386 -
i   alsa-base - ALSA driver configuration files
v   alsa-base:i386 -
p   alsa-firmware-loaders - ALSA software loaders for specific hardware
p   alsa-firmware-loaders:i386 - ALSA software loaders for specific hardware
p   alsa-oss - ALSA wrapper for OSS applications
p   alsa-oss:i386 - ALSA wrapper for OSS applications
p   alsa-source - ALSA driver sources
p   alsa-tools - Console based ALSA utilities for specific hardware
p   alsa-tools:i386 - Console based ALSA utilities for specific hardware
p   alsa-tools-gui - GUI based ALSA utilities for specific hardware
p   alsa-tools-gui:i386 - GUI based ALSA utilities for specific hardware
i   alsa-utils - Utilities for configuring and using ALSA
p   alsa-utils:i386 - Utilities for configuring and using ALSA
p   alsamixergui - graphical soundcard mixer for ALSA soundcards
p   alsamixergui:i386 - graphical soundcard mixer for ALSA soundcards
v   alsaplayer -
v   alsaplayer:i386 -
p   alsaplayer-alsa - alsaplayer output module for ALSA
p   alsaplayer-alsa:i386 - alsaplayer output module for ALSA
p   alsaplayer-common - audio player (common files)
```

Теперь давайте воспользуемся alsamixer для управления громкостью вашего компьютера. Вы можете ввести в своем терминале следующее, чтобы запустить alsamixer:

## Alsamixer

Теперь вы увидите что-то вроде следующего в зависимости от устройств, подключенных к вашей системе Linux.

Вы можете выйти из alsamixer, нажав клавишу Q. Теперь у вас есть еще один инструмент в ваших руках при игре со звуком, чтобы управлять им через терминал и без какого-либо монитора с графическим интерфейсом пользователя:



## Подключение динамика

На этом этапе мы должны подключить ваши динамики к плате Raspberry Pi Zero W. Самый простой способ сделать это — перейти в меню «Program - Программы», затем «Preferences - Настройки» и щелкнуть «Bluetooth Manager - Диспетчер Bluetooth». Ваши динамики Bluetooth должны всегда искать синхронизацию с устройством. Предполагая, что ваш динамик ожидает синхронизации, нажмите на поиск, и вы попадете в интерфейс. Теперь вы сможете соединить два устройства. Если возникнут какие-либо проблемы, вам следует снова нажать «Connect again - Подключиться» и дождаться подключения. После этого все должно быть в порядке, и вы можете проверить это, воспроизведя песню на YouTube.

## Музыка на Pi

Наша плата Raspberry Pi Zero W представляет собой компьютер с Linux, который позволяет нам делать почти все, что мы хотим, поскольку перед нами был настольный компьютер с Linux. Итак, чтобы загрузить музыку на наш Raspberry Pi Zero W, мы можем использовать старомодный способ использования USB-устройств и передавать огромное количество песен и плейлистов. С другой стороны, мы можем использовать следующую команду для прямой загрузки песен с YouTube:

```
Youtube-dl <youtube-link>
```

Если вы видите какие-либо сообщения об ошибках, возможно, это связано с тем, что у вас не установлен youtube-dl. Вы можете легко сделать это, введя следующую команду:

```
sudo apt-get install youtube-dl
```

Теперь вы должны быть в порядке и готовы загружать музыку из своей учетной записи YouTube непосредственно на Raspberry Pi Zero W. Программа youtube-dl имеет множество опций, и вы можете просмотреть подробную информацию о нем на странице руководства. Чтобы открыть и прочитать любую страницу руководства в системе Linux, введите следующую команду:

```
man <command>
```

В нашем случае <command> - это youtube-dl.

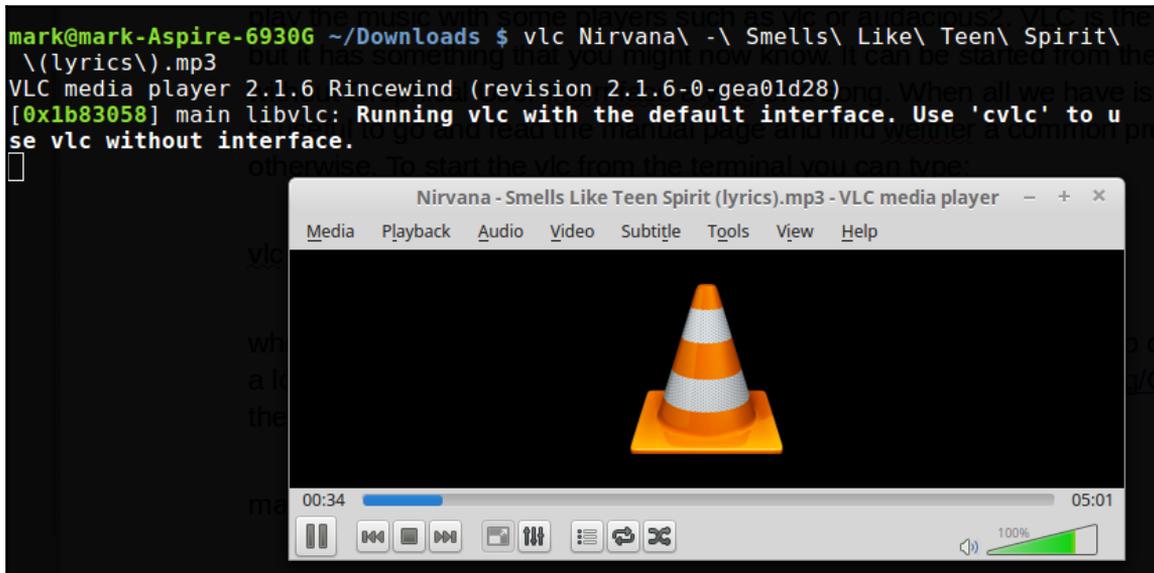
С другой стороны, если вы каким-либо образом переносите свою музыку на Raspberry Pi Zero W, вы можете воспроизводить музыку с помощью некоторых проигрывателей, таких как vlc или Audacious. VLC — известный видеоплеер, но в нем есть кое-что, что вы, возможно, уже знаете. Его можно запустить из терминала и играть без графического интерфейса пользователя. Когда все, что у нас есть, это терминал, иногда полезно пойти и прочитать страницу руководства и выяснить, можно ли использовать обычную программу иначе. Чтобы запустить vlc из терминала, вы можете ввести следующую команду:

```
vlc <options>
```

Здесь параметры могут относиться к песне, которую вы хотите воспроизвести, видео или даже плейлисту. Например, чтобы запустить vlc с графическим интерфейсом и воспроизвести песню, введите следующую команду:

```
vlc <song name>
```

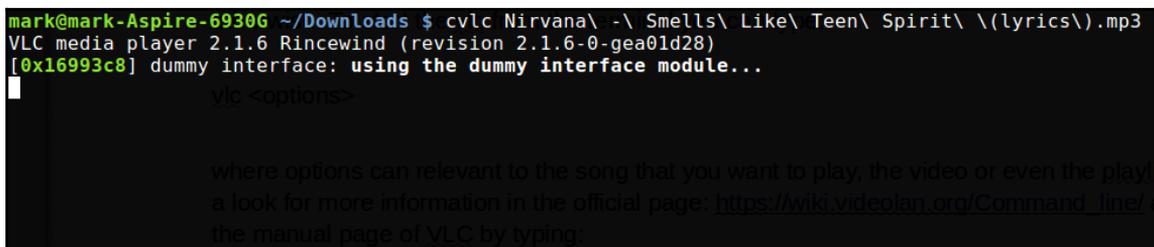
Результат предыдущей команды выглядит следующим образом:



Однако, если вы хотите запустить `vlc` только с помощью терминала и без открытия графического интерфейса, вы можете ввести следующую команду:

```
cvlc <song>
```

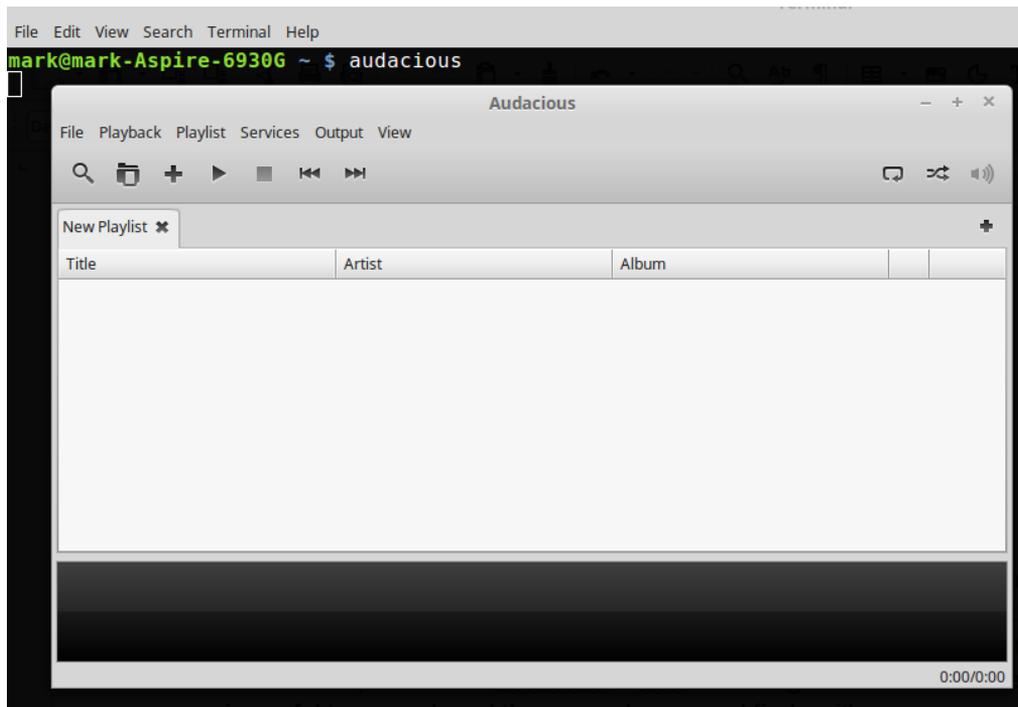
Результат предыдущей команды выглядит следующим образом:



Для получения дополнительной информации посетите официальную страницу [https://wiki.videolan.org/Command\\_line/](https://wiki.videolan.org/Command_line/) и на странице руководства VLC, введя следующую команду:

```
man vlc
```

Наконец, Audacious - еще один классный музыкальный проигрыватель с графическим пользовательским интерфейсом или графическим интерфейсом, способный менять различные темы. На следующем снимке экрана показано, как будет выглядеть тема по умолчанию:



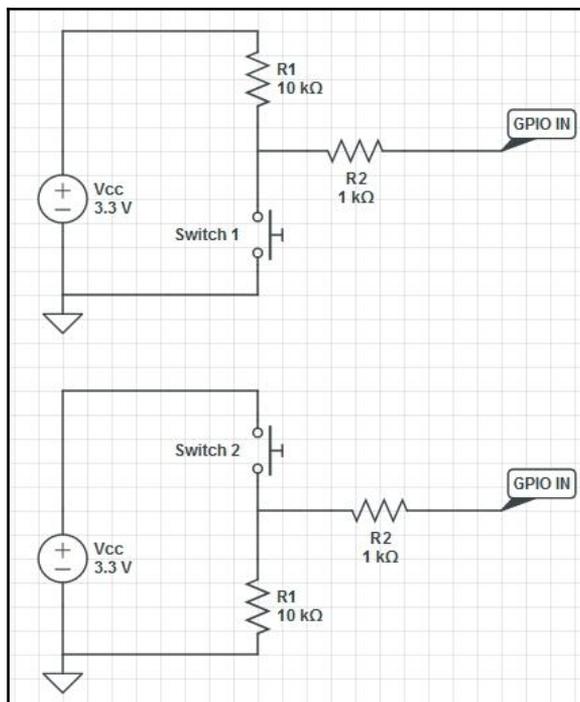
## Выбор настроения

К настоящему времени у вас должна быть полная система, которая воспроизводит музыку и подключена к вашей плате Raspberry Pi Zero W. Теперь, чтобы сделать его более умным, давайте добавим систему выбора настроения, которая воспроизводит соответствующий музыкальный плейлист в соответствии с вашим настроением. Для этого нам сначала нужно создать несколько плейлистов, которые будут соответствовать нашему настроению. Например, давайте определим первый список воспроизведения как Acoustic Playlist, а второй — как Rock Playlist. Мы можем сделать это разными способами, один из которых — создать две папки с соответствующими именами и поместить в каждую из них несколько песен. Используя `svcs`, о котором мы упоминали ранее, мы сможем подключить наш музыкальный проигрыватель к любому скрипту. Так что остается триггерная система.

Триггерная система будет реализована в соответствии с имеющимся у вас оборудованием. Самая простая система выбора настройки, которую мы можем создать, будет состоять из двух кнопок, каждая из которых будет запускать соответствующий список воспроизведения. Итак, сначала нам нужно соединить две кнопки на нашей плате Raspberry Pi Zero W, а затем написать некоторый код Python, чтобы определить, какая кнопка была нажата и какой плейлист начнет воспроизведение.

## Подключение компонентов

Что касается подключения, в соответствии с распиновкой Raspberry Pi Zero W, нам нужно подключить каждую кнопку к контакту. Две кнопки должны быть подключены к контактам 17 и 27 в соответствии со следующей схемой:



## Код

Следующий код является скриптом Python и может быть выполнен после сохранения его как `script.py` путем ввода следующей команды в консоли Python:

```
python script.py
```

Теперь давайте пройдемся по следующим строкам и посмотрим, что у нас получилось. Во-первых, нам нужно импортировать библиотеку времени, чтобы позже мы могли приостановить цикл `while`:

```
import time
```

Далее нам нужно инициализировать предыдущую входную переменную в 0. Здесь мы предполагаем, что кнопка не была нажата. Мы также создаем новую переменную с именем `state`, которая указывает состояние кнопки. 0 означает, что кнопка отпущена, а 1 означает, что она нажата:

```
prev_input = 0
state = 0
```

Затем мы создаем цикл `while` и считываем два входных контакта GPIO:

```
while True:
    input = GPIO.input(17)
    input = GPIO.input(27)
```

После этого нам нужно проверить состояние ввода и если кнопка была нажата, то нам нужно проверить состояние нажато или отпущено. Если он был нажат, мы меняем его на отпущенный, а если он был отпущен, мы меняем его на нажатый:

```
if ((not prev_input) and input):
    print("Button pressed")
    if(state == 0):
        state = 1
        // код для воспроизведения первого playlist:
        state = 0
        // код для воспроизведения второго playlist:
#update previous inputprev_input = input#slight pause to debounce
time.sleep(0.05)
```

Теперь код, который нам нужно добавить к предыдущему блоку кода, состоит из одной или двух строк, чтобы мы могли воспроизвести список воспроизведения. Например, мы можем добавить следующий код state1:

```
vlc
```

Далее мы можем добавить state2:

```
vlc song2.mp3
```

## Резюме

В этой главе мы увидели, как можно создать портативные колонки, чтобы мы могли слушать нашу любимую музыку где угодно. Поскольку музыка меняет настроение, в последних разделах был добавлен интерактивный способ понимания настроения читателя и воспроизведения определенных списков воспроизведения. Кроме того, мы увидели, как можно управлять громкостью и звуком вашей Linux-системы через терминал. Наконец, мы упомянули некоторые музыкальные и видеоплееры, необходимые для повседневного аудио- или видео-взаимодействия с Raspberry Pi Zero W.

В следующей главе мы рассмотрим разработку небольшой службы хостинга в нашем доме, используя только беспроводную плату Pi.

# 8

## Хостинг WebPi

В предыдущей главе вы узнали, как создать несколько портативных динамиков с помощью платы Raspberry Pi Zero W. В этой главе вы пройдете наиболее важную тему при разработке веб-сайта и размещении всех файлов на вашем Raspberry Pi Zero W. Вы получите общее представление о взаимодействии клиент-сервер в отношении услуг веб-хостинга, некоторые базовые знания о веб-разработке и способ, которым вы должны получать доступ и передавать свои файлы с вашего локального компьютера на Raspberry Pi Zero W. Наконец, эта глава завершится некоторыми советами по безопасности и проблемами, с которыми вы можете столкнуться при размещении ваших файлов дома. Подводя итог, в этой главе мы рассмотрим следующие темы:

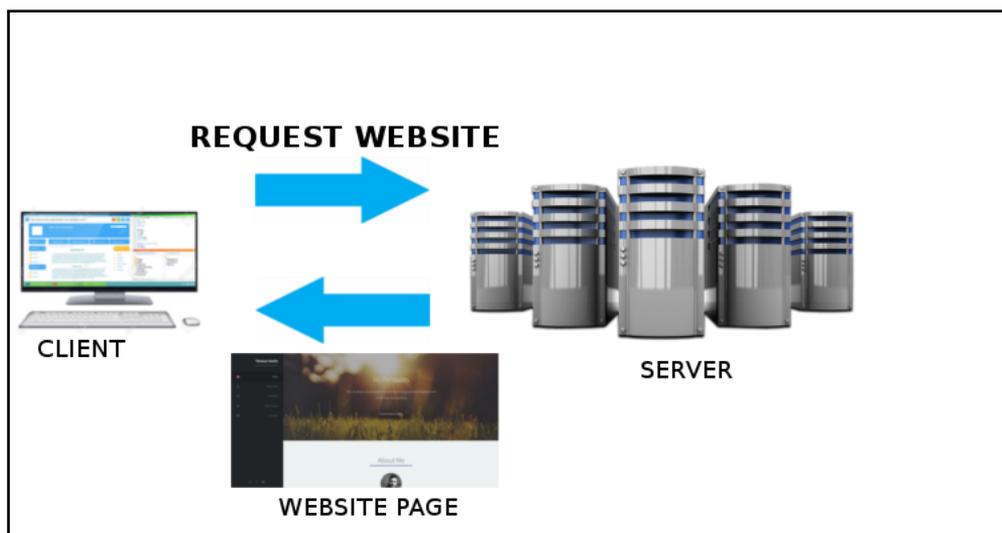
- Взаимодействие клиент-сервер
- веб хостинг
- Сетевое администрирование
- Веб-дизайн и разработка

### Веб хостинг

В этом разделе мы определим, что такое веб-хостинг и что нужно сделать, чтобы получить свою личную услугу веб-хостинга. Вы должны рассматривать это как введение к остальной части главы; поэтому, как всегда, первым шагом является определение того, что именно мы здесь делаем и что мы должны делать, если у нас нет под рукой Raspberry Pi Zero W. Существует несколько хостинговых услуг, предоставляющих различные решения в зависимости от потребностей заказчика. В оставшейся части раздела мы также опишем возможные варианты, которые есть у клиента, а также преимущества или недостатки некоторых из них.

## Определение

Веб-хостинг — это средство, с помощью которого вы можете разместить свои файлы или веб-сайт на компьютере, где вы или любой другой человек имеете к нему доступ со всего мира; например, у интернет-магазина есть веб-сайт, размещенный на сервере, и вы можете получить доступ к серверу и взаимодействовать с ним через браузер или мобильный телефон. Таким образом, веб-хостинг больше касается серверной части, где вы можете сохранять свои файлы и предоставлять их всем, кто их запрашивает. Чтобы было легче это понять, давайте посмотрим на следующее изображение, где клиент (компьютер) запрашивает веб-сайт с помощью своего браузера, а сервер отвечает веб-сайтом:



Как видно на предыдущем изображении, клиент запрашивает веб-сайт с сервера, и сервер передает эту веб-страницу клиенту; таким образом, у нас есть два направления связи и две стрелки. Пользователь может использовать настольный ПК или мобильный телефон; он также может использовать Firefox, Google Chrome или любой другой браузер. Сервер обычно представляет собой большое серверное здание, похожее на предыдущее изображение, но также может быть небольшой платой Raspberry Pi Zero W карманного размера. Важно то, что у нас всегда есть клиент, который запрашивает вещи, и сервер, готовый ответить, если, конечно, клиент авторизован для получения этих файлов. В этой главе клиентом будем мы или кто угодно в мире, а сервером будет наша плата Raspberry Pi Zero W. Теперь давайте посмотрим на некоторые варианты, которые мы обычно имеем при покупке пакета хостинга.

## Услуги хостинга

Обычно у крупных хостингов есть более одного решения. Это потому, что у всех нас разные потребности. У людей разные потребности; таким образом, почти у всех разные пакеты; например, у вас может быть несколько МБ общего пространства или неограниченное пространство и одна база данных, или дополнительные функции, такие как установка WordPress в один клик и многое другое. В зависимости от ваших потребностей вы, очевидно, выбираете услугу с наименьшей стоимостью для вашей компании или ваших личных потребностей. Далее вы можете увидеть изображение от Top Host, показывающее услуги хостинга, которые они предоставляют:

The screenshot shows the Top.Host website with a navigation menu (Hosting, Domain Names, Grid CDN, Dedicated Servers, SSL, Support) and a main banner for 'Elastic Web Hosting Speed & Security Optimized'. Below the banner are four hosting plans:

- Star** (blue): For your personal website. Features: Unmetered space & traffic, Standard resources, SSD storage, 1 dynamic site, 24/7 Extreme Support, Speed Optimized, Advanced Security, Email Protection, Free Benefits.
- Planet** (blue): For developers, designers & SMBs. Features: Unmetered space & traffic, High resources, SSD storage, 5 dynamic sites, 24/7 Extreme Support, Speed Optimized, Advanced Security, Email Protection, Free Benefits.
- Galaxy** (pink): For the needs of web professionals. Features: Unmetered space & traffic, Extreme resources, NVMe storage, Unlimited δυναμικά sites, 24/7 Extreme Support, Speed Optimized, Advanced Security, Email Protection, Free Παροχές.
- Big Bang** (blue): Ideal for demanding sites. Features: For demanding sites that require high availability. Hosting on semi-dedicated servers with NVMe disks for even higher speed, spam protection and VIP 24/7 support. Site backup 3 times a day and continuous uptime monitoring.

Как видите, вариантов четыре: в первом у вас один динамический сайт; во втором пять динамических сайтов; в третьем - неограниченный; а в последнем у вас есть все. Очевидно, что не каждая хостинговая компания предоставляет все услуги с образа, но у всех у них одинаковая логика. Вообще говоря, если вы хотите разместить некоторые файлы или веб-сайт общим размером < 500 МБ, у вас все будет хорошо, и стоимость не будет слишком большой. Однако, если вы хотите разместить веб-сайт для компании, такой как интернет-магазин или что-то подобное, вам необходимо инвестировать в хороший сервер, такой как предыдущий. Должно быть ясно, что когда у вас есть компания, вы не размещаете свой интернет-магазин на плате Raspberry Pi Zero W. Это может работать, но у вас нулевая надежность или поддержка пользователей.

## Доменное имя

Еще одна вещь, которую следует учитывать перед завершением покупки, — это выбор доменного имени. Как мы обсудим в следующих разделах, помимо веб-хостинга, нам нужно доменное имя, чтобы люди могли запомнить наш веб-сайт. Серверы и, как следствие, веб-сайты имеют IP-адрес. Это означает, что сайт facebook.com имеет IP-адрес, например 157.240.9.35. Представьте, если бы не было facebook.com, и всем пользователям приходилось бы набирать 157.240.9.35 каждый раз, когда они хотели использовать веб-сайт. Невозможно, чтобы Интернет мог иметь такие размеры, как сегодня, используя такое взаимодействие с серверами и запрашивая веб-сайты. В заключение, это причина, по которой мы придумали доменные имена; у каждого веб-сайта есть доменное имя, которое, очевидно, легче запомнить, поскольку оно состоит из букв. Выбор доменного имени может быть простым, но есть некоторые вещи, которые вы должны рассмотреть, прежде чем платить за него. Как мы уже говорили, доменное имя — это что-то похожее на facebook.com, linkedin.com, vasilistzivaras.gr или что-то еще, что делает ваш сайт уникальным в мире. Не существует доменного имени, которое соответствует двум веб-сайтам. Прежде всего, важно знать, что доменное имя очень важно для поисковой оптимизации (SEO). SEO означает, что название веб-сайта является переменной в алгоритме поиска Google. Если мы не хотим делать его доступным для Интернета и нам нужен веб-сайт для личного пользования, это нормально, и вам, вероятно, не нужно покупать доменное имя, так как я думаю, что вы можете запомнить IP-адрес. Однако, если вы создаете веб-сайт, который будет доступен для всего Интернета, и добавляете его в список результатов поисковой системы Google, тогда URL-адрес или доменное имя должны быть чем-то, что действительно представляет ваш веб-сайт; например, глупо иметь сайт об одежде с доменным именем «catsanddogs.com». На данный момент я должен сказать, что многие хостинг-провайдеры предоставляют вам бесплатный хостинг для небольшого веб-сайта при покупке доменного имени. Именно так я лично разработал свой сайт vasilistzivaras.gr.

## DNS

Кроме того, важно ввести еще один термин, относящийся к веб-хостингу и доменным именам. Нам не нужно ничего покупать здесь, но очень важно знать, что делает DNS. Мы говорили о веб-хостинге, поэтому есть компьютер, на котором размещены наши файлы. Мы также говорили о доменном имени, поэтому есть URL-адрес или имя, соответствующее IP-адресу нашего сервера. Когда клиент вводит наше доменное имя в веб-браузере, компьютер должен знать IP-адрес. Итак, сервер доменных имен (DNS) — это система, которая сопоставляет IP-адреса с доменными именами. Это не то, что мы должны покупать, это то, что мы или администрация нашего веб-сайта должны настроить.

## Настройка

Теперь давайте посмотрим, что произойдет после того, как мы купим услугу хостинга. На этом этапе у вас должен быть IP-адрес вашего веб-сайта, имя пользователя и пароль. Вы должны перейти по ссылке, которая содержится в вашем электронном письме с подтверждением, и после входа в систему вам будет предложено pdesk или какой-либо другой интерфейс, на котором вы можете настроить свой хостинг и доменные имена. Я составлю список действий, которые нам нужно выполнить, а позже, в следующих разделах, мы обсудим решения для такого рода действий-вопросов. Итак, после покупки сайта вам необходимо:

- Перенести ваши файлы или сайт на сервер хостинга
- Настроить DNS в соответствии с вашим доменным именем

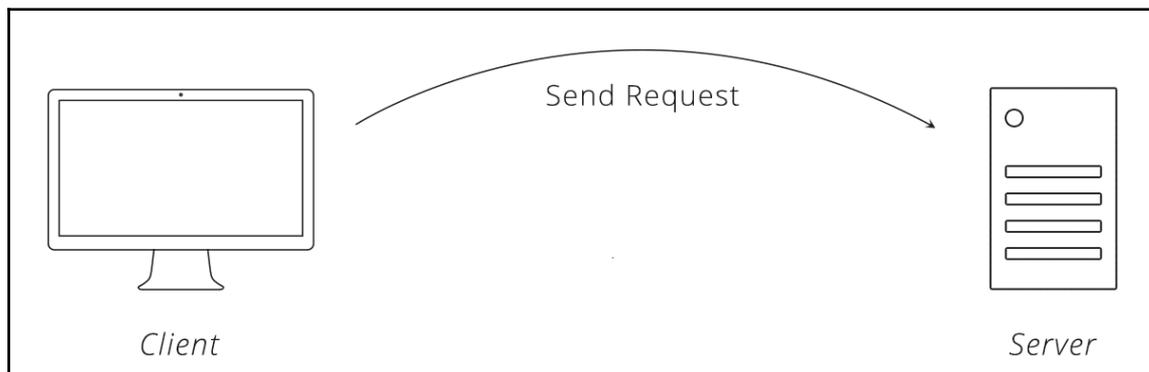
Если вы знакомы с веб-разработкой, у вас наверняка уже возникли некоторые вопросы. Здесь рождаются некоторые проблемы; например, как мы будем передавать файлы на хостинг? Обычно перетаскивание работает, но не всегда, особенно на нашей плате Raspberry Pi Zero W, но мы обсудим эту тему позже. Далее, как нам сопоставить доменное имя с IP-адресом сервера? Кроме того, что, если я захочу внести небольшое изменение? Нужно ли заново загружать все файлы? Как видите, веб-хостинг требует определенного опыта и не так прост для многих пользователей. Вот почему многие хостинг-сервисы имеют круглосуточную поддержку в режиме реального времени и страницу устранения неполадок.

## Взаимодействие клиент-сервер

Теперь, когда вы знаете, что такое веб-хостинг, у вас должны возникнуть вопросы. Поскольку вы всегда являетесь клиентом и у вас есть сервер дома или где-то в мире, как вы с ним общаетесь? Как любой клиент может связаться с сервером и получить его ответ? Идея взаимодействия клиент-сервер будет описана в этом разделе, а к концу раздела определение веб-хостинга и то, как он работает, должно быть предельно ясным.

### Запрос клиента

Как мы видели в предыдущем разделе, работа клиента состоит в том, чтобы просить. Он запрашивает файлы веб-сайта или что-либо еще с сервера. Подключение клиента к серверу показано на следующем рисунке, где клиент делает запрос через приложение:

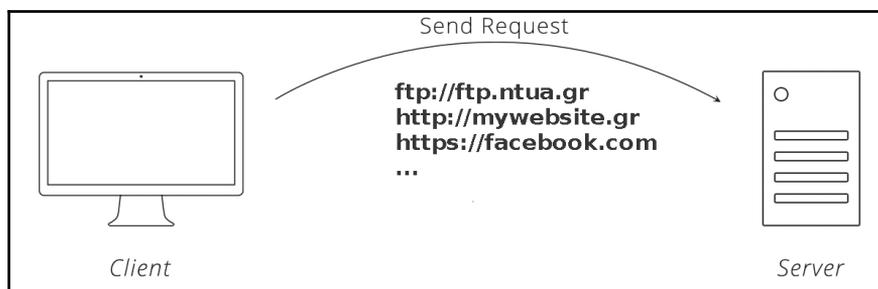


Обратите внимание, что клиент обычно является человеком, но, без сомнения, клиент может быть другим сервером. Связь клиент-сервер — это основная идея, при которой клиент запрашивает файлы протокола HTTP или HTTPS с сервера. Если на сервере нет этих файлов, то сам сервер может запросить их с другого сервера или ответить ошибкой 404: файл не найден.

## Протоколы связи

Существует множество протоколов, по которым может быть выполнен запрос. HTTP и FTP являются такими протоколами, и мы рассмотрим их в этом разделе:

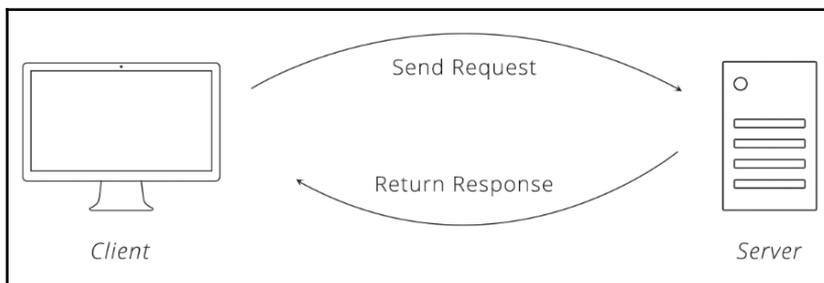
- HTTP (протокол передачи гипертекста): HTTP — это основной технологический протокол в Интернете, который позволяет создавать ссылки и просматривать. С другой стороны, HTTPS означает безопасный протокол передачи гипертекста. Некоторые веб-сайты используют HTTPS, например facebook.com, а другие, использующие HTTP, обычно имеют администратора, который понятия не имеет, что означает S в HTTPS, или по какой-то причине не обновил службу хостинга с помощью SSL. Если вы приобретете пакет хостинга, вам, вероятно, будет предложено купить и добавить SSL на свой сайт. SSL означает, что HTTP будет снабжен буквой S в конце, чтобы ваш сайт был более безопасным.
- FTP (File Transfer Protocol - протокол передачи файлов): обычно используется, когда мы хотим передавать файлы; например, есть много FTP-серверов, таких как `http://ftp.ntua.gr`, где они не обслуживают веб-сайт, а только файлы. Когда вы подключаетесь к серверу по определенному протоколу, вы должны использовать определенный номер порта. Коммуникационные протоколы работают на определенных портах. HTTP(S) использует порт 80, а FTP использует порт 21. Когда вы настраиваете сервер у себя дома, вы должны указать порт, на котором он будет работать, и если, например, ваш сервер работает на порту 2356, то вам нужно посетить `<http(s)/ftp>://<server-IP/domain-name>:<port>`. Сначала вы определяете используемый протокол, затем IP-адрес или домен и, наконец, порт. Подводя итог, ваш запрос должен следовать простому правилу `<Protocol, IP, Port>`. В результате у нас может быть много разных клиентских запросов, а именно:



В следующем разделе мы поговорим об обработке запросов на стороне сервера.

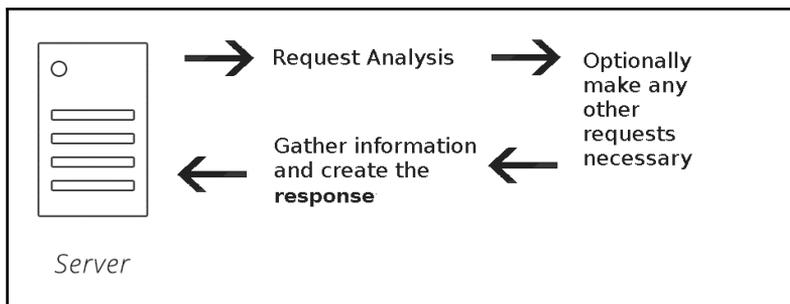
## Ответ сервера

В предыдущем разделе мы обсудили, как клиент может делать разные запросы. Все они куда-то попадают, и кто-то должен с ними справиться. Либо ответьте соответствующими файлами и ответом, либо каким-либо сообщением об ошибке. Далее вы можете увидеть, что как только сервер получил запрос от сервера, он ответил следующим образом:



Теперь мы поговорим об этом ответе и о том, что происходит до того, как ответ достигнет клиентской стороны. Допустим, кто-то разговаривает с вами и что-то спрашивает. Естественно, что вы всегда делаете, обрабатываете этот запрос, а затем отвечаете. Это именно то, что делает сервер.

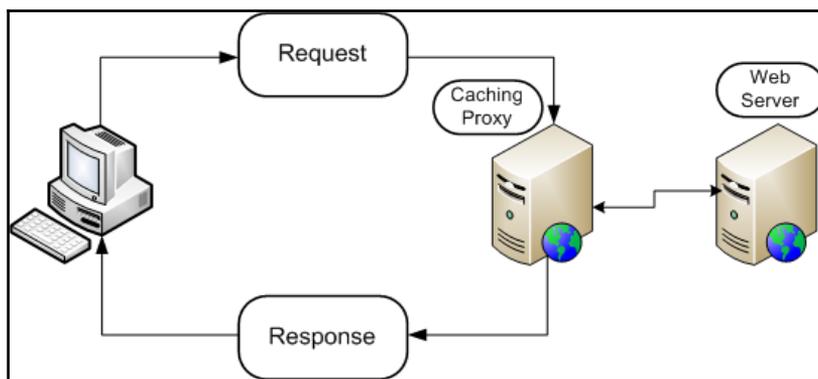
Сервер получает сотни запросов со всего мира и обрабатывает эти запросы соответствующими методами, чтобы выяснить, чего именно хочет каждый из них. Представьте, что запрос похож на волшебную коробку, внутри которой у нас есть протокол, имена всех запрашиваемых файлов, опционально, имя пользователя и пароль. Сервер открывает это окно и различает все эти компоненты. В зависимости от имени протокола он отправляет остальные компоненты на соответствующие обработчики, которые в зависимости от некоторых других параметров отправляют данные другим обработчикам и так далее. Там есть целая процедура, в рамках которой анализируется ваш запрос и готовится ваш ответ. На следующем рисунке вы можете увидеть простую демонстрацию того, что происходит на стороне сервера:



Как только сервер получает всю необходимую информацию, он отправляет ответ на IP-адрес, указанный в сообщении запроса.

## Прокси-серверы и кеширование

Другой важной частью являются прокси-серверы, где они фактически действуют как промежуточный сервер между клиентом и фактическим сервером. Обычно они используются в целях безопасности и анонимности. Как только клиент создает запрос, запрос достигает прокси-сервера, и прокси-сервер затем отвечает за определение того, что делать. Таким образом, клиент не видит фактический сервер и не может атаковать его напрямую или вызвать какие-либо другие плохие вещи. Кроме того, серверы обычно используют кэш-память. Кеширование — это то, что использует YouTube, крупные компании, и если вам нужны быстрые ответы, это крайне важно. Основная идея кеширования заключается в том, что запрашивающий сервер имеет файлы, которые вы запрашиваете, и может очень быстро ответить вам. Обычно они не такие большие и не могут вместить все. Представьте их как еще один сервер, стоящий перед настоящим большим сервером, на котором хранятся только 100 лучших запросов. Это похоже на щит, где вы сначала попадаете на кеш-сервер, а затем на реальный сервер. Если вы ищете или запрашиваете что-то, что уже запрошено многими пользователями, вы, вероятно, найдете это там и получите быстрый ответ. Если у прокси-сервера нет файла, он создает запрос от веб-сервера и отвечает, но в этом случае вам придется подождать немного дольше. На следующем рисунке мы видим, что клиент что-то запрашивает у сервера, но на самом деле он общается с прокси-сервером. Прокси-сервер находится перед основным сервером и действует как брандмауэр или другой уровень безопасности.



# Разработка сайта

Этот проект посвящен размещению вашего личного или любого веб-сайта на Raspberry Pi Zero W. В этом разделе у нас не будет аналитического обсуждения того, как вы можете развивать свой веб-сайт. Тем не менее, я упомяну различные способы, которыми вы можете работать, и дам общее указание, что вам следует и чего не следует делать. Некоторые веб-сайты имеют общий размер в МБ, а другие могут достигать ГБ в зависимости от размера необходимых файлов. Таким образом, разумный выбор при выборе типа веб-сайта, который вы будете размещать, очень важен, и если вы пропустите это, вы можете столкнуться с многими ошибками и проблемами.

Разработка веб-сайта на сто процентов зависит от ваших потребностей. Вы можете создать простой веб-сайт с кнопками для своего проекта IoT, чтобы иметь доступ к другим вашим устройствам Arduino или Raspberry Pi; или вы можете создать временный простой адаптивный веб-сайт, используя HTML, CSS и JavaScript; или вы даже можете настроить базу данных и создать интернет-магазин. Очевидно, что в этом разделе не будет рассказано, как создать продвинутый веб-сайт, но он проведет вас через некоторые основные этапы разработки простейшего веб-сайта.

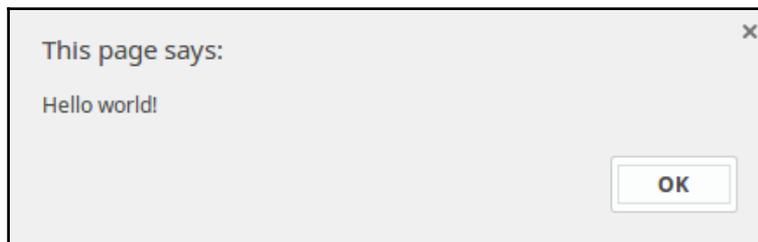
Предположим, сценарий здесь заключается в разработке веб-сайта, на котором вы нажимаете кнопку и открываете дверь. На веб-сайте должна быть кнопка и текст, указывающий на активность кнопки. На локальном компьютере откройте свой любимый редактор и создайте простой HTML-файл, в который вы вставили кнопку. Код выглядит следующим образом:

```
<!DOCTYPE html>
<html>
<body>
This button activate something:<button type="button" onclick="alert('Hello
world!')">Active</button>
</body>
</html>
```

Во-первых, мы определяем DOCTYPE как HTML. Далее внутри тега body нам нужно создать кнопку, которая при нажатии будет говорить [Hello world](#). Очевидно, вы должны изменить ее и связать с проектом Arduino, который у вас может быть. Предыдущий веб-сайт похож на следующее изображение:



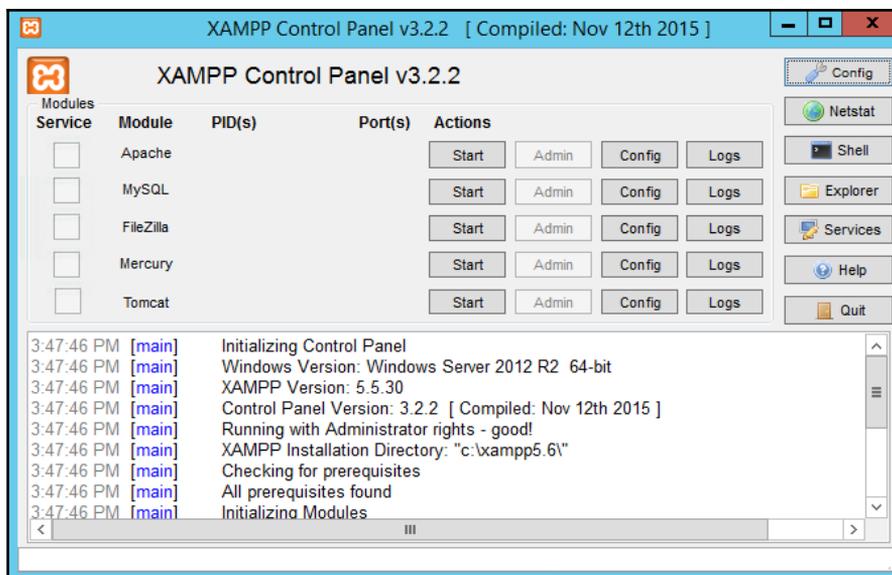
После нажатия на кнопку Active вы увидите всплывающее окно следующего вида:



## XAMPP/LAMPP

Пакет XAMPP — это бесплатный пакет для загрузки, который на самом деле представляет собой набор инструментов, необходимых для работы вашего веб-сайта. XAMPP имеет сервер Apache, MySQL и некоторые другие функции. Вы можете скачать XAMPP с <https://www.apachefriends.org/index.html>.

XAMPP поможет вам установить это программу и спросит, хотите ли вы активировать и запускать сервер Apache, который необходим для запуска и выполнения кода PHP и MySQL, если ваш веб-сайт взаимодействует с базами данных, такими как WordPress. Панель управления XAMPP выглядит следующим образом:



Через эту панель управления вы можете запускать сервер Apache, службу MySQL и многие другие службы. Для простоты мы запустим только сервер Apache и службу MySQL. Обратите внимание, что в случае каких-либо ошибок закройте приложения, работающие на порту 80 или 8080, такие как Skype или Steam. Если все пойдет хорошо, Apache и MySQL будут отмечены зеленым цветом.

XAMPP создает локальный сервер. В результате машина, на которой установлен XAMPP, является фактическим сервером. Вам необходимо установить XAMPP на плату Raspberry Pi Zero W. На официальном сайте вы обнаружите, что XAMPP доступен для многих операционных систем. XAMPP имеет определенное место, куда вам нужно поместить файлы, поэтому все это будет работать. В Windows он находится в `C:\xampp\htdocs\`, а в Linux - `/opt/lampp/htdocs`. Очевидно, что поскольку Raspberry Pi - это Linux, вам нужно будет создать папку внутри папки `htdocs` после установки.

## Доступ к веб-сайту

Для локального доступа к вашему веб-сайту вам нужен только IP-адрес платы Raspberry Pi. С этим IP-адресом любое устройство, компьютер или мобильный телефон в вашей локальной сети может посетить URL-адрес `<ipdevice>/<folder>`, где `<ipdevice>` — это IP-адрес вашей платы Raspberry Pi Zero W, а `<folder>` — это имя папки, которую вы только что создали в папке `htdocs`. Если вы сделаете все шаги, описанные ранее, вы ничего не увидите, потому что у нас нет веб-страницы. Теперь внутри папки нам нужно создать файл `index.html` или `index.php`, содержащий наш веб-сайт. Сервер Apache всегда будет сначала искать индекс имени. Если файла с именем `index` нет, будут перечислены все возможные файлы, которые у вас есть в этой папке. В следующем разделе вы узнаете, как перенести и загрузить веб-сайт в эту папку.

## Дистанционное управление

В зависимости от места, где вы размещаете свой веб-сайт или свои файлы, существуют различные способы доступа и изменения ваших файлов. Обычно веб-разработка — это не разовая работа, а это значит, что мы не загружаем файлы, выходим с сервера и дело сделано. Возможно, вам придется вернуться, внести изменения, загрузить новые файлы и многое другое. Всегда важно иметь простые и легкие способы выполнения таких задач, с графическим интерфейсом пользователя и без него. Существует два основных протокола, используемых для управления удаленным сервером и взаимодействия с ним. Первый — через SSH, как мы уже видели, а второй — через SFTP.

## SSH

SSH, что означает Secure Shell, — это старый, но все еще хороший способ подключения к вашему удаленному серверу, обычно через порт 22. SSH — это продвинутая telnet-связь, которую многие люди используют для подключения к своему серверу. Чтобы использовать SSH, вы должны установить сервер SSH на стороне сервера и использовать клиент SSH на стороне клиента. Поскольку серверный порт 22 будет запущен и будет работать в ожидании входящих подключений, вы можете легко открыть свой SSH-клиент и войти на сервер со своими аккаунтами. Далее мы упомянем некоторые способы сделать это с помощью Windows и Linux. После этого разберем скрипт, который делает это автоматически. Это сэкономит время и сделает соединение более безопасным.

## SSH-клиенты

В зависимости от вашей операционной системы вам необходимо загрузить и использовать соответствующий SSH-клиент. Все клиенты и программы SSH должны знать IP-адрес, к которому вы будете подключаться, и номер порта. Эта процедура будет описана шаг за шагом, а после этого мы сделаем ее автоматической. Итак, давайте посмотрим, какие у нас есть варианты для разных операционных систем.

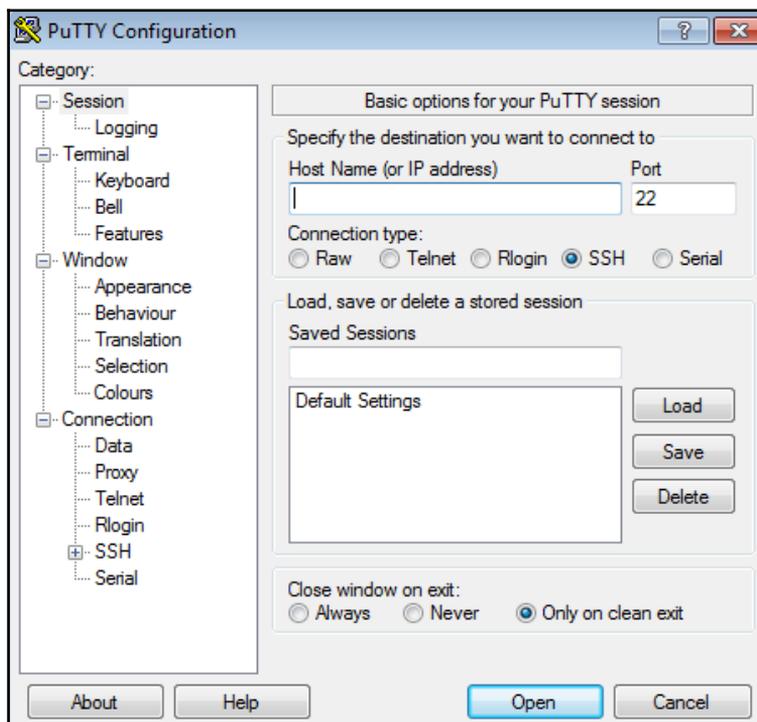
### Windows

Для Windows пользователи используют обычный ssh-клиент под названием PuTTY, который хорошо работает и является бесплатным. В Windows 10 и более поздних версиях, вероятно, будут командные инструменты, позволяющие открывать ssh-соединения. Однако, поскольку пользователи Windows знакомы с графическим интерфейсом, давайте остановимся на программе, упомянутой ранее. PuTTY можно скачать с официального сайта бесплатно:

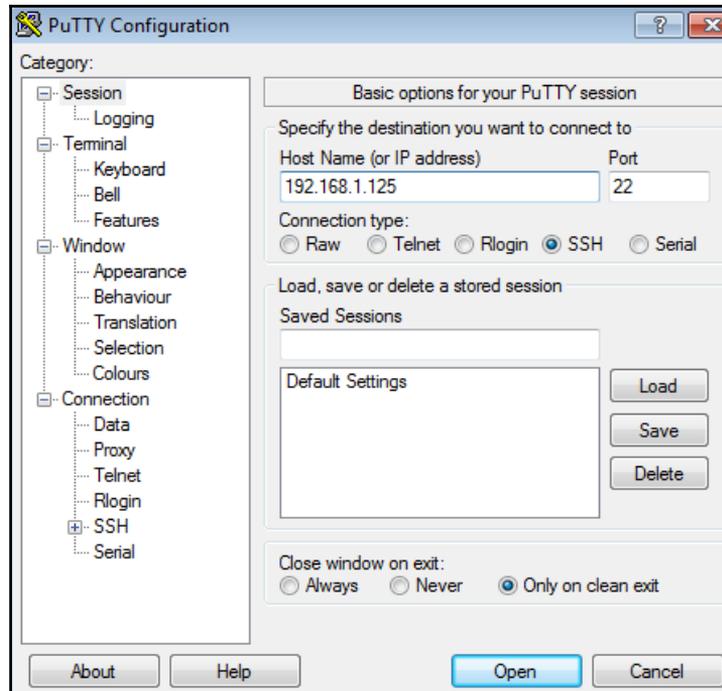
<https://www.chiark.greenend.org.uk/>

На их сайте есть много ссылок на разные версии, но мы можем остановиться наверху страницы, где легко увидеть, что в разделе Package Files есть ссылки на 32-битные и 64-битные системы Windows. Для пользователей Linux, даже если здесь есть возможность загрузить PuTTY, вы, вероятно, захотите прочитать больше и обнаружите, что у вас уже установлена программа, которая выполняет эту работу.

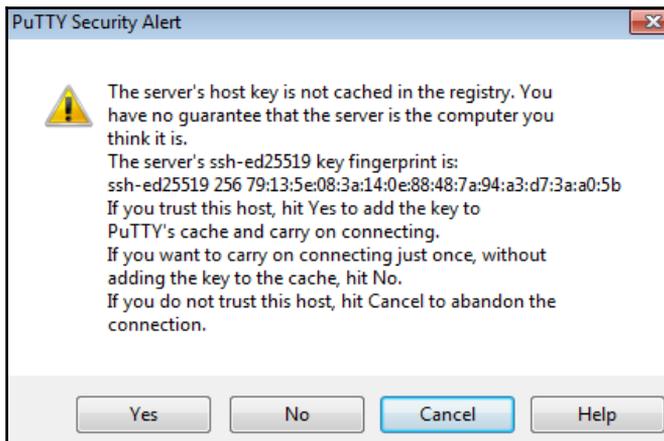
После загрузки PuTTY важно знать, что вам не нужно устанавливать эту программу, просто откройте его, и на экране должен появиться пользовательский интерфейс. Вы увидите следующий скриншот:



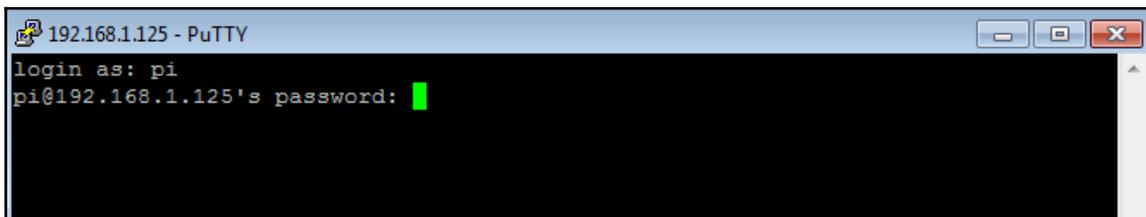
Следующим шагом является определение имени хоста IP-адреса вашего удаленного сервера. Поскольку в этом проекте мы будем подключать клиента к серверу Raspberry Pi Zero W, нам нужно найти IP-адрес Pi; например, пройдя через интерфейс вашего роутера, вы должны найти IP. После этого вы должны вставить его в поле «Host Name - Имя хоста» (или IP-адрес), как показано на следующем изображении. Мой Raspberry Pi имеет IP-адрес 192.168.1.125:



Нажав «Опен - Открыть», вы должны увидеть сообщение об отпечатке пальца. Каждый, кто создает SSH-соединение, оставляет отпечаток пальца на сервере Raspberry Pi Zero W, поэтому, поскольку вы новый клиент, вам просто нужно нажать «Да»; например, возможный отпечаток пальца должен быть следующим:



После этого вам будет предложено ввести имя пользователя и пароль удаленного сервера. После ввода имени пользователя, как вы можете видеть на следующем изображении, вам будет предложено установить поле пароля. Обратите внимание, что при вводе пароля в терминал могут не выводиться символы, поэтому вводите его медленно и правильно. На следующем снимке экрана мы ввели имя пользователя как pi и пароль по умолчанию как raspberry:



Имейте в виду, что после подключения к Raspberry Pi Zero W вы можете изменить свой пароль с помощью следующей команды:

```
sudo passwd
```

## Linux

Пользователям Linux не нужно загружать программное обеспечение; они могут открыть окно терминала и выполнить работу там. В следующем руководстве будет использоваться Linux Mint, но все дистрибутивы на основе Debian должны быть одинаковыми. Надеюсь, пользователям Mac не придется делать ничего другого, поскольку у них также есть терминал. Следующие сведения должны охватывать всех пользователей Linux.

Как упоминалось ранее, мы будем работать через терминал, поэтому откройте окно терминала или консоли и введите следующую команду:

```
ssh
```

Вы увидите что-то похожее на следующее, если вы установили ssh в своей системе (или он был установлен системой):

```
mark@zeus ~ $ ssh
usage: ssh [-1246AaCfGgKkMnQsTtVvXxYy] [-b bind address] [-c cipher_spec]
          [-D [bind address:]port] [-E log file] [-e escape_char]
          [-F configfile] [-I pkcs11] [-i identity_file] [-L address]
          [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
          [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
          [-w local_tun[:remote_tun]] [user@]hostname [command]
```

После того, как вы нашли IP-адрес Raspberry Pi Zero W, вы должны создать следующую команду и ввести ее на своем терминале, где пользователь — `pi`, а `ipaddr` — IPv4 - адрес вашей платы Raspberry Pi Zero W:

```
ssh user@ipaddr
```

Например, на следующем снимке экрана мы подключимся к 192.168.1.125:

```
mark@zeus ~ $ ssh pi@192.168.1.125
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jun 27 16:27:48 2017 from 192.168.1.117
pi@raspberrypi:~$ _
```

Имейте в виду, что на предыдущем скриншоте система не запрашивала пароль. Это связано с тем, что вход в систему обменивается некоторыми ключами в фоновом режиме, но мы поговорим об этом в следующем разделе. В вашем случае вам будет предложено ввести свой пароль. Так что это все.

## Автоматизация процедуры SSH

В предыдущем примере было необходимо вводить пароль каждый раз. Далее мы будем использовать `rsa-keygen` для автоматического входа на ваш сервер. Идея в том, что мы создадим уникальный ключ. Затем мы передаем его на сервер, и каждый раз, когда мы подключаемся к серверу по `ssh`, мы автоматически предоставляем свой ключ, который, очевидно, будет совпадать. Таким образом, мы можем подключаться без паролей, и это более безопасно. Итак, давайте шаг за шагом и посмотрим, как мы можем автоматизировать эту процедуру:

1. Откройте свой терминал на стороне клиента.
2. Первым шагом является создание пары ключей в клиенте. Вы можете сделать это, набрав в терминале следующую команду:

```
ssh-keygen -t rsa
```

- Далее вам нужно будет вставить имя файла. Как показано далее, вы можете ввести здесь все, что хотите, в качестве имени файла вашего ключа `rsa`:

```
mark@zeus ~ $ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mark/.ssh/id_rsa): myrsa_
```

3. Следующим шагом является создание пароля. Вы можете оставить его пустым и просто нажать `Enter`. После этого снова нажмите `Enter`. Вы должны увидеть что-то похожее на следующее, где создается `rsa keygen`:

```
mark@zeus ~ $ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mark/.ssh/id_rsa): myrsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in myrsa.
Your public key has been saved in myrsa.pub.
The key fingerprint is:
SHA256:HsYau9pX8GLXpvKM8GR06kw/tUK/nFwZREEpc+Uq9u4 mark@zeus
The key's randomart image is:
+----[RSA 2048]-----+
|
|   .+=|
|   0.+|
|   .+.|
|   ..+.|
|   .S+ +0+..|
|   *..@.+0. 0|
|  up to 0 .@ 0 0.0|
|   .. = *+00|
|   .o. .+E. |
+----[SHA256]-----+
mark@zeus ~ $
```

4. Не закрывая терминал и не меняя текущий каталог, вы можете найти созданные файлы, просто выполнив команду `ls`. В предыдущем примере, где мы определяем имя ключа `rsa` как `myrsa`, создаются два файла. Первый называется `myrsa`, а второй - `myrsa.pub`. На этом этапе очень важно понимать, что файл с именем `myrsa` (без расширения `pub`) является вашим ЧАСТНЫМ ключом и ТОЛЬКО для вас. Вы никогда не должны публиковать или передавать его кому-либо. Это как ваш секретный пароль.
5. Теперь вам нужно передать файл `myrsa.pub` на удаленный хост. Точнее, вам нужно поместить его в папку `~/.ssh/` удаленного сервера.
6. Последним шагом является создание (если он не существует) файла `~/.ssh/authorized_keys` и копирование содержимого файла `myrsa.pub` в файл `authorized_keys`.

Теперь вы должны быть готовы войти в систему без пароля.

Наконец, вы можете автоматизировать команду `ssh` с помощью следующего простого файла сценария. Замените IP-адрес скрипта на свой собственный IP-адрес Raspberry Pi:

```
#!/bin/bash
# This script opens an ssh connection with the Raspberry Pi
ssh pi@192.168.1.125
```

Сохраните этот скрипт под именем `spi` и поместите его в `/bin` на вашем локальном компьютере с Linux. Теперь способ, которым я люблю подключаться к моему Raspberry Pi Zero W, — это просто нажать `Ctrl + Alt + T`, чтобы открыть терминал, а затем ввести `spi` и нажать `Enter`. В течение трех секунд вы подключаетесь к Raspberry Pi.

## SFTP

SSH великолепен, и вы, возможно, уже использовали его для подключения к Raspberry Pi. Однако иногда нам нужно передавать файлы с платы Raspberry Pi и на нее. SSH на это не способен. Нам нужно использовать другой протокол, чтобы легко передавать наши файлы и не усложнять эту процедуру. Опять же, в зависимости от вашей операционной системы, вы можете использовать различные программы. Далее мы упомянем два способа сделать это; способ с использованием пользовательского интерфейса, который будет работать для операционных систем Windows и Linux, и еще один способ через Терминал.

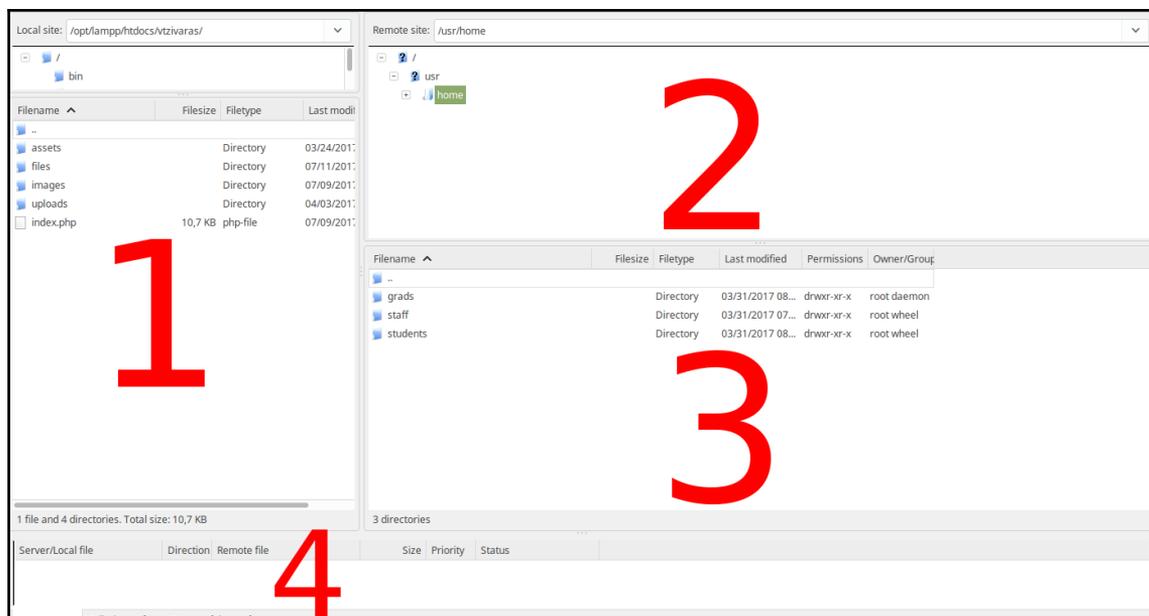
## FileZilla

FileZilla — это бесплатная программа, которую вы можете легко загрузить для операционных систем Windows и Linux. Вы можете скачать ее с <https://filezilla-project.org/download.php>.

После загрузки и установки FileZilla вам нужно определить всего несколько параметров:

- Host: здесь вам нужно указать имя хоста или IP-адрес вашей платы Raspberry Pi Zero W.
- Username: укажите имя пользователя. Обычно это pi
- Password: укажите здесь пароль. Пароль по умолчанию raspberry
- Port: номер порта 22.

Интерфейс FileZilla состоит из четырех основных разделов:



В первом разделе у вас есть файлы вашего локального компьютера. Во втором разделе у вас есть папки вашего удаленного сервера (Raspberry Pi), а в третьем разделе у вас есть файлы в выбранной папке из второго раздела. Наконец, в четвертом разделе у вас есть статус переданных файлов. Во время передачи вы увидите, что все файлы находятся в очереди и некоторые из них переносятся успешно, а другие нет. Очевидно, что при успешной передаче файлов нам нужно, чтобы все файлы были успешно переданы.

## Терминал

Хорошо известно, что все, что вы можете сделать с помощью программы с пользовательским интерфейсом, можно сделать с помощью терминала. Помимо команды `ssh`, есть еще одна, которая позволяет нам передавать файлы. Введите в терминале следующую команду:

```
sftp
```

Как и в руководстве по `ssh`, вы увидите что-то похожее на следующее; синтаксис почти такой же:

```
mark@zeus ~ $ sftp
usage: sftp [-1246aCfpqrv] [-B buffer_size] [-b batchfile] [-c cipher]
          [-D sftp_server_path] [-F ssh_config] [-i identity_file] [-l limit]
          [-o ssh_option] [-P port] [-R num_requests] [-S program]
          [-s subsystem | sftp_server] host
sftp [user@]host[:file ...]
sftp [user@]host[:dir[/]]
sftp -b batchfile [user@]host
```

Итак, чтобы подключиться к серверу с целью передачи файлов, нам нужно выполнить команду `sftp` следующим образом:

```
sftp user@IPaddr
```

Здесь пользователь — это наше имя пользователя на удаленном сервере. В Raspberry Pi это `pi`, а `IPaddr` — это IP-адрес Raspberry Pi. В нашем примере это `192.168.1.125`. Если мы выполним эту команду, мы увидим, что ей нужен наш пароль. Используя предыдущую процедуру, которую мы уже описали, мы можем автоматизировать необходимый пароль. Однако после ввода пароля мы увидим, что `bash` изменился. Теперь есть два способа взаимодействия с удаленным сервером. Мы можем ставить и получать файлы с сервера. Вы можете легко перенести файлы с удаленного сервера на локальный компьютер с помощью команды `get`:

```
get file1.txt
```

Кроме того, вы можете загружать файлы с вашего локального компьютера на удаленный сервер с помощью команды `put`:

```
put mylocalfile.txt
```

Обратите внимание, что в зависимости от каталога локального компьютера, из которого мы запустили команду `sftp`, мы можем соответствующим образом помещать и получать файлы. Это означает, что если бы мы находились в папке `/home/nick/`, мы можем напрямую загружать только файлы, которые находятся внутри папки `nick` с помощью команды `put`.



Для получения дополнительной информации о `ssh` и `sftp` прочитайте страницу руководства с:

- `man ssh`
- `man sftp`

## Сеть

Настройка вашего сайта — это одно. Настройка всех скриптов, правил и сетевых параметров, которые позволят вам получить доступ к веб-сайту по всему миру, — это еще одна и очень важная вещь. В этом разделе мы обсудим, как вы можете получить доступ к своему роутеру, чтобы определить и настроить некоторые правила, чтобы ваш веб-сайт был доступен для всех. Кроме того, мы обсудим кое-что о вашем динамическом домашнем IP-адресе, определим проблему, с которой вы можете столкнуться, и способы ее решения. Наконец, у нас будет введение в уровни безопасности, которые будут обсуждаться более подробно в следующей главе.

К настоящему времени у вас должен быть доступ к вашему сайту локально. Подводя итог всем предыдущим главам, вот пошаговое руководство, показывающее, что вы должны были сделать. Важно оглянуться назад, прежде чем читать раздел «Сеть», где мы выполним некоторые действия, относящиеся к роутеру и локальной сети.

1. Загрузите или создайте файлы своего веб-сайта.
2. Используйте FileZilla для переноса файлов веб-сайта на Raspberry Pi.
3. Установите пакет XAMPP на Raspberry Pi.
4. Создайте новую папку в `/opt/lampp/htdocs/` с именем `mywebsite`.
5. Переместите все веб-сайты в `/opt/lampp/htdocs/mywebsite`.
6. Убедитесь со своего мобильного телефона или любого другого компьютера в вашей локальной сети, что URL-адрес `<raspberrypi-IP>/mywebsite` действителен и отображает ваш веб-сайт.

## Доступ к роутеру

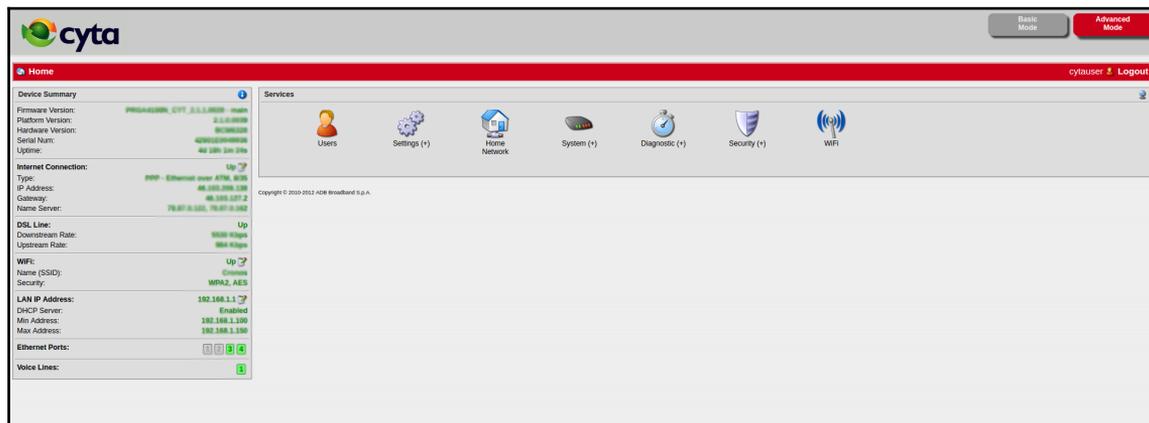
Теперь, чтобы получить доступ к вашему роутеру, вам нужно найти его IP-адрес. Обычно это 192.168.1.1, но в зависимости от вашего режима он может быть другим. Если у вас возникли проблемы, вы можете проверить руководство вашего роутера и узнать, что именно вам нужно сделать. Кроме того, вы можете позвонить своему интернет-провайдеру и спросить его. Предполагая, что вы нашли IP-адрес вашего роутера, теперь вам нужно открыть браузер и ввести следующий URL-адрес:

http://

Кроме того, вы можете ввести IP-адрес вашего роутера, если он отличается. Обратите внимание, что иногда нам нужно указать номер порта. Для этого нам нужно ввести номер порта помимо IP-адреса, чтобы сделать запрос с 192.168.1.1 на порт 8080, как показано ниже:

http://

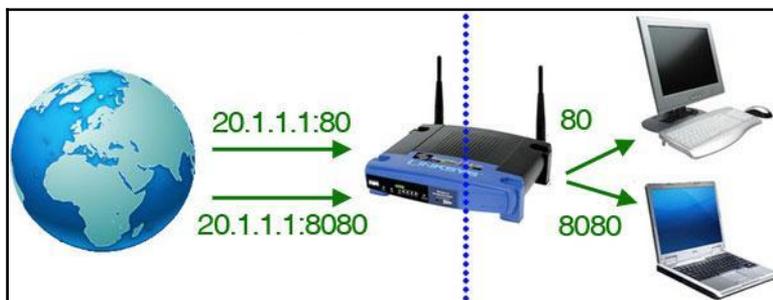
Вы должны увидеть что-то похожее на следующий скриншоте:



Это интерфейс вашего роутера и здесь вы можете многое изменить.

## Перенаправление порта

Во-первых, давайте поймем, почему переадресация портов важна, а затем мы рассмотрим способ ее настройки. Без переадресации портов наш роутер никак не мог бы определить хост-цель входящего соединения. Другими словами, когда вы создаете соединение с домашним роутером со своего мобильного телефона, вы не знаете IP-адрес своего Raspberry Pi. Теперь давайте посмотрим на общую идею переадресации портов. На следующем снимке экрана видно, что у нас есть роутер с IP 20.1.1.1 и два устройства: стационарный компьютер и ноутбук:



В результате нам нужно создать некоторые правила, которые помогут роутеру определить и решить, что делать. Здесь вы можете увидеть формулу, с помощью которой мы создаем правило. Обозначим следующую команду как INC:

```
{<incoming connection IP>, <incoming connection port>}
```

Теперь давайте сделаем то же самое с Raspberry Pi. Raspberry Pi теперь является веб-сервером, что означает, что он обрабатывает запросы через порт 80, как указывает протокол HTTP. Итак, мы обозначаем RPI следующим образом:

```
{<raspberry pi IP>, <80>}
```

На данный момент единственное, что нам нужно сделать, это создать правило следующим образом:

```
INC -> RPI
```

При этом мы на самом деле имеем в виду, что мы направляем каждое входящее соединение с любого IP-адреса в порту <incoming connection port> на <raspberry pi IP> в порту 80. Подводя итог, ваши правила переадресации портов должны выглядеть следующим образом:

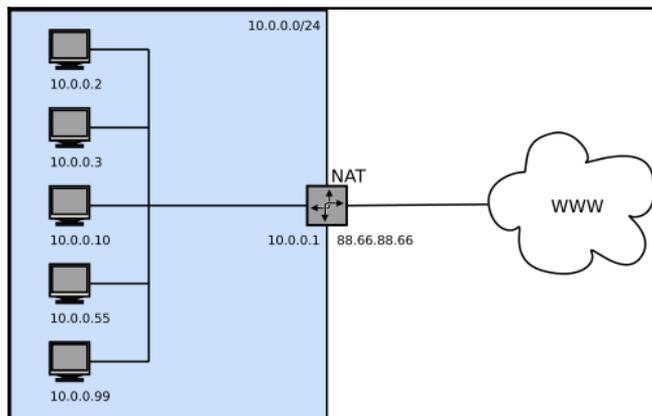
Traffic Coming from	External Port		Internal Port
Any Host PPP - Ethernet over ATM, 8/35	TCP 2880	➡	6869
Any Host PPP - Ethernet over ATM, 8/35	TCP 22	➡	Same Port
Any Host PPP - Ethernet over ATM, 8/35	TCP 4545	➡	80

В зависимости от руководства вашего роутера существует множество различных способов доступа к настройкам переадресации портов или настройкам сопоставления портов. Чтобы узнать это, прочтите руководство к роутеру. На предыдущем изображении для любого входящего соединения через порт 2880 перейдите к порту 6869 в определенном IP-адресе, который я не показывал из соображений безопасности.

## Домашний IP-адрес

Прежде чем говорить что-либо об IP-адресах, важно знать, как работает Интернет, как сети связаны друг с другом и что именно происходит. Предполагая, что в каждом доме есть роутер, каждый дом имеет IP-адрес, который не является статическим, поэтому он меняется без предупреждения. В вашей локальной сети все ваши IP-адреса имеют одну и ту же сетевую часть и другую хост-часть. Это означает, что если ваша локальная сеть — 192.168.1.X, три блока одинаковы для всех ваших устройств, включая ваш роутер, а последний, помеченный как X, отличается в зависимости от устройства. Обычно роутер имеет X = 1, ваше первое подключенное устройство имеет X = 2 и так далее. Однако это не является правилом и не гарантируется. Ваш дом (или роутер) может иметь IP-адрес, например 46.34.6.23, который необходимо использовать для получения сообщений и данных с интернет-серверов. Теперь каждый раз, когда вы отправляете пакет со своего компьютера на IP-адрес 192.168.1.125 в Интернете, ваш роутер отвечает за изменение IP-адреса 192.168.1.125 на домашний IP-адрес 46.34.6.23.

Когда он возвращается, он снова меняет 46.34.6.23 на IP 192.168.1.125. Эта процедура называется NAT, что означает преобразование сетевых адресов. На следующем изображении показан пример NAT:



На предыдущем изображении все IP-адреса, начинающиеся с 10.0.0.X, являются хостами, а маршрутизаторы имеют локальный IP-адрес 10.0.0.1. и внешний IP 88.66.88.66. Проблема здесь в том, что ваш домашний IP-адрес меняется, потому что ваш контракт с вашим интернет-провайдером имеет динамический IP-адрес.

Либо вам нужно изменить его на статический IP-адрес, либо вам нужно найти способ узнать свой домашний IP-адрес. Конечно, поскольку статические IP-адреса не безграничны, вам придется заплатить дополнительные деньги, поэтому, если ваш бюджет достаточно велик, выберите это решение. В результате вам нужно найти свой домашний IP-адрес только один раз, и он не изменится. Для всех остальных, которые не хотят платить, вы мало что можете сделать. Решение, которое я нашел, состоит в том, чтобы просто написать код Python, который создает соединение с сервером. Этот сервер может быть бесплатным или платным (если он у вас есть). На этом этапе ваш сервер знает ваш домашний IP-адрес. Итак, что вам нужно сделать, это просто запросить ваш домашний адрес с вашего сервера.

## Уровни безопасности

Чтобы создать введение к следующей главе, мы упомянем некоторые вещи о безопасности нашей системы; вещи, которые вы должны делать, и вещи, которые вы не должны делать. Вы должны помнить о настройке маршрутизатора для переадресации портов. Это не совсем дыра в безопасности, но вы точно не должны отключать какие-либо брандмауэры. Каждый маршрутизатор имеет брандмауэр для защиты вас и всей вашей локальной сети от неавторизованных пользователей. У вашего интернет-провайдера тоже есть брандмауэр, и, очевидно, на вашем компьютере есть брандмауэр. Вы можете отключить их все и позволить вашему компьютеру бесплатно пользоваться Интернетом, однако это слишком рискованно. Итак, первый уровень безопасности, о котором вы должны помнить, — это брандмауэры и целесообразно ли их отключать.

Важно знать, что только firewall - брандмауэр будет знать, будет ли входящее соединение вредоносным или нет. Firewall проверяет каждое входящее соединение и, в зависимости от вашей конфигурации, разрешает или блокирует соединения. Иногда, особенно когда мы играем в игры, это раздражает, потому что по какой-то причине соединение, относящееся к игре, блокируется, и нас выкидывает. Люди просто отключают почти все.

Следующий уровень безопасности, о котором мы поговорим, подпадает под вопрос: от кого вы хотите защититься? Мы все знаем, что мир небезопасен, поэтому весь интернет довольно опасен. Никто не может сказать: «Хорошо, теперь я на 100% в безопасности». Такого нет. Итак, вопрос в том, от кого вы хотите обезопаситься. Здесь есть несколько слоев. Во-первых, вы должны быть защищенным от своей семьи. Это требует, чтобы вы просто создавали пароли, и часто этого достаточно. Следующий уровень — быть в безопасности от друзей. Очевидно, речь идет о создании паролей, но вы также должны держать некоторые вещи при себе; например, вы не хотите публиковать свой домашний IP-адрес или порт, над которым вы работаете. Первый шаг атаки называется сбором информации, поэтому чем меньше злоумышленник знает о настройке вашей сети, тем лучше.

## Безопасность

В любой системе последним вопросом является безопасность. Очевидно, что сделать ваш веб-сайт и ваше соединение безопасным — непростая задача. Это требует знаний, и в этом разделе мы обсудим основную концепцию безопасности. Мы также упомянем некоторые действия, которые вы должны предпринять, чтобы предотвратить доступ неавторизованных пользователей к вашим файлам.

## Интерфейс роутера

Прежде всего, начнем с настроек в интерфейсе вашего роутера. Эти настройки могут различаться в зависимости от модели вашего роутера, но в конечном итоге у вас будет более или менее то же самое, что и у меня. Обычно все интернет-провайдеры предоставляют недорогие роутеры в базовой конфигурации. На следующем изображении вы можете видеть, что у нас есть некоторые параметры на вкладке безопасности роутера:

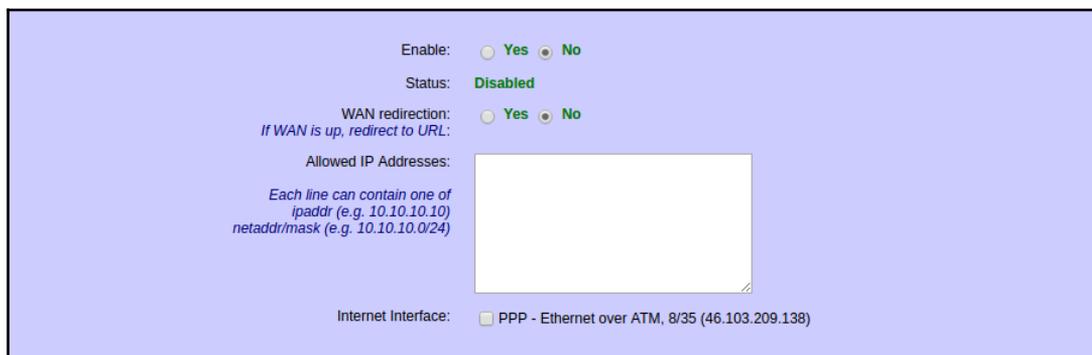


Далее, давайте посмотрим, что делает каждый из этих параметров:

- Firewall. Первый вариант — это брандмауэр. Здесь мы можем активировать или деактивировать и установить уровень безопасности. Если по какой-то причине ваше приложение не работает, рекомендуется включить низкий уровень безопасности, а не отключать его полностью.
- DMZ: означает демилитаризованную зону. DMZ — это физическая или логическая подсеть, которая отделяет внутреннюю локальную сеть (LAN) от других недоверенных сетей, таких как Интернет. В этом параметре безопасности необходимо указать внешний интерфейс, например, PPP — Ethernet over ATM, 8/35 (IP-адрес) и внутренний IP-адрес.
- Parental Control: этот параметр используется в основном для домашней безопасности, для ваших детей, ваших родителей и в подобных ситуациях. Основная идея Parental Control - родительского контроля заключается в том, что родитель определяет, какие веб-сайты могут посещать все остальные пользователи, а какие помечаются в черном списке. На следующем изображении показано, какие именно параметры у нас есть в этом параметре безопасности:



1. Captive Portal: Captive Portal — это веб-страница, которая отображается для новых пользователей перед получением доступа к Интернету. Обычно он используется для отображения опции оплаты, регистрации в баре или подобных вещей. На следующем изображении мы видим некоторые настройки, относящиеся к авторизованному portalу:



## Компьютер

Что касается компьютера, есть две или три настройки, которые мы должны выполнить. Прежде всего, поскольку мы говорим о плате Raspberry Pi Zero W, пароль по умолчанию — `raspberrypi`, поэтому, очевидно, вам нужно изменить его на свой пароль, используя буквы, цифры и символы.

Мы можем сделать это с помощью команды `passwd`. Используя терминал, введите:

```
passwd
```

Вам будет предложено ввести текущий пароль. После этого необходимо дважды ввести новый пароль. Теперь, если вы выйдете и снова войдете в систему, ваш пароль должен был измениться. Обратите внимание, что изменение вашего пароля требует дальнейших действий, если в других файлах сценариев был выполнен какой-либо код или автоматизация.

## ISP - Интернет-провайдер

Что касается Internet Service Provider (ISP) - интернет-провайдера, есть некоторые параметры, которые вы можете изменить. Вы должны позвонить своему интернет-провайдеру и запросить все настройки, относящиеся к безопасности, которые вы можете изменить. Не стесняйтесь спрашивать и изменять все настройки по своему усмотрению.

## Резюме

В этой главе вы узнали о хостинге и о том, как настроить плату Raspberry Pi ZeroW для размещения простого веб-сайта. Мы рассмотрели некоторые проблемы безопасности, которые могут возникнуть при размещении веб-сайта, и рассмотрели различные способы подключения к вашей удаленной серверной плате Pi. В следующей главе мы рассмотрим все шаги по настройке домашнего помощника. В наши дни довольно популярны помощники по дому, которые позволяют вам заказывать вещи на Amazon, слушать собственную музыку и управлять своим домом.

# 9

## AlexaPi

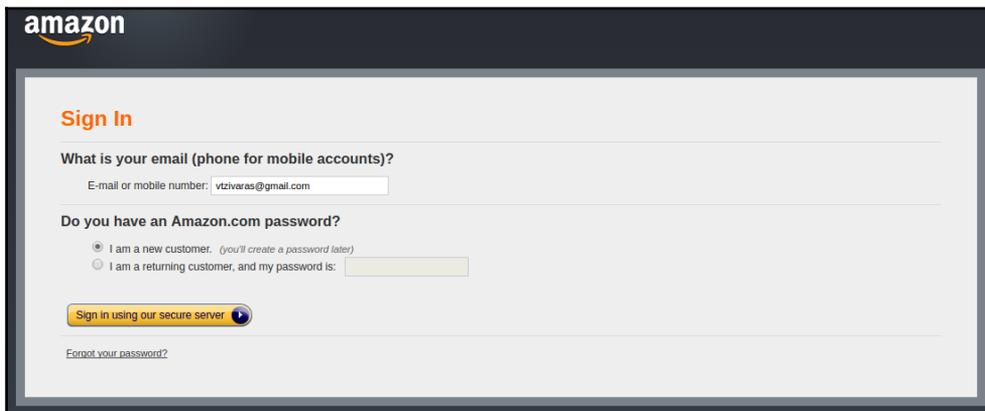
Ранее мы ознакомились с подробным руководством по проекту, как установить и настроить нашу плату Raspberry Pi Zero W для размещения нашего личного веб-сайта. Мы также говорили о сети и рассмотрели все шаги, которые мы должны предпринять, чтобы предоставить доступ к нашему веб-сайту любому человеку в мире. В этой главе мы рассмотрим другой проект, используя искусственный интеллект ИИ и ресурсы, доступные в Интернете, для настройки AlexaPi. AlexaPi — это проект-клон сервиса Alexa от Amazon. Поскольку Alexa довольно дорогая, вы можете иметь клон на своей плате Raspberry Pi Zero W за минимальную сумму денег. В конце главы мы упомянем некоторые преимущества и недостатки этого проекта. В этой главе мы поговорим о следующих темах:

- Создание аккаунта разработчика Amazon
- Настройка Raspberry Pi
- Установка Alexa
- Распознавание голоса
- Сетевое администрирование

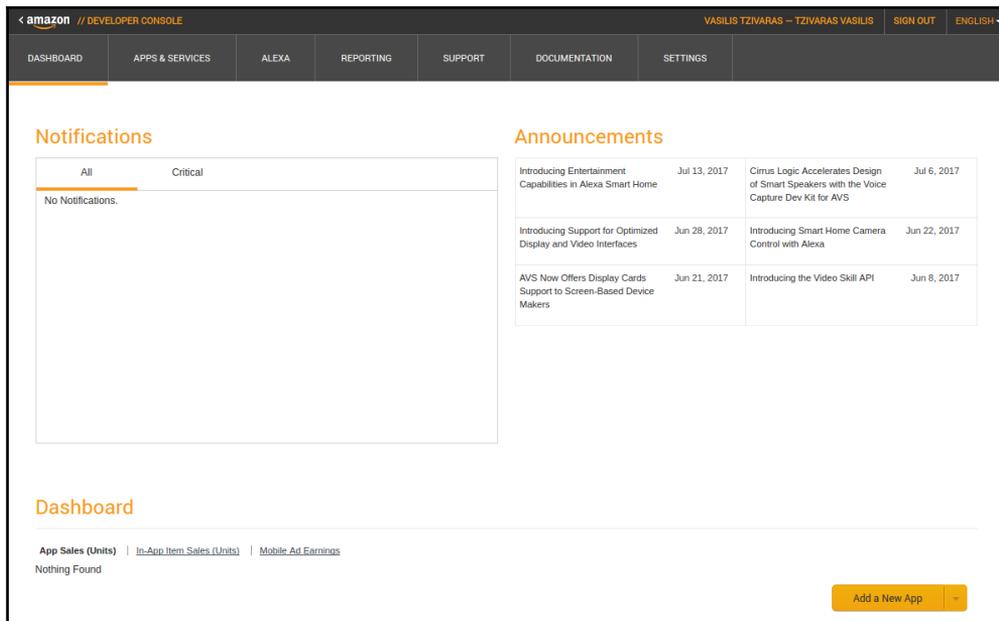
### **Создание аккаунта разработчика Amazon**

Перед загрузкой необходимой программы мы должны создать новый аккаунт разработчика Amazon, чтобы мы могли получить доступ к службам разработчиков Amazon. Затем мы будем бесплатно использовать Amazon Alexa Voice Services (AVS) на нашей плате Raspberry Pi Zero W. Мы должны загрузить его <https://github.com/>, а затем установить на нашу плату Raspberry Pi Zero W. После этого мы протестируем его, чтобы увидеть, как он работает.

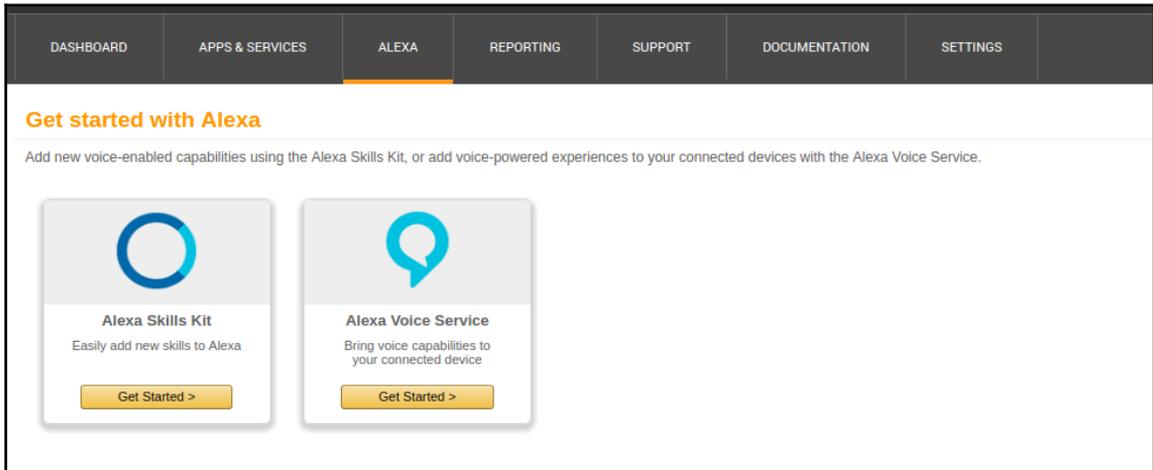
Чтобы создать свой личный аккаунт, если у вас его нет, просто посетите <https://developer.amazon.com/> и нажмите кнопку «Sign - Войти» в правом верхнем углу, и вы увидите следующий экран:



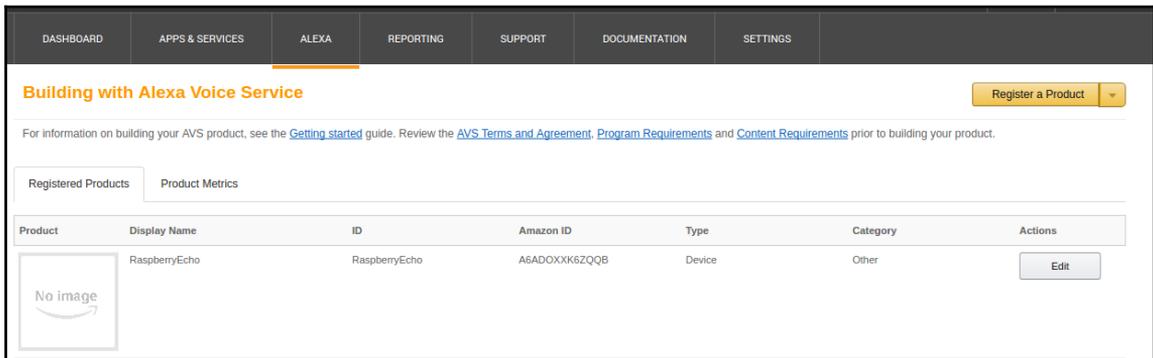
Здесь вы можете войти или зарегистрироваться, если у вас нет аккаунта. После этого подтвердите адрес электронной почты от Amazon и войдите на страницу разработчика. Вы должны открыть страницу панели инструментов разработчика, которая будет выглядеть примерно так:



Следующим шагом будет переход на вкладку ALEXA, где вы увидите две доступные опции. Первый — это Alexa Skills Kit, а второй — Alexa Voice Service. Как обсуждалось ранее, здесь мы будем использовать второй вариант, который называется Alexa Voice Service или AVS. Нажимаем на кнопку «Get Started - Начать», и давайте создадим и настроим все необходимые действия:



Alexa Voice Service откроет список устройств. В вашем случае, если вы ранее не выполняли эту процедуру, скорее всего, у вас там будет ноль устройств. Однако, как вы можете видеть далее, уже есть одно устройство, которое мы проигнорируем для простоты:



Теперь нам нужно щелкнуть стрелку на кнопке «Register a Product - Зарегистрировать продукт» и выбрать «Device - Устройство».

Наш Raspberry Pi Zero W будет устройством, которое мы регистрируем в AVS. После этого вы должны увидеть следующее изображение, где вам предлагается заполнить необходимые поля:

< Back to the list

**Create a new Device Type**

Getting started  
AVS Agreement  
AVS Program Requirements  
AVS Content Requirements

\* Fields required

**Device Type Info** ✓

**Security Profile** ✓

**Device Details** ✓

**Device Capabilities** ✓

**Company Name**  
The name of the company you listed in your developer account profile. Tzivaras Vasilis

**Device Type ID \***  
Choose a unique name that identifies your device. This name will not be shown to end users. It may only contain letters, numbers and underscores with no spaces.

**Display Name \***  
Choose a device name that you want shown to your end users when they register their devices with Amazon.

Next

В ID типа устройства вы должны указать простой ID для вашего продукта. Вы можете называть это как хотите. Кроме того, в поле «Display Name - Отображаемое имя» необходимо указать отображаемое имя портала разработчика. Обычно идентификатор типа устройства — это имя без пробелов (можно использовать \_), а отображаемое имя — это поле, в котором вы указываете имя, чтобы вам было легче понять и определить, что это такое. Когда вы будете готовы, нажмите «Next -Далее», и вы увидите следующий раздел, касающийся профиля безопасности, как показано ниже:

**Device Type Info** ✓

**Security Profile** ✓

**Device Details** ✓

**Device Capabilities** ✓

You need a security profile to identify your device. Your security profile credentials - client ID and client secret - allow your device to securely identify itself to the Alexa Voice Service. If you are building a website, click here to [Learn More](#). If you are building an Android or iOS app, click here to [Learn More](#).

**Security Profile** ⓘ \*

A security profile is how Amazon identifies your device.

General Web Settings Android/Kindle Settings iOS Settings

**Security Profile Description**  
Choose a description for your security profile for Amazon services to use in communicating with you.

**Security Profile ID**  
This ID will identify your security profile in Amazon services.

**Client ID** ⓘ  
This is a value specific to you that is assigned to you when you register with Login with Amazon.

**Client Secret** ⓘ  
This is a secret specific to you that is assigned to you when you register with Login with Amazon. Confidential.

В разделе «Security Profile - Профиль безопасности» в раскрывающемся меню «Select Security Profile - Выбрать профиль безопасности» необходимо выбрать профиль регистрируемого устройства. У него должен быть только один вариант. Если нет, создайте новый профиль. Прежде чем вы нажмете кнопку «Next - Далее», в горизонтальном меню должны появиться дополнительные настройки. На вкладке «General - Общие», вы должны указать имя профиля безопасности и описание профиля безопасности:

The screenshot shows the 'Create Security Profile' page in the AWS IAM console. At the top right, there are links for 'AWS Program Requirements' and 'AWS Content Requirements'. The main text explains that a security profile is needed to identify the device. Below this, there is a 'Security Profile' section with a 'Create a new profile' button. The 'General' tab is active, showing two input fields: 'Security Profile Name' and 'Security Profile Description'. A 'Next' button is located at the bottom right of the form.

Например, вы можете установить имя Security Profile Name - профиля безопасности в качестве Sample app security profile - профиля безопасности образца приложения и описание Security Profile Description - профиля безопасности в качестве примера Alexa Voice. После того, как вы установили эти два параметра, нажмите «Next - Далее», и вы увидите, что для вас будут сгенерированы следующие идентификаторы:

The screenshot shows the 'Generated Identifiers' section of the 'Create Security Profile' page. The 'General' tab is selected. It displays the following information:

<b>Security Profile Description</b> Choose a description for your security profile for Amazon services to use in communicating with you.	alexa example 2
<b>Security Profile ID</b> This ID will identify your security profile in Amazon services.	amzn1.application.752b14170ee940f58addb134086f169c
<b>Client ID</b> This is a value specific to you that is assigned to you when you register with Login with Amazon.	amzn1.application-0a2-client.7f54149c501e4ac48bb98b5484d8d028
<b>Client Secret</b> This is a secret specific to you that is assigned to you when you register with Login with Amazon. Confidential.	b131fa6e5e8b4cc7919bce63d01e078512b3e38f0776bc19a46f00fe6b3f5a7c

Теперь перейдите на вкладку «Web Settings - Веб-настройки». Убедитесь, что профиль безопасности выбран в раскрывающемся меню. При нажатии кнопки «Edit - Изменить» справа от раскрывающегося меню, вы сможете добавить другие Allowed Origins and Allowed Return URLs - разрешенные исходные и разрешенные URL-адреса возврата:

Device Type Info

Security Profile

Device Details

Device Capabilities

You need a security profile to identify your device. Your security profile credentials - client ID and client secret - allow your device to securely identify itself to the Alexa Voice Service. If you are building a website, click here to [Learn More](#). If you are building an Android or iOS app, click here to [Learn More](#).

Security Profile

General Web Settings Android/Kindle Settings iOS Settings

Allowed Origins   
Your website origin, when using Login with Amazon.

Allowed Return URLs   
If you make HTTPS calls to Login with Amazon with redirect\_uris, specify them here.

В разделе «Allowed Origins - Разрешенные источники» нажмите «Add Another - Добавить другое» и введите `https://localhost:3000`. Кроме того, в разделе «Allowed Return URLs - Разрешенные URL-адреса возврата» нажмите ссылку «Add Another - Добавить еще один», затем введите `https://ЛОКАЛЬНЫЙ :3000/authresponse`. Теперь мы должны нажать кнопку «Next - Далее» и перейти к следующему разделу. На следующем изображении вы можете видеть, что в разделе «Device Details - Сведения об устройстве» мы можем определить, как устройство будет выглядеть на экране инициализации, о котором упоминалось в начале руководства:

< Back to the list

Create a new Device Type

Getting started  
AVS Agreement  
AVS Program Requirements  
AVS Content Requirements

\* Fields required

Device Type Info

Security Profile

Device Details

Device Capabilities

Image   
Upload an image sized 142(width)x130(height) pixels in either PNG or JPG format. This image is displayed on your customer's [Manage Your Content and Devices](#) page.

Category \*

Description \*

Do you have plans to make your product available to the general public? \*  Yes  No

Is your device a children's product or is it otherwise directed to children under the age of 13? \*  Yes  No  
Not sure? [Learn more](#)

Здесь вы должны указать изображение для вашего устройства. В Google выберите изображение Raspberry Pi для размещения. Этого должно быть достаточно. После этого в разделе «Category - Категория» выберите «Other - Другое» и в описании укажите простое описание вашего проекта, например, образец теста службы Alexa. Затем нажмите «Next - Далее», и вы увидите, что мы почти готовы. Должна появиться следующая страница, где вам будет предложено просмотреть настройки и нажать кнопку «Submit - Отправить»:

< Back to the list

**Create a new Device Type** [Getting started](#)  
[AVS Agreement](#)  
[AVS Program Requirements](#)  
[AVS Content Requirements](#)

\* Fields required

**Device Type Info**  Select all capabilities that your product supports. AVS will provide access to specialized APIs for each supported capability.

**Security Profile**

**Device Details**

**Device Capabilities**  **Display Cards**   
Display Cards for Alexa allow products to render "Now Playing" information for music, as well as static display cards for general knowledge questions, weather, to-do and shopping lists, calendar updates, and Alexa skills. Your product should have a screen to enable this capability.

By submitting this form, you agree to [Alexa Voice Service Agreement](#).

**Submit**

Как вы можете видеть на следующем изображении, у меня есть два устройства, так как у меня уже было одно, но вы также увидите это:

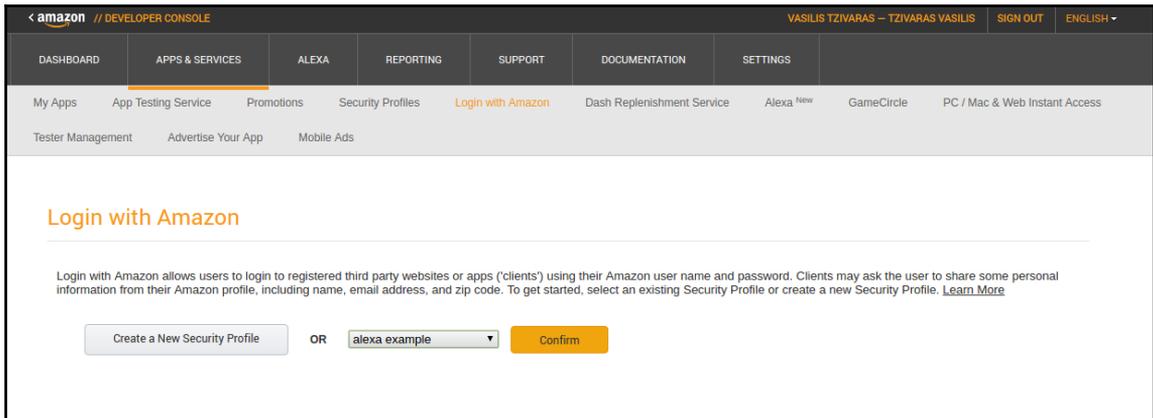
**Building with Alexa Voice Service** [Register a Product](#)

For information on building your AVS product, see the [Getting started](#) guide. Review the [AVS Terms and Agreement](#), [Program Requirements](#) and [Content Requirements](#) prior to building your product.

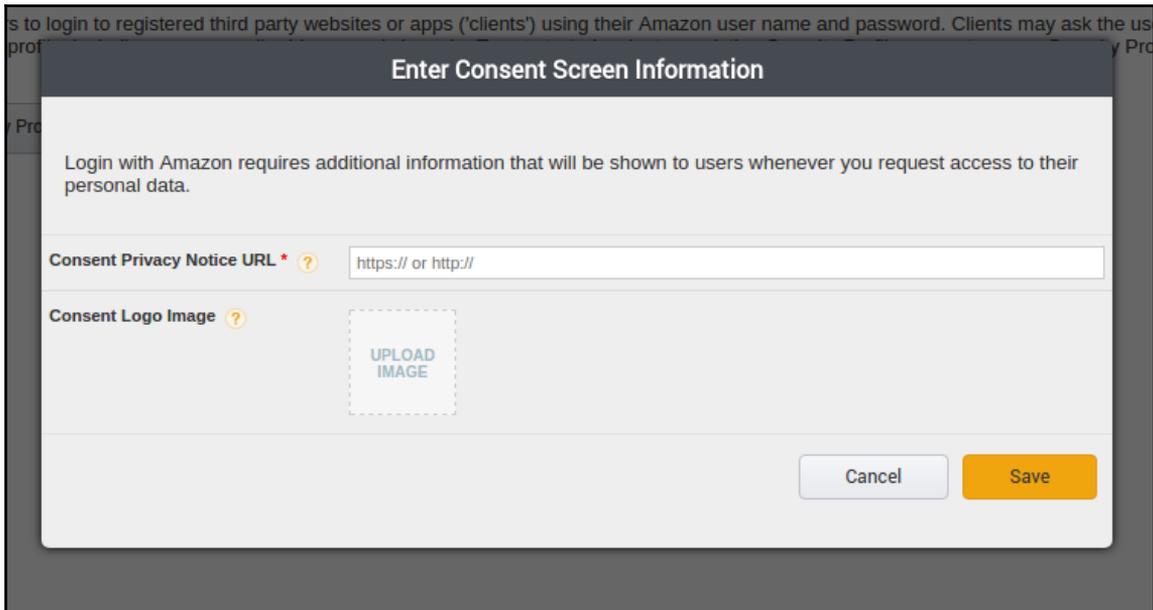
Registered Products | Product Metrics

Product	Display Name	ID	Amazon ID	Type	Category	Actions
	RaspberryEcho	RaspberryEcho	A6AD0XXK6ZQQB	Device	Other	<a href="#">Edit</a>
	My Test Dev	mytest	A1B5L4XOFCLRFA	Device	Other	<a href="#">Edit</a>

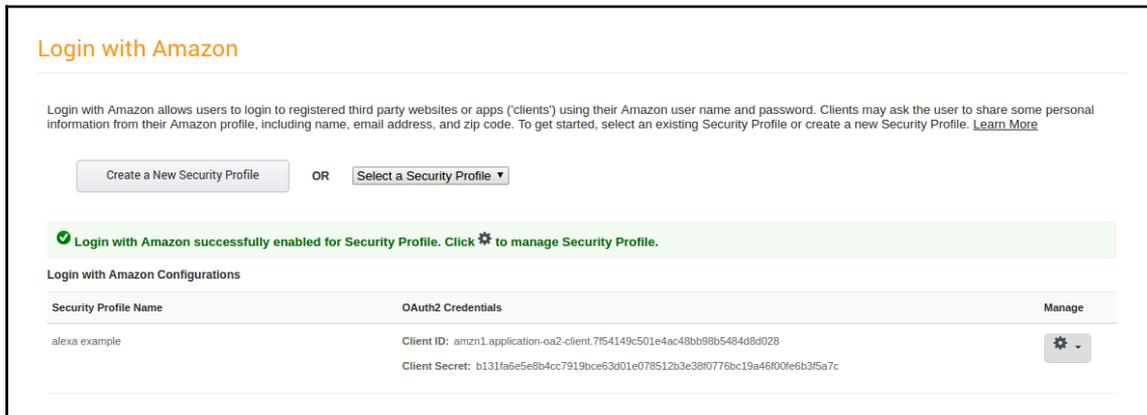
Теперь нам нужно включить только что созданный профиль безопасности. Для этого нам нужно посетить <https://developer.amazon.com/lwa/sp/overview.html> и выбрать созданный нами профиль безопасности, затем нажать кнопку «Confirm - Подтвердить». На следующем изображении показана страница, на которой вы должны активировать свой профиль безопасности:



Вам нужно ввести URL-адрес, начинающийся с <http://> или <https://> , например, <http://mytestexample.com> . После этого можно нажать на кнопку Save - Сохранить:



Рядом с образцом Alexa Voice Service Sample App Security Profile - профиля безопасности приложения Alexa Voice Service необходимо нажать «Show Client ID and Client Secret - Показать идентификатор клиента и секрет клиента». В результате вы увидите идентификатор клиента и секрет клиента. Вам нужно скопировать и вставить их в блокнот, потому что эти значения понадобятся нам позже:



На данный момент вы успешно создали свой аккаунт разработчика Amazon.

## Настройка Raspberry Pi

Прежде чем устанавливать все на нашу плату Raspberry Pi Zero W, нам нужно загрузить последний дистрибутив Raspbian на наш компьютер и использовать любой инструмент для установки программы на нашу SD-карту. После этого нам нужно вставить SD-карту в Raspberry Pi и включить его. Для следующего шага мы будем использовать репозиторий GitHub. Итак, откройте терминал и введите следующую команду:

```
git clone
```

С помощью предыдущей команды вы клонируете репозиторий [alexa-avs-sample-app](#) на свой локальный компьютер. В этом репозитории нужно найти скрипт установки и выполнить его, но перед этим нужно внести некоторые изменения. При создании нашего аккаунта разработчика и настроив его, мы прошли через три важных поля. Итак, нам нужно найти **ProductID**, **ClientID** и **ClientSecret**. После того, как мы нашли эти значения, отредактируйте файл `Automatic_install.sh` с помощью вашего любимого редактора и измените соответствующие поля.

В результате должно получиться вот так:

```
ProductID="RaspMyExample"  
ClientID="amzn. xxxxxx. xxxxxxxxxxxx"  
ClientSecret="2662rge4xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx6b4f9"
```

Далее вам нужно будет запустить скрипт, который установит все зависимости. Чтобы запустить скрипт, вам нужно сначала открыть терминал, а после перехода к загруженным файлам выполнить следующую команду:

```
.automated_install.sh
```

## Установка Alexa

Теперь нам нужно открыть два разных терминала и выполнить некоторые команды. В первом терминале мы авторизуем наше тестовое приложение с помощью AVS. Для этого вы можете запустить следующую команду:

```
cd /alexa-avs-sample-app/samples  
cd companionService && npm start
```

Если у вас не установлен `npm`, вы можете установить его, выполнив следующую команду:

```
sudo apt-get install npm
```

Вы можете видеть, что после выполнения этих команд сервер будет работать на порту 3000.

В новом окне терминала, не закрывая первое, мы запустим пример приложения, которое взаимодействует с AVS. Для этого сначала перейдите в папку с примерами, как и раньше, а затем используйте команду `mvm`:

```
cd alexa-avs-sample-app/samples  
cd javaclient && mvn exec:exec
```

На этом этапе вы должны увидеть всплывающее сообщение: «Пожалуйста, зарегистрируйте свое устройство, посетив следующий URL-адрес в веб-браузере и следуя инструкциям:

<https://localhost:3000/provision/d340f234343443h3h3h34h27>. Хотите, чтобы URL автоматически открывался в браузере по умолчанию? Вы должны нажать «Да» и открыть его в своем браузере.

На этом этапе вы должны увидеть всплывающее сообщение: «Please register your device by visiting the following URL in a web browser and following the instructions: <https://localhost:3000/provision/d340f234343443h3h3h34h27>. Would you like to open the URL automatically in your default browser? Вы должны нажать «Да» и открыть его в своем браузере.

В вашем браузере Chromium вы можете столкнуться с некоторыми ошибками, но вы можете попытаться решить их, нажав «Go Advanced - Перейти к дополнительным» и продолжить с localhost (небезопасно). Затем введите свои учетные данные на веб-сайте Amazon и нажмите «Next - Далее». Наконец, вы будете перенаправлены на веб-страницу, которая начинается с `https://localhost:3000/authresponse`, и вам нужно проверить, что в теле веб-страницы указано, что `device tokens ready` - токены устройств готовы. Вернитесь к своему приложению и нажмите ОК.

Есть еще одна вещь, которую нам нужно сделать, чтобы это заработало. Конечно, если вы не хотите, чтобы слово пробуждения активировало вашу систему, вы можете пропустить этот шаг. Однако обычно проще что-то сказать, затем произнести команду, затем нажать на кнопку или что-то в этом роде. Таким образом, вам нужно снова открыть окно терминала и вы будете использовать движок Sensory wake ward, поэтому введите следующую команду:

```
cd alexa-avs-sample-app/samples
cd wakeWordAgent/src && ./wakeWordAgent -e sensory
```

В качестве альтернативы введите это, чтобы использовать движок пробуждения KITT.AI:

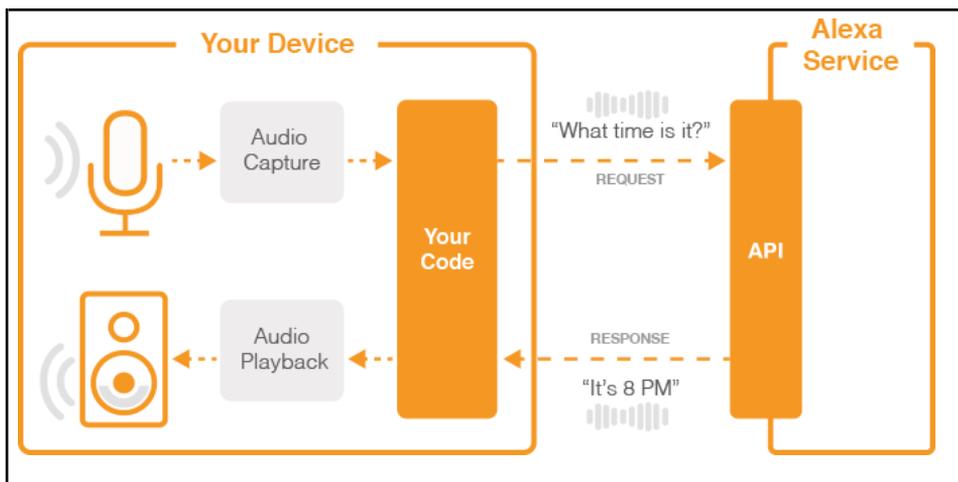
```
cd alexa-avs-sample-app/samples
cd wakeWordAgent/src && ./wakeWordAgent -e kitt_ai
```

Теперь вы должны использовать Alexa и разговаривать с ней. Вы можете активировать программу, просто сказав [Alexa](#). Затем вы можете сказать, например, "Который час?"

## Распознавание голоса

На данный момент у вас есть невероятный опыт общения с вашим Raspberry Pi и получения ответов на ваши вопросы. Это не то, что может сделать Raspberry Pi ZeroW. В связи с этим важно уточнить, как это работает. На следующем рисунке вы можете видеть, что ваша плата Raspberry Pi Zero W находится слева, и как только она запускается словом Alexa, она переходит в режим прослушивания. В режиме прослушивания она слышит все, что вы говорите, а когда вы перестаете говорить, обрабатывает введенную речь и отправляет ее обратно на серверы Amazon.

Теперь эти серверы каким-то образом находят ответы, используя искусственный интеллект и некоторые другие технологии, чтобы отправить ответ обратно на Raspberry Pi Zero W. В результате Raspberry Pi отправляет ответ в виде вывода обратно вам в ваши Bluetooth-колонки или наушники:



На предыдущем изображении вы можете видеть, что захваченный звук использует наш Raspberry Pi Zero W code, чтобы сделать запрос от AVS API и получить ответ. Как только это будет завершено, начнется воспроизведение звука. Вы понимаете, что эта процедура не может быть выполнена за 1 секунду, поэтому иногда у вас возникает задержка в 4-5 и более секунд в зависимости от вашей сети и вашего местоположения.

## Официальная Alexa против AlexaPi

Теперь пришло время сравнить эти две технологии. В этой главе мы разработали клон «Alexa» поверх нашего Raspberry Pi Zero W. Очевидно, что обе технологии великолепны и предоставляют так много функций, что преимущества перевешивают общую стоимость. Однако официальная Alexa довольно дорогой по сравнению с Raspberry Pi Zero W. Так что в этом все очки достаются Raspberry. Кроме того, AlexaPi довольно мал по сравнению с официальной Alexa. И снова очки достаются Raspberry Pi. Однако Raspberry Pi Zero W не заменит официальную Alexa, и это потому, что это недорогое устройство, которое просто работает. Этот проект лучше работает с Raspberry Pi 3, так как у него больше железа и его проще настроить, но даже при этом официальная Alexa создавалась именно для этой цели. Вероятно, у него более длительный срок службы и минимум ошибок, которые вы можете найти в такой системе.

В заключение, на мой взгляд, если вы хотите обустроить свой дом и купить больше умных вещей Amazon, купите Alexa и создайте свою систему. Однако, если у вас небольшой бюджет или вы просто хотите проверить это и немного поиграть, выберите плату Raspberry Pi.

## Сетевое администрирование

Мы увидели, как мы можем настроить нашу Raspberry Pi Zero Wboard и добавить все функции Alexa. Теперь мы обсудим, как вы можете развить этот проект и использовать сеть в вашем доме для управления им. Вы можете найти в магазине Amazon много умных вещей, таких как лампы, которые вы можете зарегистрировать в своей учетной записи Amazon Developer и получить к ним доступ через свой AVS. На следующем изображении вы можете увидеть умную лампу от Philips, которая может прослушивать вашу Alexa и выполнять действия в соответствии с вашим AVS.

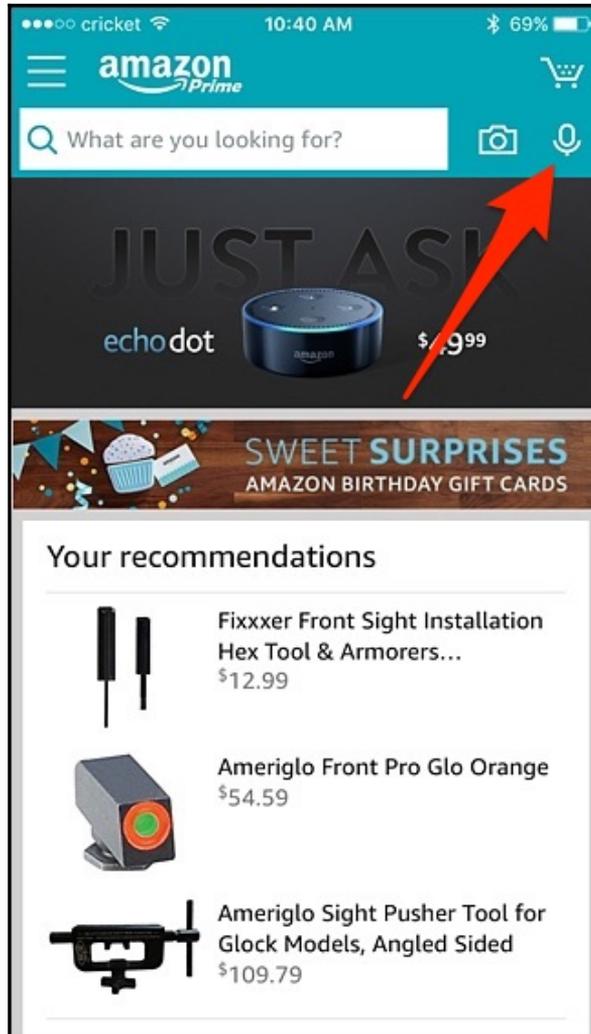


На следующем изображении вы можете увидеть систему интеллектуальных замков, которая также может общаться с вашей Alexa или любым другим устройством, которое у вас может быть, и открывать или закрывать дверь для вас:

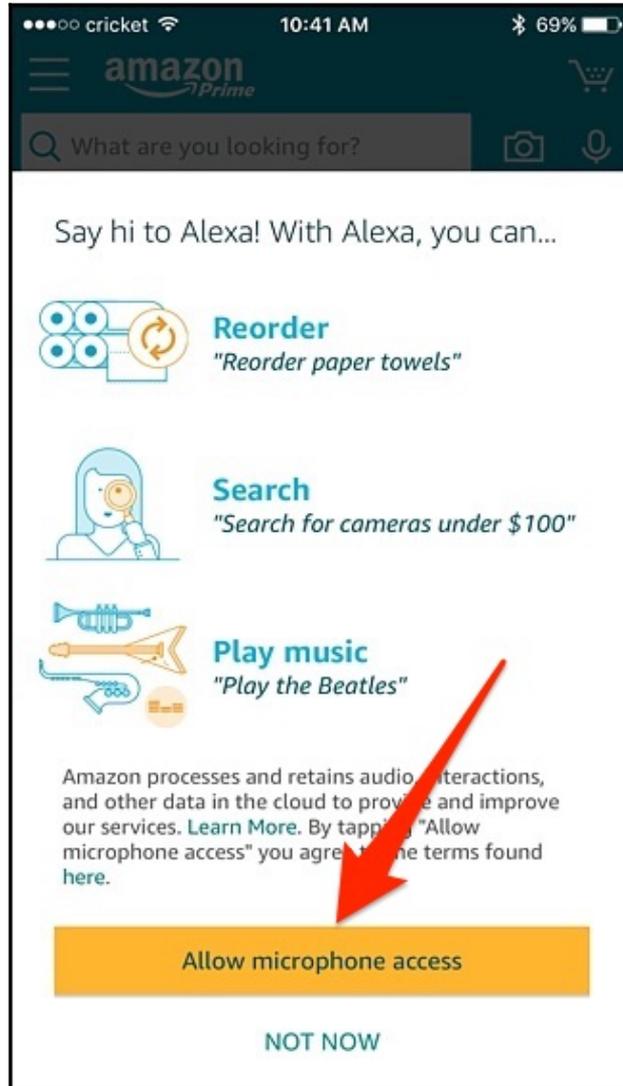


Эти устройства — только два из сотен устройств, которые есть в магазине Amazon. Вы можете найти больше устройств, посетив магазин Amazon и выполнив поиск Alexa. Все эти устройства общаются по сети и, очевидно, иногда нужно управлять ими с большого расстояния.

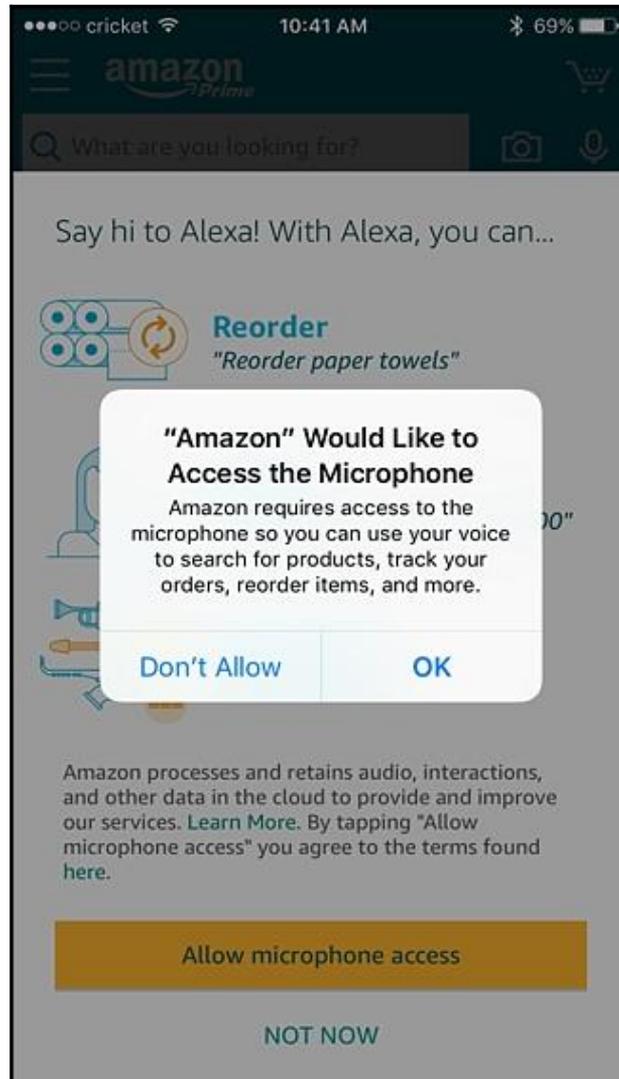
Вы можете использовать приложение UBI для пользователей Android, поскольку Amazon еще не выпустил приложение для Android. Для пользователей IOS вы можете просто загрузить приложение Amazon.



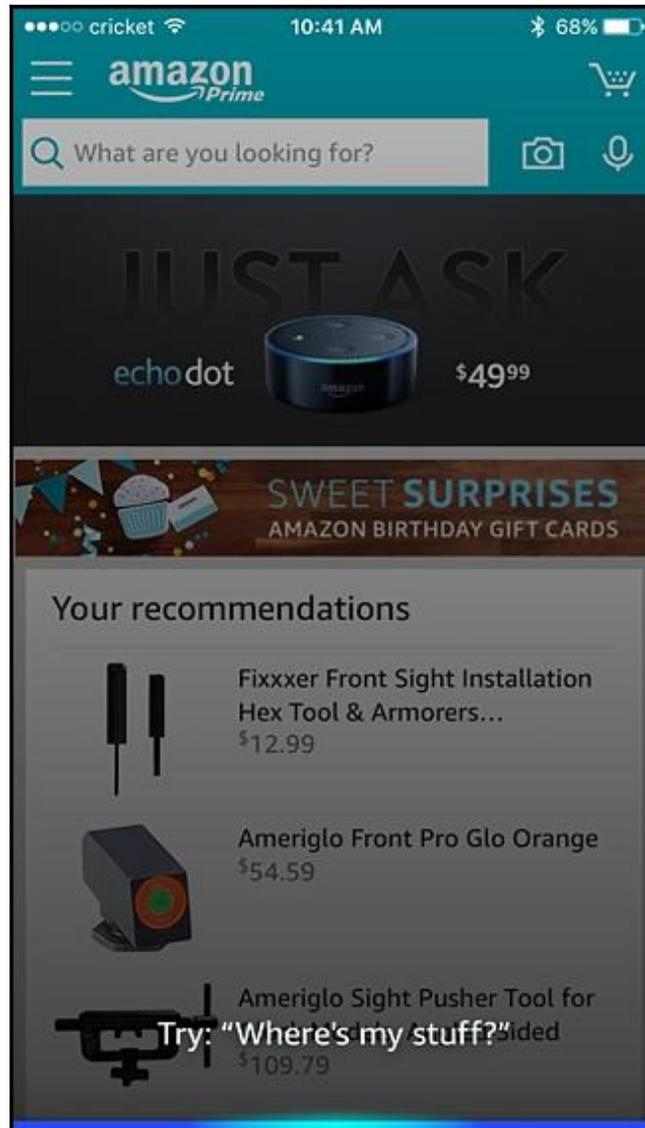
В правом верхнем углу вы найдете микрофон. Если вы нажмете на этот значок, вы увидите, что у вас есть несколько вариантов. Нажмите кнопку [Allow microphone access](#) - Разрешить доступ к микрофону, как показано на следующем изображении:



Затем разрешите все разрешения, необходимые для этого действия, как показано на следующем изображении:



Наконец, вы можете разговаривать со своей Alexa, даже когда вас нет дома. Через сети Wi-Fi и Интернет вы можете подключиться к Alexa из любой точки мира.



## **Резюме**

В этой главе мы увидели простой способ настройки Alexa на вашей плате Raspberry Pi ZeroW. Даже если Raspberry Pi Zero W не имеет таких возможностей, с помощью сервера Amazon мы разработали систему, которая слушает вас и отвечает на ваши вопросы. В следующей главе, которая будет последней, мы построим метеостанцию непосредственно с платой Raspberry Pi Zero W с использованием рыночных компонентов и покажем, что возможности Raspberry Pi Zero W безграничны.

# 10

## Метеостанция Pi

В предыдущей главе мы увидели, как установить Alexa на нашу новую плату Raspberry Pi Zero W. Alexa — отличный помощник, но возможности Raspberry Pi Zero W выше. В этой главе мы настроим метеостанцию с нашей платой Raspberry Pi Zero W. Он сможет передавать беспроводные данные и подключаться к другим устройствам, которые у вас могут быть, расширяя ваш умный дом еще одним беспроводным проектом. Мы будем использовать Sense HAT, который обычно используется во многих проектах. В этой главе мы рассмотрим следующие темы:

- Модуль Sense HAT
- Настройка метеостанции
- Начальное состояние
- Запуск

### Модуль Sense HAT

Sense HAT — это дополнительная плата для Raspberry Pi 2 и 3, но она также работает с Raspberry Pi Zero. Она была сделана специально для миссии Astro Pi.

Модуль Sense HAT выглядит следующим образом:



Вы можете купить его на официальном сайте <https://www.raspberrypi.org/products/sense-hat/>. Sense HAT позволяет Pi ощущать окружающий мир. С помощью Sense HAT мы можем отображать сообщения и изображения, управлять ориентацией, собирать данные датчиков и реагировать на движение. В результате Sense HAT — это самый простой способ добавить массу датчиков на плату Raspberry Pi Zero W без покупки и подключения перемычек к макетной плате.

Sense HAT имеет следующие функции:

- Ориентация (рыскание, тангаж и крен) с помощью акселерометра, трехмерного гироскопа и магнитометра.
- Температура
- Влажность
- Давление

Он также оснащен светодиодной матрицей 8x8, которая позволяет отображать данные с помощью различных датчиков. С помощью светодиодной матрицы вы также можете определить направление геомагнитного севера, запрограммировав компас с помощью магнитометра, или вы можете играть в такие игры, как старый классический тетрис. Кроме того, вы можете использовать джойстик, чтобы пользователь-человек мог взаимодействовать с программами на вашей плате Raspberry Pi. Как вы увидите в этой главе, писать программы для Sense HAT довольно просто, используя библиотеку Python. Ниже приведены некоторые технические подробности модуля:

- **Гироскоп:** Датчик угловой скорости (dps) - ~245/500/2000
- **Акселерометр:** Датчик линейного ускорения (g) - ~2/4/8/16
- **Магнитометр:** Магнитный датчик (Гаусс) - ~4/8/12/16
- **Барометр:** абсолютный диапазон 260@1260 гПа (точность зависит от температуры и давления, ~0,1 гПа при нормальных условиях)
- **Датчик температуры:** точность до ~2°C в диапазоне 0-65°C
- **Датчик относительной влажности:** точность до ~4,5% в диапазоне относительной влажности 20-80%, точность до ~0,5°C в диапазоне 15-40°C
- Светодиодный матричный дисплей 8x8
- Маленький пятикнопочный джойстик
- Подставки могут быть пластиковыми или металлическими.

## Метеостанция

Прежде чем начать, вам необходимо подключить модуль Sense HAT к вашему Raspberry Pi Zero W. Обычно Raspberry Pi Zero и Zero W поставляются с не припаянными контактами. Поэтому, если вы ничего не меняли и не устанавливали, вам, вероятно, придется припаять разъем GPIO, на который вы сможете легко установить Sense HAT. На самом деле вам решать, как настроить этот модуль, в зависимости от вашего проекта и вашего оборудования.

Для начала давайте обновим нашу плату Raspberry Pi Zero W последними программами и обновлениями. На плате Raspberry Pi Zero и Zero W есть некоторые проблемы, и необходимы обновления. Сначала выполните следующую команду, чтобы обновить Raspberry последними пакетами:

```
sudo apt-get update
```

Теперь выполните следующую команду, чтобы обновить операционную систему до последней версии:

```
sudo apt-get upgrade
```

Теперь выполните следующую команду, чтобы максимально снизить вероятность ошибок и последующих ошибок:

```
sudo rpi-update
```

Затем добавьте следующую строку в конец файла /boot/config.txt:

```
core_freq=250
```

Вы можете отредактировать файл с помощью редактора [vim](#) или [nano](#). После этого все должно быть в порядке. Итак, продолжим установку всех необходимых библиотек и проведем полную настройку. Следующим шагом здесь является установка программного пакета Sense HAT, чтобы мы могли взаимодействовать с модулем. Для этого просто запустите:

```
sudo apt-get install sense-hat
```

Теперь важно перезагрузить вашу систему. Вы можете сделать это с помощью следующей команды:

```
sudo reboot
```

После того, как вы вернетесь к Raspberry Pi Zero W, вы должны убедиться, что все прошло нормально и что ваш модуль установлен правильно. Для этого напишем простую программу. Создайте файл `sensehat_test` в своем любимом редакторе (например, с помощью `vim`):

```
sudo vim ~/sensehat_test.py
```

Теперь нам нужно подключить необходимые библиотеки. Во-первых, из библиотеки `sense_hat` нам нужно включить модуль `Sense HAT`, который позволит нам взаимодействовать с модулем `Sense HAT`. Для этого добавим первую строчку кода:

```
from sense_hat import SenseHat
```

Следующим шагом является создание ссылки на библиотеку, и поэтому мы инициализируем переменную, которая позволит нам взаимодействовать с дальнейшими обращениями к этой библиотеке. Сделайте это с помощью следующей команды:

```
sense = SenseHat ()
```

Наконец, с помощью следующей команды вы увидите «Hello World» на светодиодах Sense HAT.

Мы можем написать сообщение прямо в Sense HAT:

```
sense.show_message("Hello World")
```

Теперь сохраните и закройте файл. Чтобы запустить ваш скрипт, вы должны просто выполнить следующую команду:

```
sudo python ~/sensehat_test.py
```

На этом этапе, как мы уже говорили ранее, вы должны увидеть текст «Hello world» в модуле Sense HAT. Если нет, попробуйте заменить контакты и правильно вставить их в распиновку Raspberry Pi Zero W. Если вы получаете ошибки, попробуйте перезагрузить Raspberry Pi и снова запустить этот скрипт Python. Далее вы можете написать полный код, чтобы проверить это:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.show_message("Hello World")
```

На этом этапе Sense HAT должен быть готов, и теперь мы можем перейти ко второму шагу, где мы установим и настроим метеостанцию. Создайте новый файл в своем любимом редакторе:

```
sudo vim ~/weather_script.py
```

Теперь начните писать код. Прежде всего, как и в каждом скрипте Python, нам нужно импортировать все необходимые библиотеки. Мы можем сделать это с помощью следующих строк кода:

```
#!/usr/bin/python
from sense_hat import SenseHat
import time
import sys
```

Библиотека [sense\\_hat](#) необходима для взаимодействия с Sense HAT, потому что мы будем читать с датчиков. Библиотека [Time](#) из стандартных библиотек Python необходима для использования ее для многих операций со временем, например, для добавления задержки. Наконец, библиотека [sys](#) позволяет нам получить доступ к переменным и функциям, которыми обычно управляет интерпретатор. Как мы уже говорили ранее, первая строка, которую мы видим в следующем блоке кода, инициализирует Sense HAT, поэтому мы можем выполнять дальнейшие вызовы функций, относящихся к модулю Sense HAT:

```
sense = SenseHat()
```

Следующая строка очищает светодиодную матрицу в модуле Sense HAT, поэтому мы можем иметь пустую матрицу для записи результатов:

```
sense.clear()
```

В этом блоке кода мы создадим цикл `while`, который будет работать вечно и остановится при нажатии `Ctrl+C`. Мы можем сделать это с помощью Python, используя `Indent`, где это необходимо, со следующим блоком кода:

```
try:
    while True:
```

Следующим шагом будет получение температуры из библиотеки Sense HAT. Мы можем получить это с помощью одной строки кода, используя библиотеку, которую мы импортировали в начале. Обратите внимание, что выходные данные представлены в градусах Цельсия, поэтому, если вы хотите преобразовать их в градусы Фаренгейта, вы должны сделать преобразование самостоятельно:

```
temp = sense.get_temperature()
```

Теперь вам может понадобиться преобразовать температуру в ближайшее десятичное число. Вы можете сделать это, используя следующий код:

```
round(temp, 1)
```

Итак, подводя итог, если вы хотите, чтобы температура выражалась в градусах Цельсия, вам нужно выполнить следующую команду:

```
temp = round(temp, 1)
```

С другой стороны, если вы хотите, чтобы температура выражалась в градусах Фаренгейта, вам нужно выполнить следующую команду:

```
temp = 1.8 * round(temp, 1) + 32
```

Последняя строка кода, которую мы напишем, предназначена для вывода значения температуры. Если вы хотите распечатать температуру по Цельсию, введите следующую команду:

```
print("Temperature C",temp)
```

Вы также можете заменить C на F, если вы ранее преобразовали температуру в градусы Фаренгейта. Помимо температуры, мы получим значения влажности и давления от Sense HAT. Sense HAT предоставляет эти значения точно так же, как и температура, поэтому это практически те же строки кода:

```
humidity = sense.get_humidity()
humidity = round(humidity, 1)
print("Humidity :",humidity)

pressure = sense.get_pressure()
pressure = round(pressure, 1)
print("Pressure:",pressure)
```

Последняя строка кода использует библиотеку времени. Нам нужно приостановить выполнение кода на одну секунду, чтобы мы могли видеть поток данных. Мы сделаем это с помощью следующего кода:

```
time.sleep(1)
```

Когда срабатывает `KeyboardInterrupt`, мы игнорируем исключение, поэтому мы можем заставить скрипт оставить цикл `while` запущенным. Мы сделаем это с помощью следующего кода:

```
except KeyboardInterrupt:  
    pass
```

Без предшествующих строк кода, когда пользователь нажимал `Ctrl+C`, программа заканчивалась. Ниже приведен полный код:

```
#!/usr/bin/python  
from sense_hat import SenseHat  
import time  
import sys  
  
sense = SenseHat()  
sense.clear()  
  
try:  
    while True:  
        temp = sense.get_temperature()  
        temp = round(temp, 1)  
        print("Temperature C",temp)  
  
        humidity = sense.get_humidity()  
        humidity = round(humidity, 1)  
        print("Humidity :",humidity)  
  
        pressure = sense.get_pressure()  
        pressure = round(pressure, 1)  
        print("Pressure:",pressure)  
  
        time.sleep(1)  
except KeyboardInterrupt:  
    pass
```

Сохраните и выйдите из редактора с именем файла `weather_script.py`.

Затем запустите скрипт Python:

```
python weather_script.py
```

Теперь вы должны увидеть температуру, влажность и давление на экране, например:

```
('Temperature C', 30.0)
('Humidity :', 39.8)
('Pressure:', 1025.7)
```

На этом этапе, если все пойдет хорошо, вы сможете увидеть эти значения на экране. Если вы хотите изменить способ их отображения, вы всегда можете вернуться к файлу сценария Python, чтобы отредактировать способ их печати. В следующем разделе мы отобразим эти значения в более удобном виде.

Изменение написанного нами Python-скрипта и отображение данных на светодиодной матрице в Sense HAT не составляет большого труда. Уже есть вызов Sense HAT, где можно вывести строку в матрицу. Итак, что нам нужно сделать, так это просто объединить все данные, которые мы хотим передать Sense HAT, в одну строку. Первый шаг — отредактировать файл, который мы создали ранее. Сделайте это с помощью vim или любого другого редактора, который вам нравится; например, используя vim, мы можем сделать это с помощью следующего кода:

```
sudo vim weather_script.py
```

Затем мы используем команду, которая переводит вашу программу в спящий режим на одну секунду:

```
time.sleep(1)
```

Нам нужно добавить новую строку кода. Эта новая строка кода объединяет все данные с датчика, которые мы получаем при запуске программы, и отображает их в одной строке в Sense HAT:

```
sense.show_message("Temperature C" + str(temp) + "Humidity:" +
str(humidity) + "Pressure:" + str(pressure), scroll_speed=(0.08),
back_colour= [0,0,200])
```

Важно записать предыдущую команду в одну строку кода. В смысле, выводим сообщение. Сообщение представляет собой конкатенацию " " строк, таких как "Temperature C" и str(), которые представляют собой целые числа, преобразованные в строки с помощью команды str() из библиотеки Python. Скорость прокрутки установлена на 0,08, но мы можем изменить ее на большую или меньшую. В зависимости от того, как был изменен текст, цвет был установлен на синий. Далее нам нужно сделать вызов смысла понятным с помощью следующего кода:

```
sense.clear().
```

Таким образом, мы обеспечим полную очистку светодиодной матрицы, если по какой-либо причине завершим скрипт. Полный код отображается следующим образом:

```
#!/usr/bin/python
from sense_hat import SenseHat
import time
import sys

sense = SenseHat()
sense.clear()

try:
    while True:
        temp = sense.get_temperature()
        temp = round(temp, 1)
        print("Temperature C",temp)

        humidity = sense.get_humidity()
        humidity = round(humidity, 1)
        print("Humidity :",humidity)

        pressure = sense.get_pressure()
        pressure = round(pressure, 1)
        print("Pressure:",pressure)

        sense.show_message("Temperature C" + str(temp) + "Humidity:" +
str(humidity) + "Pressure:" + str(pressure), scroll_speed=(0.08),
back_colour= [0,0,200])

        time.sleep(1)
except KeyboardInterrupt:
    pass

sense.clear()
```

Теперь сохраните и закройте файл, который вы редактировали. Запустите скрипт еще раз со следующим кодом:

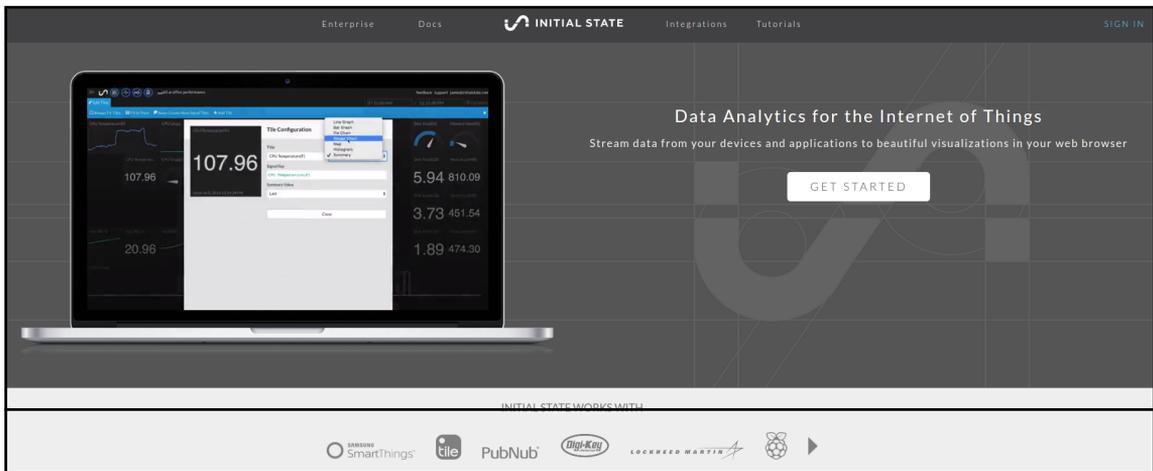
```
sudo python weather_script.py
```

Теперь вы должны увидеть прокрутку текста в области светодиодной матрицы Sense HAT. Вы заметите, что здесь есть одна проблема. Вы не можете отобразить все данные сразу. Это можно сделать с помощью внешнего монитора или в следующем разделе главы. Теперь вы увидите, как вы можете улучшить свою метеостанцию Raspberry Pi с помощью начального состояния.

## Начальное состояние

Мы продолжим создание метеостанции с Initial State, веб-сайтом, который служит хранилищем и анализом данных для проектов IoT с использованием Arduino, Raspberry Pi и подобных. Существуют и другие службы, такие как Initial State, но не все из них поддерживают Sense HAT. Первым шагом здесь является создание аккаунта с начальным состоянием. Я объясню все в следующих изображениях. Посетите <https://www.initialstate.com/> и зарегистрируйте свой аккаунт.

На следующем изображении показана страница веб-сайта INITIAL STATE. Нажмите на ссылку SIGN IN - ВХОД в правом верхнем углу:



Нажмите на ссылку регистрации и заполните регистрационную форму, которую вы видите рядом, указав свой Email и Password - пароль. Вы также можете запросить двухэтапную аутентификацию, используя номер своего мобильного телефона. Рекомендуется добавить еще один уровень безопасности в свою учетную запись:

### Account Registration

Email [Sign Into An Existing Account](#)

Password 8+ letters or numbers

Repeat Password An Anti-Typo Technology™

Subscription Code ...if you have one.

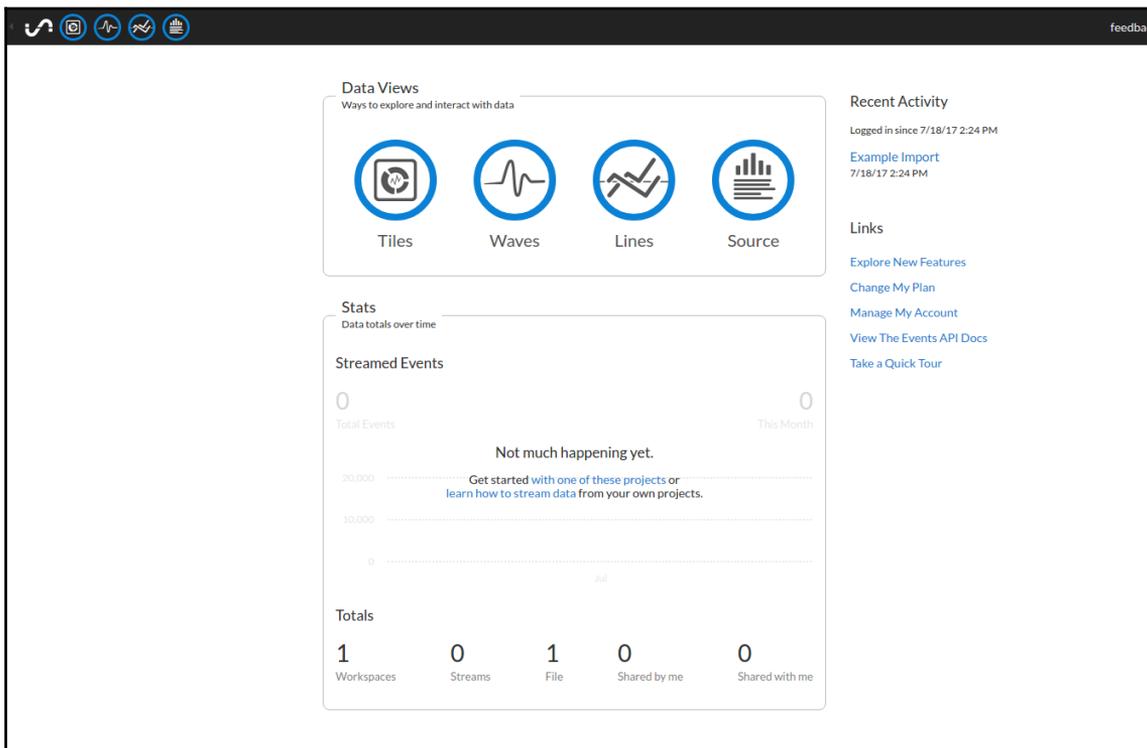
Use two-factor authentication to sign in

SMS-Capable Phone Number

Stay signed in

Registering implies agreement to the [Terms of Use](#).

Когда вы заполните форму, вы можете нажать на опцию «Stay signed - Остаться в системе», чтобы войти в систему автоматически. Вы будете перенаправлены на панель инструментов, которая должна выглядеть примерно так, как показано на следующем изображении:



Затем нажмите на адрес электронной почты в правом верхнем углу и прокрутите страницу вниз, где вы можете найти опцию, позволяющую создать новый ключ. Кнопка должна выглядеть следующим образом:



## Метеостанция Pi

---

Нажмите «Create A New Key - Создать новый ключ», а затем запишите где-нибудь на бумажке созданный ключ. Теперь вернитесь к плате Raspberry Pi Zero W и откройте окно терминала. Теперь нам нужно установить стример [Initial State Python](#). Перед выполнением команды для захвата начального состояния вы можете проверить веб-сайт с помощью <https://get.initialstate.com/python>.

Затем вы можете просто запустить следующий код:

```
curl -sSL
```

Чтобы получить данные из исходного состояния на Raspberry Pi Zero W через SSH, откажитесь от загрузки примера кода и после завершения установки перейдите и отредактируйте файл вашей метеостанции. Вы можете сделать это с помощью следующего кода:

```
sudo vim weather_script.py
```

Предыдущий код должен быть вставлен ниже строки:

```
import sys
```

Добавьте в скрипт следующую строку кода, которая будет включать пакет стримера Initial State:

```
from ISStreamer.Streamer import Streamer
```

Теперь мы можем использовать этот пакет для настройки вашего скрипта Python и подключения его к [Initial State API](#) перед строкой:

```
sense = SenseHat()
```

Добавьте следующую строку кода, чтобы вы могли создать стример и инициализировать соединение. Вы должны заменить YOUR\_KEY\_HERE соответствующим ключом, который вы получили в нижней части веб-сайта исходного состояния до:

```
logger = Streamer(bucket_name="Sense Hat Sensor Data",  
access_key="YOUR_KEY_HERE")
```

Затем просмотрите весь свой код и найдите все экземпляры:

```
print(
```

Все экземпляры print должны быть заменены следующим кодом:

```
logger.log(
```

Это изменение означает, что вы печатаете сообщения и информацию на сервер исходного состояния, а не на свой экран. Наконец, ваш код должен выглядеть следующим образом:

```
#!/usr/bin/python
from sense_hat import SenseHat
import time
import sys
from ISStreamer.Streamer import Streamer

sense = SenseHat()
logger = Streamer(bucket_name="Sense Hat Sensor Data",
access_key="YOUR_KEY_HERE")
sense.clear()

try:
    while True:
        temp = sense.get_temperature()
        temp = round(temp, 1)
        logger.log("Temperature C",temp)

        humidity = sense.get_humidity()
        humidity = round(humidity, 1)
        logger.log("Humidity :",humidity)

        pressure = sense.get_pressure()
        pressure = round(pressure, 1)
        logger.log("Pressure:",pressure)

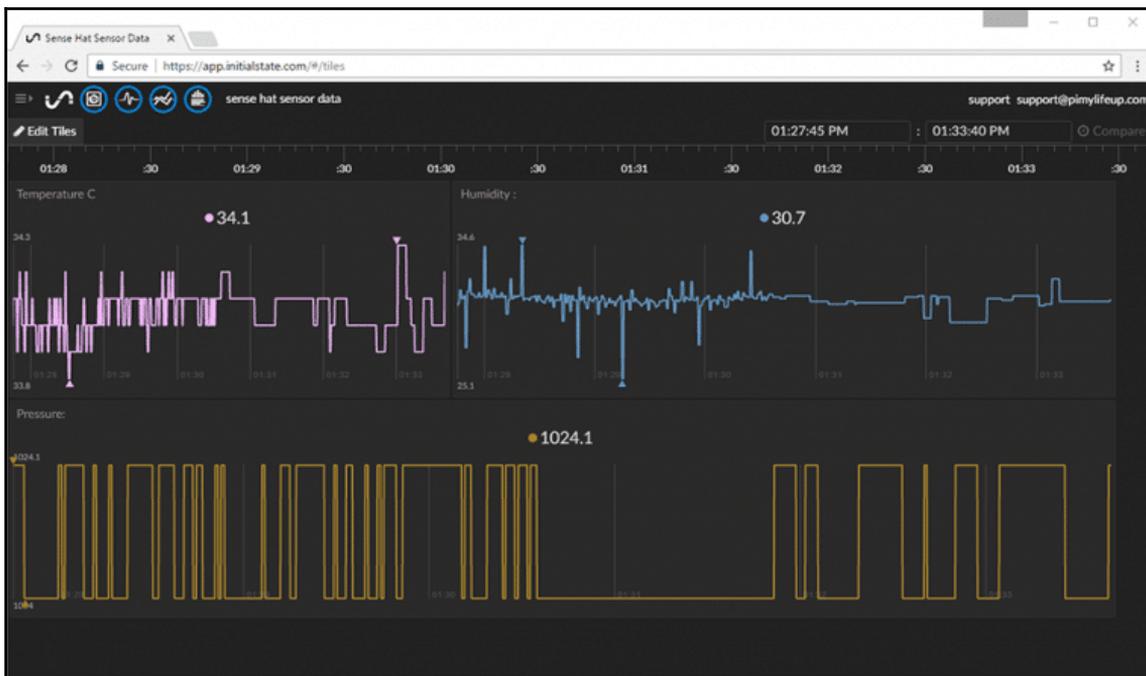
        time.sleep(1)
except KeyboardInterrupt:
    pass
```

Сделав это, сохраните и закройте файл, а затем запустите новый файл сценария со следующим кодом:

```
python weather_script.py
```

Новый код будет напрямую отправлять данные обратно на сервер исходного состояния. Если вы получаете какие-либо ошибки, проверьте подключение к Интернету. Теперь вернитесь на веб-сайт исходного состояния и откройте панель инструментов. Вы можете увидеть новые данные в верхнем левом углу. Выберите свои данные, а затем нажмите «[Tiles - Плитки](#)», чтобы просмотреть потрясающий график ваших данных.

Теперь данные визуализируются, как показано на следующем изображении:



## Запуск

На этом этапе ваша станция WeatherPi должна быть готова. Есть еще одна вещь, которую нужно сделать здесь.

Мы можем настроить запуск нашего скрипта Python при загрузке платы Raspberry Pi Zero W. Для этого нам нужно установить пакет с именем `dos2unix`, который на самом деле преобразует окончания строк в стиле DOS во что-то дружелюбное для Unix. Кроме того, вы можете использовать `crontab` (установите его, если у вас его нет) или отредактируйте файл `/etc/init.d/`, чтобы добавить несколько строк и запустить скрипт с другими встроенными службами Raspberry Pi Zero W. Чтобы установить упомянутый ранее пакет, нам нужно запустить следующий код:

```
sudo apt-get install dos2unix
```

Затем нам нужно настроить наш скрипт Python как службу, чтобы мы могли запустить его при загрузке Raspberry Pi Zero W. Для этого мы создадим новый скрипт, который позволит нам запускать/останавливать его и обрабатывать скрипт Python так, как мы хотим. Как всегда, откройте свой любимый редактор и начните создавать новый скрипт. Мы будем использовать `vim` в качестве нашего редактора:

```
sudo vim /etc/init.d/weatherstation
```

Пришло время написать здесь немного кода. Начните с написания следующего кода:

```
#!/bin/bash
### BEGIN INIT INFO
# Provides:          weatherstation
# Required-Start:
# Required-Stop:
# Default-Start:    2 3 4 5
# Default-Stop:     0 1 6
# Short-Description: Start/stops the weatherstation
# Description:      Start/stops the weatherstation
### END INIT INFO

DIR=/home/pi
DAEMON=$DIR/weather_script.py
DAEMON_NAME=weatherstation

DAEMON_USER=root

PIDFILE=/var/run/$DAEMON_NAME.pid

. /lib/lsb/init-functions

do_start () {
    log_daemon_msg "Starting system $DAEMON_NAME daemon"
    start-stop-daemon --start --background --pidfile $PIDFILE --make-
pidfile --user $DAEMON_USER --chuid $DAEMON_USER --startas $DAEMON
    log_end_msg $?
}

do_stop () {
    log_daemon_msg "Stopping system $DAEMON_NAME daemon"
    start-stop-daemon --stop --pidfile $PIDFILE --retry 10
    log_end_msg $?
}

case "$1" in
    start|stop)
        do_${1}
        ;;
    restart|reload|force-reload)
        do_stop
```

```
        do_start
        ;;
status)
    status_of_proc "$DAEMON_NAME" "$DAEMON" && exit 0 || exit $?
    ;;
*)
    echo "Usage: /etc/init.d/$DAEMON_NAME {start|stop|restart|status}"
    exit 1
    ;;
esac
exit 0
```

После написания всего этого кода в редакторе вы можете сохранить файл и выйти. Чтобы убедиться, что все работает гладко, вам нужно запустить [dos2unix](#) для файла, который вы только что создали. Для этого вам нужно запустить следующий код:

```
sudo dos2unix /etc/init.d/weatherstation
```

С помощью этой команды вы убедитесь, что файл записан правильно и все в порядке. Теперь вам нужно изменить разрешения в вашем скрипте Python. Если вы этого не сделаете, ваш скрипт `bash` не будет работать. Чтобы сделать это и изменить свои разрешения, выполните следующую команду:

```
sudo chmod 755 /home/pi/weather_script.py
```

Следующим шагом является модификация и изменение разрешений `bash`-скрипта вашей метеостанции.

Вы должны дать права на выполнение, и вы можете легко сделать это с помощью команды `chmod`:

```
sudo chmod +x /etc/init.d/weatherstation
```

Создайте символическую ссылку между вашим сценарием `bash` и папками `rc.d`:

```
sudo update-rc.d weatherstation defaults
```

Теперь запустите скрипт Python:

```
sudo service weatherstation start
```

К этому моменту вы должны быть готовы. Служба вашей метеостанции должна автоматически запускаться при загрузке. Вы можете проверить это, перезагрузив плату Raspberry Pi Zero W. Кроме того, вы можете взаимодействовать со своей метеостанцией с помощью дополнительных команд; например, вы можете запустить службу с помощью следующего кода:

```
sudo service weatherstation start
```

В качестве альтернативы, если по какой-либо причине вы хотите остановить работающую службу, вы можете просто запустить следующую команду:

```
sudo service weather station stop
```

Теперь осталось выполнить еще две команды. Итак, чтобы перезагрузить метеостанцию, достаточно выполнить следующую команду:

```
sudo service weatherstation reload
```

Наконец, чтобы получить текущий статус метеостанции, выполните следующую команду:

```
sudo service weather station status
```

Таким образом, вы сможете определить, работает ли она или остановилась.

## Резюме

Из этой главы вы должны были узнать, как создавать, настраивать и улучшать свою личную метеостанцию с помощью [Sense HAT](#). Есть несколько модулей, которые можно купить и добавить в свой проект. [Sense HAT](#) дает нам возможность создать эту метеостанцию, а поскольку Raspberry Pi Zero W такой маленький, его можно использовать для создания портативной метеостанции. Поскольку это последняя глава книги, вы узнали, что существует множество различных модулей, которые вы можете купить для своей платы Raspberry Pi, чтобы расширить ее возможности. Предполагая, что вы можете работать с беспроводной связью, как вас учили в этой книге, вы можете создавать беспроводные проекты, используя всего лишь плату за 10 долларов. Это отличная возможность создавать полезные проекты, развивая навыки и ваши знания.