

ПЕРСОНАЛЬНЫЙ АССИСТЕНТ ДЛЯ ПЛАТФОРМЫ RASPBERRY PI 2

Э.О. Козловский, М.И. Давидовская

В статье «Персональный ассистент для платформы Raspberry PI 2» рассматривается современное программное и аппаратное обеспечение, позволяющее создавать приложения, использующие голосовое управление для взаимодействия с пользователем. Среди разнообразных методов реализации подобного интерфейса взаимодействия с пользователем выделяются вопрос-ответ ориентированные системы. Представлена архитектура системы для платформы Raspberry PI 2 и реализовано приложение с голосовым управлением.

Введение

С течением времени приложения, выполняющие роль персонального ассистента, становятся незаменимыми и приобретают все большую популярность по мере того, как растет список их возможностей и увеличивается качество распознавания произнесенного текста. Самыми известными представителями персональных ассистентов является программное решение компании «Apple» Siri и компании «Microsoft» - Cortana. Приложения Siri и Cortana спроектированы для мобильных платформ и специализируются на работе с аудио и видео файлами, заметками, списками контактов, веб-страницами.

Мобильные платформы сегодня - это мощные вычислительные ресурсы и ресурсы памяти. Возможно ли решить проблему управления голосом, например, бытовыми приборами, базирующимся на встроенных платформах с ограниченными ресурсами памяти и процессора? Для исследования проблемы был использован одноплатный компьютер Raspberry PI 2 (рисунок 1). Для реализации голосового персонального ассистента были сформулированы задачи:

1. Изучить ОС Raspbian и одноплатный компьютер Raspberry PI 2.
2. Исследовать решения для распознавания речи.
3. Реализовать приложение, предоставляющее информацию о прогнозе погоды, текущее время, проигрывание аудио и видеофайлов, открытие веб-страниц.

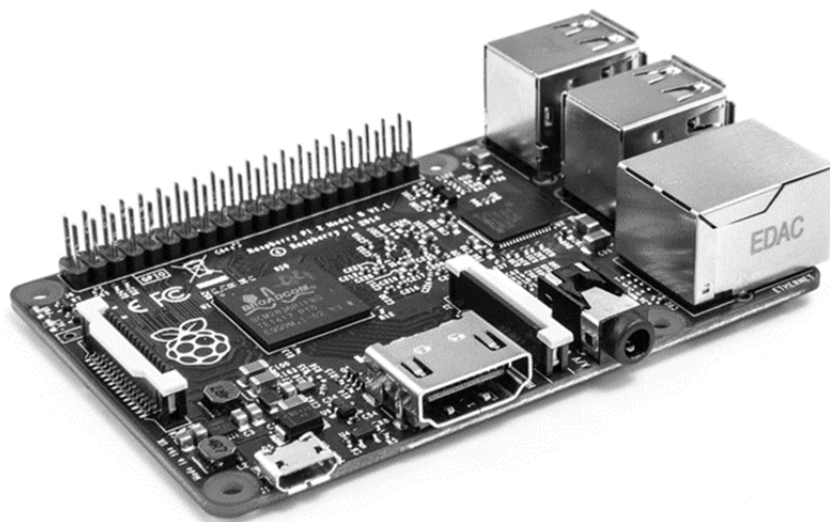


Рис.1. Фото платы Raspberry PI 2

Структура приложения

Для развертывания приложения голосового помощника используем архитектуру, состоящую из трех устройств: одноплатный компьютер Raspberry PI 2, телевизор или монитор и микрофон. Подключенный к Интернет одноплатный компьютер используется для выполнения запросов на удаленные сервисы, телевизор или монитор - для отображения результатов работы, а микрофон - для записи команд пользователя (см. рис. 2).

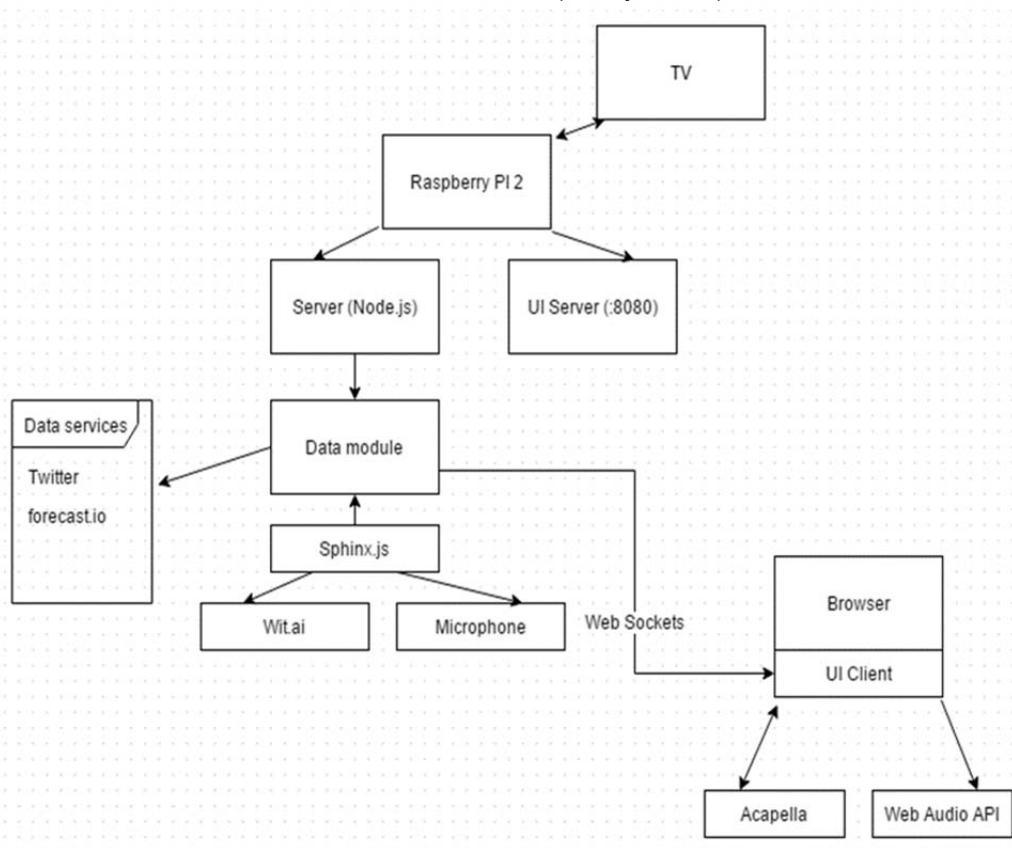


Рис.2. Структура приложение “alipin”

Голосовой помощник, приложение «alipin», состоит из двух частей: сервер, который ждет команд от пользователя, выполняет их обработку и отправку результатов на клиент, и клиент, который ждет сообщений от сервера и их отображает, а также воспроизводит голосовое сообщение - ответ приложения.

Клиент использует сторонний сервис asarella для получения wav-файла ответа приложения и воспроизводит его с помощью Web Audio API. Поэтому поддержка Web Audio API важна для браузера, который будет использоваться для открытия клиента. Клиент показывает результаты команды и на экране, так, например, последние заметки в новостной ленте twitter будут плавно выплывать сверху и отображаться в виде списка. Открытие веб страниц осуществляется в новой вкладке, что предоставляет пользователю абсолютную свободу действия на данном сайте. Открытие и воспроизведение медиа-файлов осуществляется с помощью стандартных средств браузера, а именно путем добавления/изменения тега video.

Основную и самую «тяжелую» работу выполняет сервер. В начале работы он запускает Sphinx.js, оболочку над драйвером PocketSphinx, предоставляющим API для распознавания речи, и ожидает изменений выходного файла обработки звуковой дорожки. Как только этот файл изменен, сервер, с использованием стороннего сервиса wit.ai вычисляет вероятность того, что была произнесена какая-либо команда пользователя и затем решает, выполнять ли ее. Эту функцию выполняет модуль DataManager - главный распределитель работы. В случае выполнения команды сервер передает результаты ее выполнения полученные в ходе работы на клиент и тот их доводит до сведения пользователя. Например, если прозвучал запрос для получения значений погоды, то запрос отправляется на сервис forecast.io. После этого результат воспроизводится клиентом.

Особенности реализации

В реализации приложения “alipin” используется команда непрерывной обработки аудио потока с микрофона: `rocketsphinx_continuous`.

Перед началом работы необходимо скачать, распаковать и собрать драйвер PocketSphinx. После этого его можно использовать.

Самые важные параметры данного метода:

-hup - указание выходного файла - имя файла должно совпадать с тем, который будет отслеживаться в sphinx.js,

-dict - указание словаря,

-hmm - указание акустическую модели,

-lm - указание списка команд.

Стандартная акустическая модель содержится в уже собранном пакете PocketSphinx и нам лишь необходимо указать его. Для приложения “alipin” в первой версии мы используем управление на английском языке, а значит значением данного параметра будет:” /usr/local/share/pocketsphinx/model/en-us/en-us”.

Языковой словарь и файл списка команд генерируются на основе списка предложений

Конечная команда запускающая распознавание речи выглядит следующим образом:

```
pocketsphinx_continuous -hmm /usr/local/share/pocketsphinx/model/en-us/en-us -lm acoustic_model.lm -dict dictionary.dic -hyp out.txt
```

Команда, распознанная сервисом Pocketsphinx проходит по нескольким ступеням: уточнение, когда отправляется запрос на сервис wit.ai, выполнение, когда сервер выполняет необходимую команду и получает данные с использованием сторонних api или внутренних средств, передача клиенту и отображение, и воспроизведение результата.

Приложение, имея такую структуру, обладает рядом преимуществ, таких как масштабируемость, можно легко добавить команду, необходимо лишь определить ее в словаре и реализовать обработчики на клиенте и сервере. Так же приложение обладает портируемостью, так, например, использование движка Pocketsphinx дает возможность использовать данное приложение на любой плате, не обязательно Raspberry PI2, лишь с тем условием, чтобы в браузере была поддержка воспроизведения аудио.

Заключение

Функционал разрабатываемого персонального ассистента позволяет сэкономить время пользователям, а также упростить выполнение ряда команд и может быть использован для голосового управления различными устройствами, включая возможность управления приборами для пользователей с ограниченными возможностями.

В ходе выполнения исследования были выполнены следующие работы:

- изучена ОС Raspbian и одноплатный компьютер Raspberry PI 2;
- проанализирован механизм распознавания речи;
- рассмотрен сервис распознавания речи PocketSphinx;
- спроектирован макет приложения;
- разработано приложение с базовым функционалом.

На данный момент первая рабочая версия персонального ассистента обладает возможностями:

- Воспроизводить информацию о прогнозе погоды в различных городах мира на несколько дней вперед.
- Визуализировать текущее время.
- Отображать список новостей из социальной сети twitter,
- Проигрывать аудио или видео файл, находящийся на носителе.
- Открывать веб-страницы.

Список литературы

1. Крокфорд, Д. Javascript: сильные стороны / Д. Крокфорд. – 1-е изд. – Санкт-Петербург: Питер, 2012. – 174 с.;
2. Node.js в действии / М. Кантелон [и др.]; под общ. ред. А. Сергеева. – 1-е изд. – Санкт-Петербург: Питер, 2014. – 548 с.;
3. Даккет, Д. HTML и CSS. Разработка и создание веб-сайтов / Д. Даккет. – Москва: Эксмо, 2013. - 478с.;
4. Collaboratively developed documentation for the CMU Sphinx speech recognition engines [Electronic resource] / Carnegie Mellon University. USA. – Pittsburgh 2016/ - Mode of access: <http://cmusphinx.sourceforge.net/wiki/>. – Date of access: 06.04.2016

Козловский Эдвард Оярович, студент факультета прикладной математики и информатики Белорусского государственного университета, edward2769940@gmail.com

Давидовская Мария Ивановна, старший преподаватель кафедры технологий программирования факультета прикладной математики и информатики Белорусского государственного университета, davidouskaia@bsu.by