Voice for the Voiceless: Building an Open Source and Affordable TTS Device

By Travis Smith
Professor Vasilios “Billy” Pappademetriou

Presented by Travis Smith

About Travis Smith (Presenter)

- 5th year Co-terminal Graduate Student at IIT
  - Masters in Cyber Forensics and Security
  - Bachelors in Computer Information Systems
  - Minor in Information Security
  - Graduating May 2021
- Undergraduate Research
  - Developed working model for this project.
  - Publishing Paper in progress
  - Currently research project: Using AI for fighting disinformation online
- Undergraduate Research Manager
  - Supervised other Grad/Undergrad students
  - Current Research project: Mesh Networking and alternative communication methods
  - Featured in and syndicated via IIT news
- Platform Engineer at Vail Systems
  - Recognized me from this TTS research
  - Led to Summer internship
  - Offered Full time opportunity
  - Working on developing automation tools for our DevOps pipeline
- Course Developer and Teacher Assistant
  - Developed Intro to Go Programming language class with Professor Billy
  - Co-Taught the class Summer 2020
  - Currently Teacher Assistant for Professor Billy
- President of the HAM Radio and Research Club
  - KD9-NZT
  - Technician License

About Professor Billy

- Professor Vasilios Pappademetriou
  - Known by everyone as “Professor Billy”
- Education
  - Master of Information Technology and Management - Illinois Tech
  - Bachelor of Information Technology and Management - Illinois Tech
- Academic Teaching Experience
  - College of DuPage -- 2015 - Present
    - CIS Adjunct Faculty
  - Illinois Tech -- 2016 - Present
    - ITM Adjunct Industry Associate Professor
    - Undergraduate Research
    - Ham Radio Club Advisor

Vasilios Pappademetriou
Page | 1

VASILIOS PAPPADEMETRIOU

E-Mail: REACHOUTPROFESSORBILLY.COM

AREA(S) OF SPECIALIZATION

Education, Certifications, ITF+, A+, N+, Linux, Security, Operating Systems, System Administration, Forensics

EDUCATION DEGREES

2017	Master of Information Technology and Management Illinois Institute of Technology
2014	Bachelor of Information Technology and Management Cum Laude, Illinois Institute of Technology
2009	Associate of Applied Science, Computer Networking and Support, Triton College

EDUCATION CERTIFICATES

2009	College of DuPage, Computer Forensics
2005	Triton College, PC End User Support Specialist
2005	Triton College, Wireless Network Management – CWNA
2005	Triton College, Internet and Network Security Management
2005	Triton College, Advanced Computer Maintenance

ACADEMIC TEACHING EXPERIENCE

2016 – Present	**Adjunct Industry Associate Professor**, Illinois Institute of Technology, Chicago, Illinois https://appliedtech.iit.edu/people/vasilios-billy-pappademetriou
2018 – Present 2015 – 2016	**Adjunct Faculty**, College of DuPage CIS Department, Glen Ellyn, Illinois
2017	**Lecturer Faculty**, College of DuPage CIS Department, Glen Ellyn, Illinois
2017 – 2018	**Instructor**, College of DuPage Continuing Education Department, Glen Ellyn, Illinois
2018 – 2019	**Adjunct Instructor**, Triton College CIS Department, River Grove, Illinois

Click on image to open CV

About the Research

- Conducted in an Independent Research class
  - ITM department (ITM-497)
  - Spring 2019
  - Co-Authored and supervised by Professor Billy
- Focused on bringing TTS technology to the disadvantaged
  - Device is a proof of concept
  - Accomplished via using open source components and software
  - Using low-cost and readily available hardware
  - Writing the software from scratch
  - Designing the device to work offline
  - Goal of emulating the user experience of existing expensive TTS alternatives
- Presentation will be regarding
  - Background of Speech-Generating Devices
  - Our TTS Device design decisions and methodologies
  - Overview of the hardware/software
  - Plans for the future

Background: Speech Generating Devices

- What are they?
  - Any device that assists a person to verbally communicate
    - Loss of hearing, speech, or voice
- What kinds of SGD devices exist?
  - Assistive Listening Devices
  - Augmentative and Alternative Communication Devices (AAC)
    - What our TTS device is classified as
  - Alerting Devices
- What types of AAC devices exist?
  - Unaided
  - Low Tech
  - High Tech

Problems with Commercial AAC Devices

- The cost
  - Non-dedicated vs Dedicated devices
    - Dedicated device shown to the right costs $6000 USD
    - Non-dedicated IPad with similar software costs < $1000
    - Our open source AAC device < $150
- Lack of updates and features
  - Closed source software
  - At the mercy of developers
    - Application updates
    - Bug fixes
    - Supporting newer device hardware and OS versions
- Upgradability and maintainability
  - Pay manufacturer
  - Replace the device
- Internet connection requirement
  - User cannot afford
  - Not available to the user

Design Requirements for our AAC System

- Low cost
  - Half the cost of the cheapest iPad (without software)
  - Includes hardware and software
- Portable
  - Cannot be much larger than a tablet
- Configurable
  - Created using off the shelf components
  - Easily upgradable and maintainable
  - Customizable to each user
- Open Source
  - Customizable and maintainable by the AAC community
  - Avoids software abandonment
- Function without an internet connection
  - Improves access
  - Limits speech synthesis options
    - Could be added as an optional feature

Hardware Design

Main Compute Device

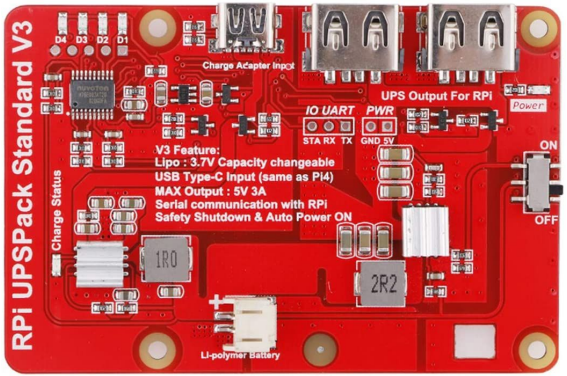
- Raspberry Pi 3 A+
  - Small fully capable computer
  - Power and Portability
- Specifications
  - 64 bit quad-core 1.4GHz Arm Processor
  - 512MB Memory
  - AC Wireless
  - Bluetooth
  - Audio/Video output
- Price
  - $30 USD
- Why was it chosen?

Touchscreen

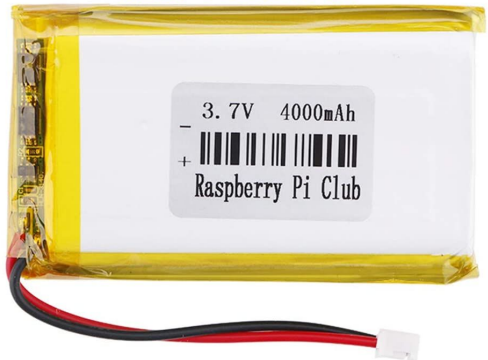
- Official Raspberry Pi 7” Touchscreen Display
- Specifications
  - 800 x 480 Resolution
  - 10 Finger Capacitive Touch
  - Powered and driven by conversion board
  - Easy installation
- Price
  - $30 USD
- Why was it chosen?

Power Solution

- 3.7V 4000mAH battery pack
  - USB conversion board
  - Connects to Raspberry Pi via conversion board
- Price
  - $24 USD
- Why was it chosen?



NEW Version V3



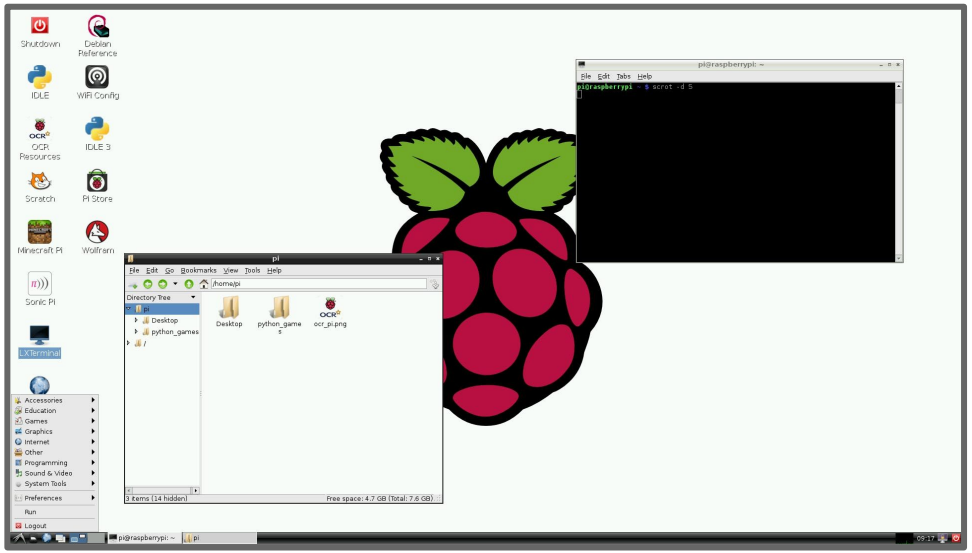
Case

- SmartPi Touch 2
- Bulky but durable case
- Comes with removable stand
- Price
  - $30 USD
- Why was it chosen?

Software Design

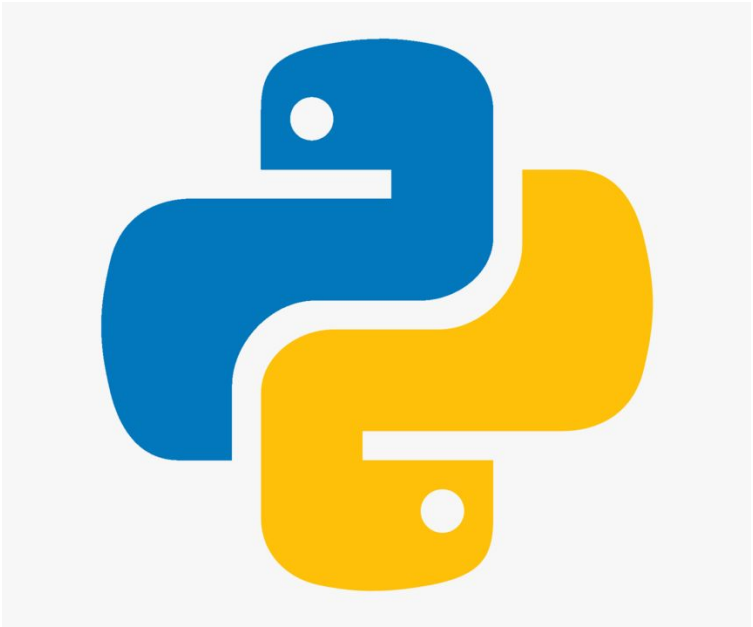
Software Design: Operating System Choice

- Destined to be a flavour of Linux
- Raspbian OS
  - Debian-based
  - Officially provided and supported by the Raspberry Pi foundation
  - Licensed using free and open source software licences
  - Comes with tools and driver's pre-installed
    - Python
- Other considerations
  - Ubuntu
  - Centos
  - TinyCore



Software Design: Programming Language

- Language was chosen with these considerations
  - Offline TTS libraries
  - GUI libraries
  - Ease of use and popular with Raspberry Pi
    - Maximize contributions and availability
- Python met these requirements
  - Pyttsx3 offline TTS
  - Multiple great GUI libraries
    - Tkinter
    - PyQT
    - Wx-Python
  - Arguably most beginner friendly language
  - Most popular for Raspberry Pi projects
- Comes pre-installed with Raspbian



Software Design: Text to Speech Engines

- Offline TTS Engines
  - TTS software is hosted locally on client device
  - Speech synthesis computed locally
  - No internet connection required
- Pro:
  - Self sustaining
    - Not reliant on internet provider or speech synthesis service
  - No subscription required
  - Ideal for our TTS use-case
- Cons:
  - Voice options can be limited
  - Manual updates to engine
  - Limited by power of local device

- Online TTS Engines
  - Text is sent over the internet to TTS service
  - Service uses powerful backend infrastructure to perform the speech synthesis
  - Sends audio result back to client
- Pro:
  - Synthesized voice sounds more human
  - More voice options
    - Age, Sex, Ethnicity
    - Custom voices
  - Not limited by compute power of client device
- Con:
  - Requires an internet connection
  - Some require a subscription
  - Service only possible if provider is online

Software Design: Text to Speech Engine Choice

- Pyttsx3
  - Works completely offline
  - Uses OS included TTS engines
    - Sapi5
      - Windows
    - Nsss
      - MacOS
    - Espeak
      - Linux
  - Voice customization
    - Speed/Rate
    - Volume
    - Voice type
  - Easy and intuitive to use

Software Design: Pyttsx3

- Raspbian
  - Espeak TTS engine comes pre-installed
  - For debian based distributions without Espeak, it can be installed easily using the apt package manager
- The example to the right showcases the different voice configurations
- For our TTS project, pyttsx3 was used with default configurations for Espeak
- The GUI buttons call pyttsx3 speak methods to perform the speech synthesis on the fly
  - Saving to a file was not used

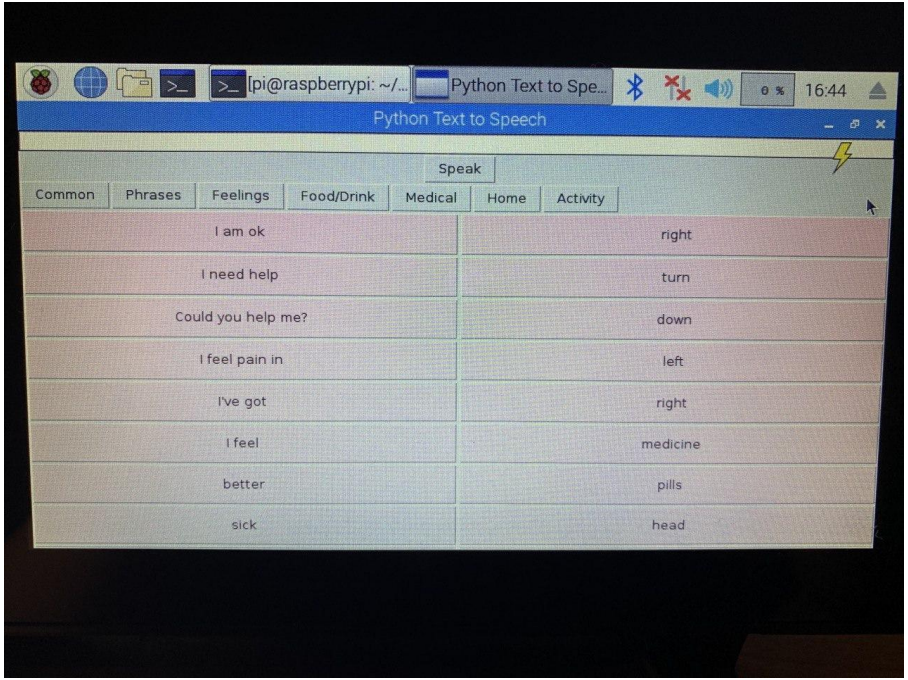
```
import pyttsx3
engine = pyttsx3.init() # object creation

""" RATE"""
rate = engine.getProperty('rate')   # getting details of current speaking rate
print (rate)                        #printing current voice rate
engine.setProperty('rate', 125)     # setting up new voice rate


"""VOLUME"""
volume = engine.getProperty('volume')   #getting to know current volume level (min=0 and max=1)
print (volume)                          #printing current volume level
engine.setProperty('volume',1.0)    # setting up volume level  between 0 and 1

"""VOICE"""
voices = engine.getProperty('voices')       #getting details of current voice
#engine.setProperty('voice', voices[0].id)  #changing index, changes voices. o for male
engine.setProperty('voice', voices[1].id)   #changing index, changes voices. 1 for female

engine.say("Hello World!")
engine.say('My current speaking rate is ' + str(rate))
engine.runAndWait()
engine.stop()


"""Saving Voice to a file"""
# On linux make sure that 'espeak' and 'ffmpeg' are installed
engine.save_to_file('Hello World', 'test.mp3')
engine.runAndWait()
```

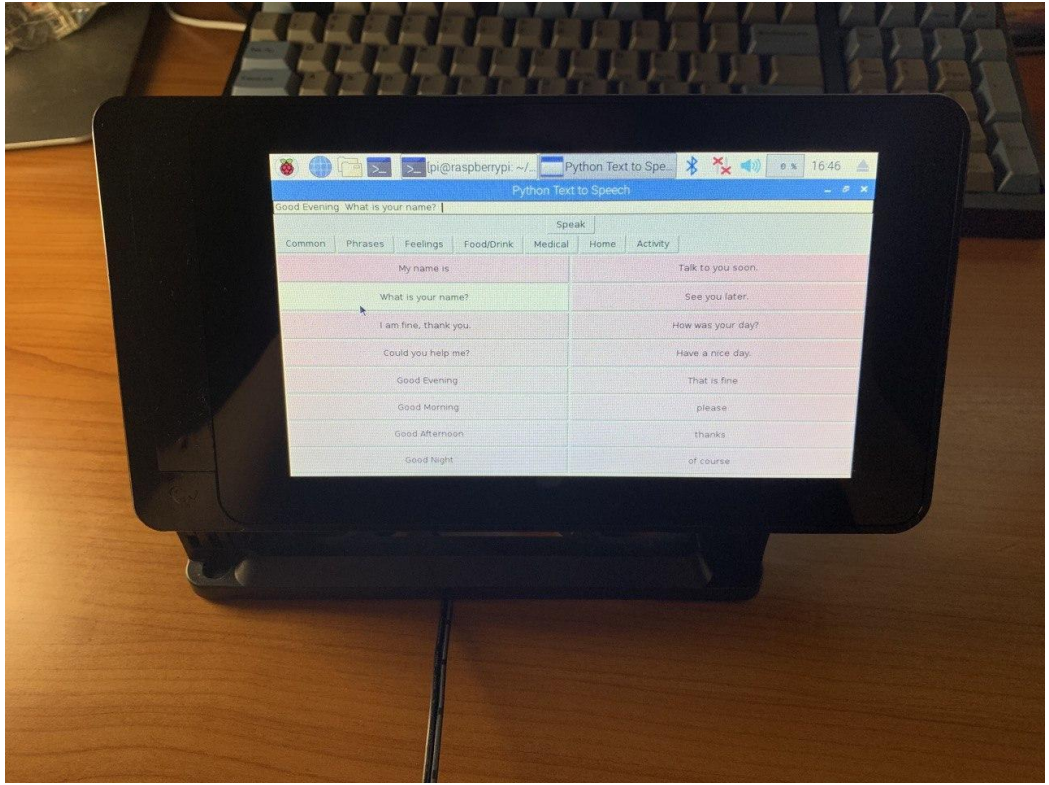
Software Design: Graphical User Interface

- Design considerations
  - Computational cost
    - Raspberry Pi limitations
      - CPU/Memory
      - Battery life
  - Simple and intuitive for AAC applications
    - Don’t want to confused users
    - No need for advanced menus and options
    - Touchscreen friendly
- GUI libraries examined
  - PyQt
  - Tkinter
  - wx-Python
  - PyGTK

Software Design: Graphical User Interface Choice

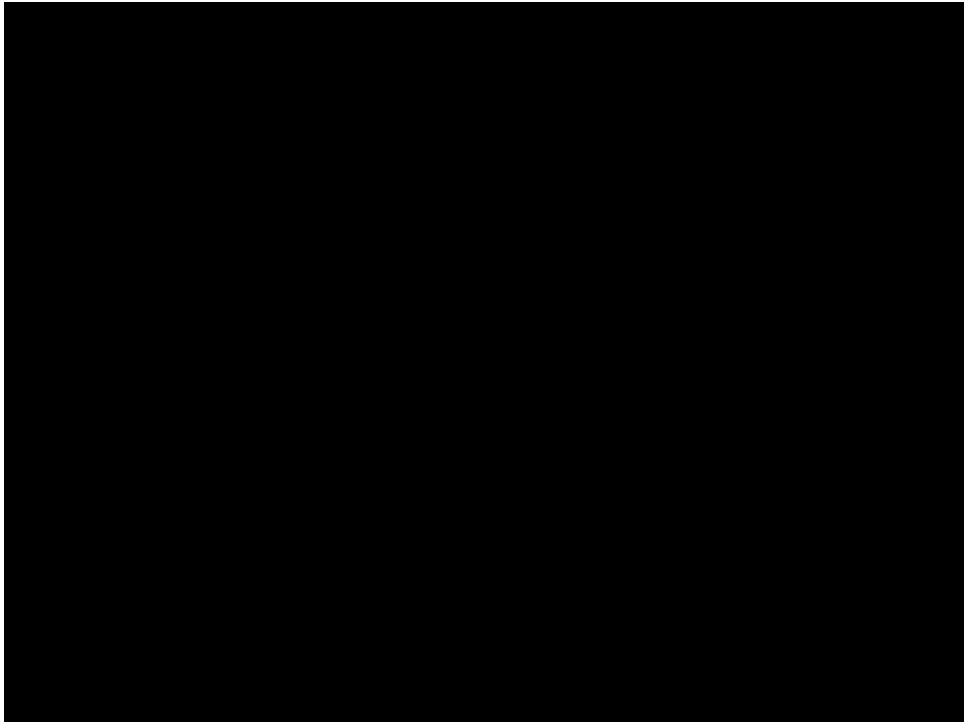
- Tkinter
  - Ideal for simple designs and applications
  - Basic functionality
    - Tabs
    - Buttons
    - Screens
  - Our AAC application had a few different components
    - Sentence builder
    - Speak button
    - Speech Category tabs and screens
      - Word buttons that populate sentence builder
    - Resizable
  - Tkinter allowed these to be built and displayed simply





Front View of the Device

Video Demonstration



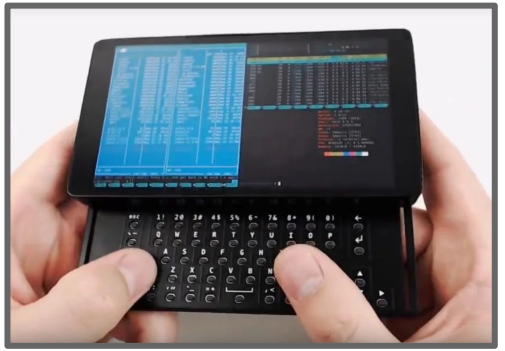
What can be improved?

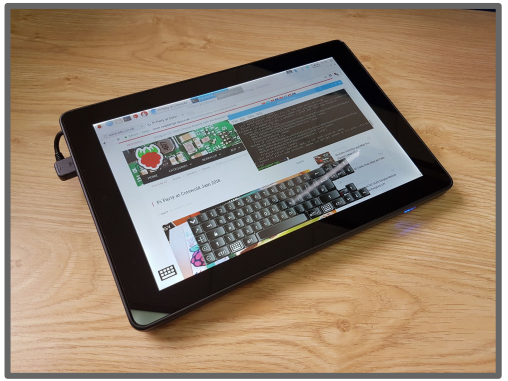
- Restrictions on the device mainly come from the two requirements
  - Off the shelf hardware components
  - Offline functionality
- Off the shelf hardware
  - Readily available, but not as customizable
    - Shape, size, color, etc
  - The case is the main target here
    - 3D printing alternative
      - Cost less
      - Reduce size of device
    - Could be more difficult to mass produce

- Offline functionality
  - Improves access to the device
  - Vasly decreases TTS synthesis quality and options
  - Alternative would be a togglable engine option
    - User can choose online or offline option depending on needs
    - Provide users with access to internet an experience closer to more expensive AAC devices
- Raspberry Pi's are very customizable
  - Our implementation is a proof of concept
    - Showcases that medical devices can be emulated with cheaper, DIY alternatives to increase access
  - Different screens, cases, and software can be used to achieve desired target and results

Future Plans

- Hardware experimentation
  - Make the device more compact
    - Raspberry Pi zero
    - Smaller touch-screens
    - Custom cases
- Software expansion
  - User customizability
  - Offline and online functionality
    - Multiple TTS engines
  - GUI improvements
    - Images and words on the buttons
    - Users add their own buttons
- Examples on the right showcase how customizable Raspberry Pi's can be
  - Not a one size fits all solution
  - Exciting for the field of AAC devices





Thank you!