



[Главная](#) → [Публикации](#)

Raspberry Pi, Python и распознавание голоса

Одна серьезная проблема относительно покупки еды у многих - это когда ты полностью забываешь что купил и когда она испортится. Мы используем Raspberry Pi, Python и распознавание голоса, чтобы создать систему, которая добавляет или удаляет элементы в списке того, что находится в холодильнике.

Оглавление [[скрыть](#)]

- Что нам пригодится
- Установка библиотек Python и настройка аудио
- Как работает голосовой контроль списка холодильника
- Полный код
- Само устройство

Что нам пригодится

Аппаратные средства:

- 1 - Raspberry Pi модель B (с Wi-Fi)
- 1 - USB аудиокарта
- 1 - Микрофон

Программное обеспечение:

1 - Python 3

1 - Библиотека распознавания речи

Python

1 - Библиотека PyAudio Python

Установка библиотек Python и настройка аудио

Прежде чем мы сможем запустить нашу программу на Python, нам необходимо установить две библиотеки - распознавание речи и PyAudio. Чтобы установить эти две библиотеки, вам нужно запустить следующие две команды в окне терминала:

```
$ sudo pip install SpeechRecognition  
$ sudo pip install PyAudio
```

Если у вас возникли проблемы с установкой PyAudio, вы можете установить его, используя следующую последовательность инструкций:

```
$ sudo apt-get install git  
$ sudo git clone http://people.csail.mit.edu/hayk/  
$ sudo apt-get install libportaudio0 libportaudiolib  
$ sudo apt-get install python-dev  
$ cd pyaudio  
$ sudo python setup.py install
```

После того, как эти две библиотеки будут установлены, вам необходимо отключить встроенный аудио драйвер на Raspberry Pi, так как это может помешать работе PyAudio. Для этого

начните с открытия нового терминала
и выполните следующие команды:

```
$ cd /etc/modprobe.d  
$ sudo nano alsa-blacklist.conf
```

Nano - простой текстовый редактор для терминала, и когда он загружается, вам нужно только ввести одну строку:

```
blacklist snd_bcm2835
```

Нажмите Ctrl + X, чтобы выйти из nano и сохранить файл с именем alsa-blacklist.conf. Этот простой файл отключает аудиосистему Broadcom от Raspberry Pi, поэтому единственной аудиосистемой, доступной на Pi, является звуковая карта USB.

Как работает голосовой контроль списка холодильника

Сценарий Python начинается с импорта модуля распознавания речи, который используется для преобразования устных слов в строку. Когда модуль был импортирован, мы создаем объект "r", который является объектом распознавания речи и используется для записи звука с микрофона, а затем запрашивает преобразование. После определения объекта распознавания

речи мы также определяем наши переменные, включая список элементов, команду, текущий элемент и массив, в котором содержатся анализируемые команды.

```
import speech_recognition as sr

r = sr.Recognizer()
items = dict()
command = ""
item = ""
```

При первоначальной конфигурации следующий кусок кода, который должен быть выполнен, является основным циклом. Первая задача в цикле состоит в том, чтобы сообщить пользователю говорить, напечатав слово «Speak», а затем создайте аудио-объект под названием «audio», который будет содержать наш поток с микрофона.

```
while(1):
    with sr.Microphone() as source:
        print("Speak:")
        audio = r.listen(source)
```

Когда микрофон обнаружил звук и закончил запись (запись прекращается, когда уровень звука падает ниже порогового значения), он передает записанный звук в наш объект распознавателя. Поступая таким образом, **r** будет использовать службы Google, чтобы попытаться преобразовать аудио в предложение,

которое затем передается переменной, называемой **speechString**. Весь этот код выполняется в блоке **try/except**, если звук не был понят или если услуга недоступна. Полученная строка также анализируется на **parsedCommands**, где разделителем является пробел. Так что, если слова «добавить бекон» будут сказаны, результатом будет то, что `parsedCommands[0]` будет «добавить», а `parsedCommands[1]` будет «беконом».

```
try:
    speechString = r.recognize_google(audio)
    parsedCommands = speechString.split()
except sr.UnknownValueError:
    print("Could not understand audio")
except sr.RequestError as e:
    print("Could not request results; {0}".format(e))

if len(parsedCommands) > 0:
    command = parsedCommands[0]

    if len(parsedCommands) > 1:
        item = parsedCommands[1]
```

Теперь, когда у нас есть наши анализируемые команды и элементы, мы можем добавить их в наш список товаров. Однако, чтобы сохранить чистоту, мы выполняем несколько проверок, которые выполняют следующие действия:

- Если элемент уже существует и добавляется, то затем увеличиваем значение элемента

- Если элемент не существует и добавлен, добавить элемент в список
- Если элемент уже существует и удаляется, то вычитать 1, если сумма больше 1
- Если элемент уже существует, указание удалить, и если остается только один элемент, удалить элемент
- Если элемент не существует, игнорировать команду

```
if(command == "add"):
    if item in items:
        items[item] = str(int(items[item]) + 1)
    else:
        items[item] = str(1)

    print(item + " added")

if(command == "remove"):
    if item in items:
        if(int(items[item]) > 1):
            items[item] = str(int(items[item]) - 1)
        else:
            try:
                items.pop(item, None)
            except:
                pass
    print(item + " removed")
```

Последняя команда в этом простом скрипте - это «display», который печатает содержимое элемента переменной на дисплее.

```
if(command == "display"):
    print(items)
```

Полный код

```
import speech_recognition as sr

r = sr.Recognizer()
items = dict()
command = ""
item = ""

while(1):
    with sr.Microphone() as source:
        print("Speak:")
        audio = r.listen(source)
    try:
        speechString = r.recognize_google(audio)
        parsedCommands = speechString.split()
    except sr.UnknownValueError:
        print("Could not understand audio")
    except sr.RequestError as e:
        print("Could not request results; {0}".format(e))

    if(len(parsedCommands) > 0):
        command = parsedCommands[0]

    if(len(parsedCommands) > 1):
        item = parsedCommands[1]

    if(command == "add"):
        if item in items:
            items[item] = str(int(items[item]) + 1)
        else:
            items[item] = str(1)

        print(item + " added")

    if(command == "remove"):
        if item in items:
            if(int(items[item]) > 1):
                items[item] = str(int(items[item]) - 1)
```

```
else:
    try:
        items.pop(item, None)
    except:
        pass
    print(item + " removed")

if(command == "display"):
    print(items)

command = ""
item = ""
days = ""
parsedCommands.clear()
```

Само устройство



Этот проект реализован на основе Raspberry Pi и не требует никакой схемы или оборудования помимо микрофона и дисплея.

Обычный монитор или телевизор можно использовать, но это не очень практично для монтажа, поэтому в этом проекте был взят небольшой 3,5-

дюймовый дисплей Raspberry Pi с разрешением 480x320. Хотя это слишком мало для нормального использования, оно идеально подходит для работы с командной строкой. А если наша "малина" программируется и используется по сети (используя SSH), вы можете запустить программу Python с любого компьютера, подключенного к Интернету.

По теме

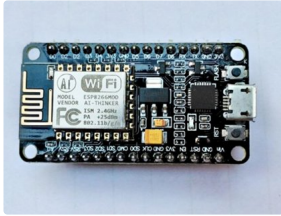
- [Как написать и запустить код Python на Raspberry Pi](#)
- [Как подключить Raspberry Pi к Ардуино?](#)
- [Создаем бота Telegram и используем с Raspberry Pi](#)

3

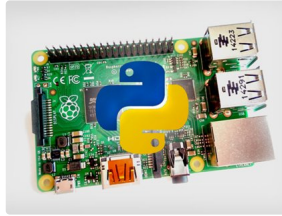
6

📅 25.05.2018 | Публикации | Python, Raspberry, Звук, Программирование | **Автор:** Дмитрий Васильев

Смотрите также



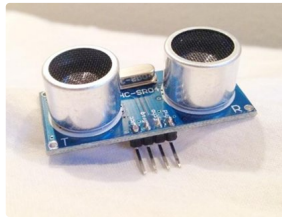
WiFi модуль
Arduino:
варианты
устройств и
их...



Программирова
платы
Raspberry Pi
через...



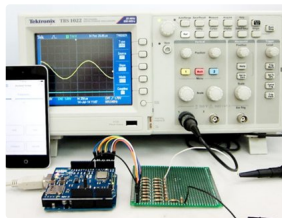
Подключение
Raspberry Pi к
Ардуино:
настройка...



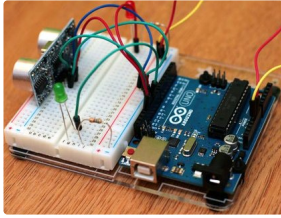
Arduino hc hc
sr04 или
собираем
простой...



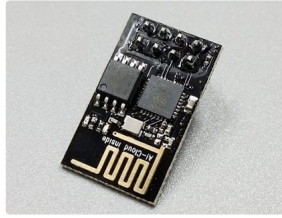
Ардуино
модули: виды,
описание,
примеры...



Генератор
частоты на
Ардуино:
сборка...



Arduino:
дополнительны
модули,
расширяющие..



ESP8266 уроки
контроля
устройств
через интернет



Raspberry Pi и
Telegram:
создаем бота
для...

← Используем сервопривод для
Ардуино проектов

Клон Ардуино AAdiuno размером с
“пальчиковую” батарейку →

Arduino+

ArduinoPlus.ru — сайт о Arduino, Raspberry, Wemos и других микроконтроллерах, программировании, разработке устройств

Проект

Наши авторы

Авторам

О сайте

Популярное

Справочник программиста

Дроны

LCD генератор символов

Уроки Arduino: сделай сам

Калькулятор Ома

Ардуино Нано с нуля

Выберите плату

Arduino

Orange Pi

Raspberry Pi

Wemos

Важно! Мы используем файлы cookie на сайте, которые позволяют нам предоставлять качественные сервисы. Продолжая использовать наши сервисы, вы соглашаетесь на использование нами файлов cookies.

2017-2018 © Arduino+. Все права защищены. Копирование материалов запрещено, если не указано другого.