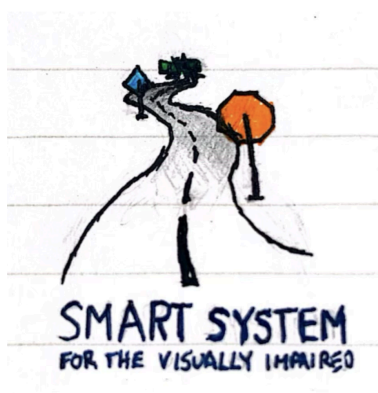
Boston University
Electrical & Computer Engineering

EC463 Senior Design Project

First Prototype Testing Plan
Smart System for Visually Impaired



by

Team 29
Smart Bears

Team Members

Lukas Chin	lchin10@bu.edu
Jake Lee	jaketlee@bu.edu
Shamir Legaspi	slegaspi@bu.edu
Jason Li	jli3469@bu.edu
David Li	dav@bu.edu

Required Materials

Hardware Related:

- 10mm Vibrating Mini Motor Disc (2V-5V) (11000 RPM)
- 7x25mm Micro Vibrating Motor (1-6V) (8000-20000 RPM)
- TinyPico ESP32 Microcontroller Board
- Raspberry Pi Camera Module v2 8-megapixel
- Raspberry Pi Zero W
- Router

Software Related:

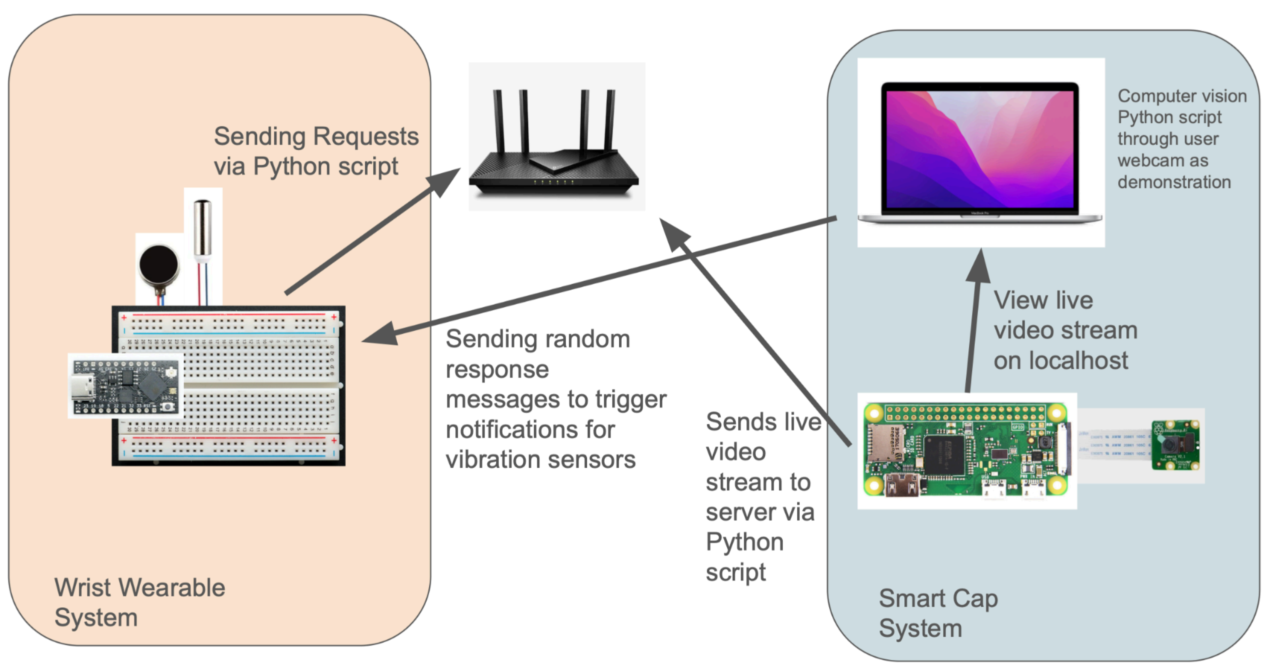
- Python3 Scripts:
    - Trained computer vision model
        - Able to detect objects with considerable accuracy
        - Able to trigger based on desired target object
    - Router web socket to microcontroller and Raspberry Pi
        - Transmits UDP traffic from microcontroller to Raspberry Pi to control vibration motors from router
    - Vibration motor control
        - Different voltage sent for varying intensities based on ‘notification’
            - These notifications are function placeholders for object sensing, etc.

Set Up

Our current prototype involves successful demonstration of the key triggers and feedback of the "Smart System for Visually Impaired" that is varying haptic feedback, computer vision assistance, and wifi and server connection from both systems in the pair.

The smart cap system in the pair utilizes a python-powered computer vision script through the user's webcam as a prototype and can accurately determine basic street obstacles in a photograph. This will later be scaled to work on our Raspberry Pi Camera Module v2 and connected to a cloud where the computer vision processing will occur. The Raspberry Pi Zero W which will eventually power this smart cap system will connect with the Raspberry Pi Camera Module v2, and for our prototype, a demonstration of live video feedback from the Raspberry Pi Camera Module v2 connected with the Raspberry Pi Zero W will be showcased.

The wrist wearable system in the pair currently utilizes two separate vibration motors, one with on/off vibration functionality and the other with varying vibration based on voltage output from the GPIO pins on the TinyPico microcontroller. The on/off vibration motor operates on 5V while the varying vibration motor operates on a 1-5V range. This voltage range is what gives it its varying vibration. A python script on the microcontroller will showcase the two separate vibration motors turning on and off and also increasing/decreasing in intensity based on certain "notifications". These notifications will later be replaced with feedback from the smart cap in an intuitive way to allow the user to receive appropriate feedback based on their surroundings.



Pre-Testing Setup Procedure:

1. Ensure the router is plugged in and operating.
2. Check Raspberry Pi and TinyPico's connectivity to the router via FreshTomato control panel.
3. Ensure the user's computer is connected to the same network as the Raspberry Pi and TinyPico.

Testing Procedure

To test smart cap computer vision capabilities:

1. Run python script to enable computer vision through user's computer webcam/phone camera
2. Place basic objects in front of camera to visualize script capturing the webcam contents appropriately
3. The accuracy of the object detection is measured in real-time, observe the accuracy % as well as the real-time feedback from the user's terminal
4. Pictures can also be uploaded to test for object detection of basic obstacles (cars on a street for example)

To test Raspberry Pi Zero W camera feedback:

1. Have pre-setup router plugged in to connect Raspberry Pi Zero W
2. Run python script to enable live camera streaming from Raspberry Pi Camera Module v2 with the Raspberry Pi Zero W
3. Observe the live video streaming on localhost with user's computer also connected to the same network as the Raspberry Pi Zero W (the router)

To test wrist-wearable:

1. Have pre-setup router plugged in to connect TinyPico microcontroller and server device (computer)
2. Run server python script on server device to enable response generation
3. Run python script on microcontroller to enable vibration motor testing
4. The TinyPico microcontroller will send a request to the server via WiFi when connected to the router
5. Server will send a randomly generated response received by TinyPico
6. This random response will determine the "signal" emitted by the vibration motors
7. Showcase three distinct vibration patterns to indicate different event triggers:
    a. Three pulses (indicate the presence of an obstacle or item nearby)

b. Gradual increase in vibration intensity (indicates the user is getting closer to an object or target)
c. Constant vibration intensity (signals a stationary object or a consistent action required)

To test full connectivity

1. Have pre-setup router plugged in to connect TinyPico microcontroller, Raspberry Pi Zero W, and server device (computer)
2. Run camera_stream_server.py on the computer to initialize the computer vision processing server
3. Run camera_stream_client.py on Raspberry Pi W to send video stream to server for processing via TCP connection
4. Run udp.py script on TinyPico microcontroller to listen for incoming packets
5. Pan camera around the room to showcase voice callouts on detected objects
6. Aim the camera toward the target object (person) to showcase connectivity with the haptic motors to alert the user

<u>Measurable Criteria</u>

The criteria that indicate success are as follows:

I. The Raspberry Pi should be able to capture and stream video, viewable by devices on the local network.
II. The computer vision model should be able to detect relevant objects in input images at a rate of 80% success.
III. On the breadboard that the TinyPico is connected to, there is a buzzer. The TinyPico should be able to communicate with a server on the network and signal different vibrations through the buzzer based on the response received from the server. Users should be able to differentiate between the different vibration patterns 90% of the time.

<u>Score Sheet</u>

Tasks	**Successful? (Y/N)**
Raspberry Pi and TinyPico are connected to router's network	Y
Raspberry Pi's camera module's live stream can be observed on user's computer	Y
TinyPico successfully sends request and receives responses	Y

TinyPico's received responses trigger "notifications" which prompt vibration motors to turn on	Y
Computer vision Python script through user's computer's webcam successfully detects objects and reports accuracy of detection	Y
Able to communicate audibly to the user the detected objects	Y
User is able to sense a vibration when a target object is in frame	Y
User feedback in under 1 second	N
Result →	**100%**

Prototype Results

The initial prototype testing of the "Smart System for Visually Impaired" demonstrated success across all evaluation criteria. The hardware integration proved flawless, with both the Raspberry Pi Zero W and TinyPico microcontroller establishing stable network connections through the router. The computer vision component, powered by Python scripts, successfully achieved its target detection rate, while the Raspberry Pi Camera Module v2 effectively delivered real-time video streaming capabilities. The system's core functionality included accurate object detection through the computer vision model and seamless communication between components.

The haptic feedback system, built around two distinct vibration motors, performed well in user testing. The system successfully implemented three distinct vibration patterns: triple pulse for obstacle detection, gradual intensity increase for proximity alerts, and constant vibration for stationary objects. The TinyPico microcontroller effectively managed voltage control, ranging from 1-5V, to provide varying vibration intensities based on different triggers. The comprehensive testing validated all components' functionality, achieving a perfect 100% success rate across all evaluation criteria, including network connectivity, video streaming, server communication, and haptic feedback response.