TiresiaScope
Winter Quarter Design Review

DEVON PORCHER, JOHN BOWMAN, BRIAN YOUNG, TIMOTHY KWONG, TREVOR HECHT
Development Team

- Devon Porcher: Team Leader, Prototyping, Software Design
- John Bowman: System Design Lead, Software Design
- Brian Young: PCB Design Lead, System Design
- Timothy Kwong: Software Design Lead
- Trevor Hecht: Apparatus Design Lead
Introduction - What is the TiresiaScope?

• A proximity-sensing device for the blind
• Detects nearby objects with ranging sensors, recognizes nearby faces and/or objects with a camera
• Relays information to user through sound: musical tones for object location and distance, synthesized speech to notify of nearby people
Block Diagram
PYNQ

- Dual-Cortex ARM Cortex A9 processor supports coding in Python
- Individual Microblaze processors on FPGA control I/O for arduino and PMOD headers
- Microblazes communicate with processor using shared memory
- HDMI, USB, Ethernet also supported
- Why PYNQ?
  - Possibility: taking advantage of the ARM
  - Processor for image recognition, etc.
- Audio out is mono only
Camera: OpenMV M7

- On board STM32F765VI ARM Cortex M7 processor
- 640x480 8-bit grayscale images or 320x240 16-bit RGB565 images at 30 FPS
- Built-in face detection on its own processor
Ultrasonic Sensor:
Ultrasonic Range Finder - LV-MaxSonar-EZ1

- Detection range: 160mm to 6.45m
- 20-Hz refresh rate
- Reliable and stable range data
- Pulse-Width, Analog, Pseudo-UART Interface options
- Using UART
- Operates at 5V
Optical Sensor:
Simblee™ IoT 3D ToF Sensor Module

- Detection range: 100 mm to 2 meters
- 10-Hz refresh rate
- Breakout Board for mounting
- I2C interface
- Operates at 3.3V
Audio Codec PROTO: WM8731

- Stereo audio output (and input)
- SPI or I2C control interface
- I2S, left-justified or right-justified formats for audio interface
- Used commonly in digital TVs
PCB Schematic
PCB Layout
Power Distribution

• The PYNQ simplifies our power system considerably
• One power supply into the PYNQ, which itself provides access to 3.3V for our peripherals
• PYNQ accepts 7-15 V DC
• A simple rechargeable battery supply to the PYNQ should function well and be portable
Software

Python is used for the backend processing

Sensors

• Converts sensor value inputs into noise frequency outputs of a certain tone depending on the range

• Uses multithreading in order to have each sensor read and output values independently

Camera

• Will capture images. Current plan is to use built-in facial detection, or to go further if time allows
Implementation

• Reading sensor values as ascii characters from the microblaze that get driven into Python script
• Using input along with static parameters to determine which sensor position is reading, convert value into frequency for the audio codec
• Output value into audio codec through microblaze to play a frequency
Wearable Apparatus

Current plan:

• Skateboard helmet, with sections removed to make space for mounting

Mounting:

• PYNQ set into top of the helmet
• Camera at front
• Sensors distributed around all sides
Progress This Quarter

• UART sensor working with PYNQ
• I2C sensor working with PYNQ
• Audio codec working with Arduino
• PCB just arrived
• Advancement on Python code
  • Each direction now produces one of five tones
  • Sound plays simultaneously, with information designated by direction
Conclusion

Moving Forwards:
• Prototyping multiple sensor system
• Sensors detecting lower objects
• Camera functionality: facial detection, object detection
• Designing software to function with the sound system

Thank you to professor Yogananda Isukapalli, Celeste Bean, and Caio Motta

Questions?