TiresiaScope

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

- Devon Porcher: Team Leader, Prototyping
- John Bowman: System Design Lead, Software Design
- Brian Young: PCB Design Lead, System Design
- Timothy Kwong: Software Design Lead
- Trevor Hecht: Apparatus Design Lead
Mission

• As technology advances, so does society’s ability to help those that have physical disabilities.

• TiresiaScope’s objective is to help the blind by creating a headset that uses sounds to assist in the navigation of their surroundings.
Motivation

• To begin, we wanted to make an improvement upon the tools already available to people who are blind—namely, the cane.

• We decided to make a device which could inform the user of nearby objects, without having to physically touch them.

• We also wished to possibly provide some sort of feedback as to what those objects are
What is the TiresiaScope?

• A proximity-sensing device for the blind
• Detects nearby objects with ranging sensors, detects nearby faces with a camera
• Relays information to user through sound: musical tones for object distance, surround sound to simulate object location, and a beep 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
- Why did we choose PYNQ?
  - Power for computer vision
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, simple to program
Ultrasonic Sensor:
Ultrasonic Range Finder - LV-MaxSonar-EZ1

• Detection range: 160mm to 6.45m
  • Our long-range sensor
• Reliable and stable range data
• Multiple interface options
  • We use UART
Optical Sensor:
Simblee™ IoT 3D ToF Sensor Module

- Detection range: 100 mm to 2 meters
  - Our backup sensor, more reliable in shorter distances
- I2C interface
Audio Codec PROTO:
WM8731

- Stereo audio output (and input)
- SPI data interface, I2C control interface
- Used commonly in digital TVs, other applications
- Why an audio codec?
  - PYNQ has *mono* output.
PCB Layout
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

Camera

• Captures images. Uses built-in facial detection.
System Flow

- The PYNQ periodically polls the array of sensors for range data, polls camera for face detection
- Calculate range bins for closest objects
- Forms a sound sample from calculated ranges
- Pushes the sample through to the audio codec to play for the user
Wearable Apparatus

• Skateboard helmet, that is sturdy enough to mount sensors and camera

Mounting:

• PYNQ set on top of the helmet
• Camera at front
• Sensors distributed around the front
Conclusion

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

Questions?