Sensory Feedback System with Raspberry Pi Pico W

Course Name: Human Computer Interaction

Professor: Dr. Yingcai Xiao

Final Project

1. **Introduction:**

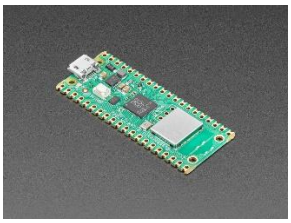
   Background Knowledge:

   Human-Computer Interaction (HCI) focuses on designing systems that create intuitive and effective interactions between users and technology. Sensory feedback, such as audio, visual, or touch responses, plays a vital role in HCI by enhancing the way users experience and engage with devices. Sensory feedback can provide users with immediate, context-specific information, guiding actions, creating immersive experiences, and making technology more accessible. System that integrates sensory feedback help bridge the gap between digital processes and human perception, allowing users to feel connected to digital interactions in more natural and intuitive manner.

   Project Overview:

   This project was designed as an interactive sensory feedback system utilizing a Raspberry Pi Pico W, combined with an ultrasonic sensor, a buzzer, an RGB LED, and an MRP 121 touch sensor module. The goal was to create a real time feedback system where different sensory inputs -distance and touch- trigger specific responses in sound and color. When a hand is detected within specific distance ranges from the ultrasonic sensor, a passive buzzer emits sounds of varying frequencies which tells how far the hand from the sensor is and the RGB LED lights up in random colors, giving users both auditory and visual feedback. The addition of the MPR121 touch sensor extends the project by providing touch-based sensory feedback. This project emphasizes sensory feedback to create an multisensory interaction.

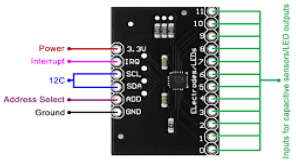
2. Design and Components



Raspberry Pi Pico W: For processing inputs and controlling outputs

Ultrasonic Sensor: To measure the distance between the sensor and an object

Passive Buzzer: Provides auditory feedback through variable frequencies based on distance



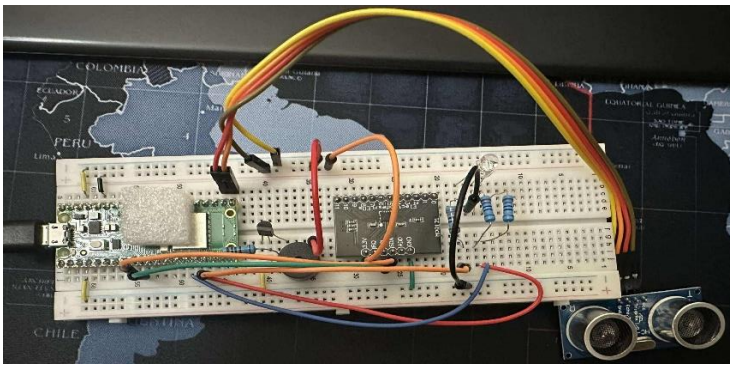
MPR121 Module: To control the different touch sensation

- RGB LED:
  Adds visual feedback by providing random colors when the hands is detected within a certain range
- Transistor: To increase the electrical output
- Resistors: To control the flow of current

System Design:

The project's circuit centres on a Raspberry Pi Pico W microcontroller, which collects data from an ultrasonic sensor for distance measurement, a passive buzzer for sound output, an RGB LED for color feedback, and a MPR 121 module to enhance interactivity.

Wiring Connections



1. Ultrasonic Sensor:
   Trigger Pin (GPIO 17) sends a pulse, and Echo Pin (GPIO 16) receive the reflection to measure distance.
   Power: Connected to the Pico W's 3.3V and GND.
2. Passive Buzzer:
   PWM PIN: Outputs carrying sounds frequencies based on the measured distance.
   Power: Connected to Pico W's GPIO 15 and ground.
3. RGB LED (GPIO 13,12,11):
   Pins control red, green, and blue channels using PWM for random color feedback.

Power: Each color channel is grounded through the Pico W.

4. MPR121 Module (12C):
   12C pins (SDA, SCL) allows communication with the Pico W, while VCC and GND connects to the 3.3V and ground on the Pico.

Implementation:

```
# Function to measure distance using the ultrasonic sensor
def distance():
    TRIG.low()
    time.sleep_us(2)
    TRIG.high()
    time.sleep_us(10)
    TRIG.low()
    while not ECHO.value():
        pass
    time1 = time.ticks_us()
    while ECHO.value():
        pass
    time2 = time.ticks_us()
    duration = time.ticks_diff(time2, time1)
    return duration * 340 / 2 / 10000  # Return distance in cm
```

```
# Function to play a tone on the buzzer at a specified frequency
def tone(frequency):
    buzzer.freq(frequency)  # Set buzzer frequency
    buzzer.duty_u16(30000)  # Set duty cycle to 50% (approx)
```

The distance measurement function calculates the time it takes for sound to travel to an object and back, converting it into centimetres. Frequency mapping was implemented to change the pitch of the buzzer according to specified distance ranges, creating an intuitive, auditory representation of distance.

```
# Function to randomly light up the RGB LED with random color values
def lightup():
    red.duty_u16(int(urandom.uniform(0, 65535)))  # Set random intensity for red
    green.duty_u16(int(urandom.uniform(0, 65535)))  # Set random intensity for green
    blue.duty_u16(int(urandom.uniform(0, 65535)))  # Set random intensity for blue
```

The lightup function calculate the random color to lightup the RGB LED.

Lessons Learned:

This project strengthened my skills in programming and electronic. I learned to generate specific tones through the buzzer and dynamically adjust the RGB LED Colors. I was able to read the ultrasonic sensor data and convert it into the correct metric for understanding(cm). I had problem in step of the different component in breadboard like RGB light as I had to use different resistors to get the right power to the bulb. Combining the code for both RGB and ultrasonic sensor was little time consuming for me.

Possible Future Work:

For future enhancements, integrating a small display could provide direct distance readouts, making feedback clearer for users. Adding more refined tone variation or color patterns tied to specific ranges could also help users interpret proximity more intuitively. Additionally, incorporating wireless connectivity through Pico W's Wi-Fi application could help to monitor the device remotely, which will open up a new possibility for real world applications like interactive installations.