Group 1.9 - IoT Voice Recognition using MinnowBoard Turbot

Industry Support: Intel

MinnowBoard

The MinnowBoard is a MiniPC that features an x86 architecture, the Specs are:

- Dual Core Intel® Atom™ (2 x 1.66GHz, 1MB cache)
- 2GB DDR3 on board
- 1x Micro SD Slot, 1x USB 2.0 host, 1x USB 3.0 host, 8x buffered GPIO, I2C, SPI, IGB Ethernet
- Supports Linux (Debian, Fedora, Yocto), Windows (10 IoT, 8.1), Android 4.4
- Open Source Hardware
- Low Cost and Small Form Factor

The components circled in red are the ones relevant to our project:

- Switch
- Micro-HDMI
- Gigabit Ethernet Port
- SD Card SDIO
- USB 2 (Upper)
- USB 3 (Lower)

ARD-Audio

Ard-Audio

- 16K ADC
- Stereo Input
- Stereo Output
- Optional SMT mix
- Optional pins for Speaker

Connections & Add-Ons

- GPIO Pin Layout
  - I2C/SCI: clock for I2C data
  - I2C SDA: data sent across I2C
  - I2C SCL: data sent across I2C
  - LRCLK: clock to choose left and right channel
  - BCLK: bit clock
  - MCLK: master clock to synchronize LRCLK and MCLK

Voice Assistants APIs

Amazon

- Native API written in Node.JS
- We are using a Python Wrapper
- Flash-Ask, which
- Voice commands are called "Skills"

Google

- Native API written in Java
- We are going to use a python wrapper called Flash-Assistant
- Voice commands are called "Tasks"

Software Application

- MinnowBoard needs 5V DC in order to operate
- ARD-Audio Board needs 3.3V DC in order to operate

Project Constraints

- Implementing a voice recognition solution using either Amazon Alexa, or Google Home
- Using the DA7212 Drivers to get the Ard Audio to drive input and output

Linux, Drivers, Libraries

Linux Distro

- Fedora 25

Drivers

- DA7212 Ard Audio

Libraries

- ALSA (Advanced Linux Sound Architecture)
- Flash
- Flash-Ask (Amazon Skills)
- Flash-Assistant (Google Tasks)

Testing and Verification

- GPIO (I2S, I2C) connections
- Ensure Drivers are loading properly
- USB and Line In microphones
- Speaker capability on the add on board
- API Communication latency
- API Buffer length testing
- Compare performance to Amazon and Google IoT devices
- Benchmarking Application performance

2nd Semester Goals

- Integration with flash-assistant (Google)
- More custom Skills and Tasks orientated for technical applications
- Optimizing the application stack
- Investigating the DSP modules on the SoC
- Implement the Testing and Verification goals
- Documentation and tutorial

Conclusion

Our project has provided us ample opportunities to learn about various topics such as IoT, OpenSource Hardware/Software, Linux Kernel Development, and working as a cohesive team. This If you have any question, ideas, or concerns please talk to one of our three team members as we are more than eager to clarify the specs.

References

- https://wiki.minnowboard.org/images/0/06/MB-Turbot-AD9-9806-150907.png

Acknowledgments

Texas State
- Dr. Stapleton
- Dr. Compeau
- Dr. Londa
- Dr. Aslan

Intel
- John Zeiter
- Dr. Bossart

Areas of Interest

- Voice Assistants API (Alexa skills, Google Tasks)
- DSP
- IoT
- Linux OS (Kernel, Drivers, and Libraries)
- BASH and Linux Command Line
- Security
- Computer Networking
- Exposure to Linux Kernel and Driver development
- Researching ADC's and Audio input/output
- Determined the Ard-Audio board as our ADC for voice input
- Determined which libraries we are using
- Set up a SSH server for remote administration

1st Semester Work

Challenges

- Voice recognition application
  - Google cloud speech has a maximum free usage
  - The python wrapper for Google's cloud speech is in Alpha
  - Amazon prevents the use of a wake up command, thus we have to develop a solution around this.
- Ard-Audio Board
  - We need to provide the correct voltage for I2C and I2S
  - The connection of the Ard-Audio to the MinnowBoard does not match up
  - Soldering an on board microphone needs to be surface mounted
- Security
  - With a microphone we want to make sure we are not always recording Audio, so that the users privacy is protected
  - We also want to make sure usernames and other sensitive material are not passed unencrypted

Importance

Current implementations of Voice Recognition devices pursue alternative low powered architectures. The project we are developing uses an x86 SoC which allows use to investigate the performance of this device and to tailor a solution around the Amazon Alexa and Google Home APIs.