Name: Dharmik Shah **#: N01581796**

Hardware Report for VCNL4040 Proximity Sensor Integration

1.0 Introduction

This project uses the Raspberry Pi Broadcom development platform, which previously supported multiple environmental sensors such as CO_2, temperature, and humidity sensors, sending real-time data to a Firebase backend. The addition of the VCNL4040 proximity sensor enhances the system by enabling motion detection near the phytoplankton air purification device. This allows interactive features within the companion Android app, enriching user engagement through visual effects and notifications.

2.0 Added Functionality

By integrating the VCNL4040 proximity sensor, the system detects when a user approaches the device. The Raspberry Pi reads sensor data and sends proximity information to Firebase over Wi-Fi. The Android app listens for these proximity updates to trigger gamified animations, such as water ripple effects and interactive notifications thanking users for their presence. This extends the project's interactivity and improves user experience.

2.1 Sensor/Effector Purchase

- The proximity sensor used is the **VCNL4040**, purchased as an **Adafruit breakout board** from Digi-Key.
- The parts ordered included:
    - Stacking header for Raspberry Pi (part #1528-1783-ND)
    - Grove 2mm 4-pin right angle connectors (1597-1083-ND)
    - 4-pin STEMMA/GROVE QWIIC cables of varying lengths for I^2C connection (1528-4528-ND, 1568-22726-ND, 1528-5385-ND)
    - Machine screws for mounting (145-50M025045I016-ND)
- These components allowed easy connection and mounting of the sensor breakout board to the Raspberry Pi via the I^2C interface using STEMMA QT/QWIIC connectors.

2.2 PCB Design and Soldering

- A custom PCB schematic was designed using **KiCad**, following the sensor breakout pinout and Raspberry Pi I^2C header layout.
- This PCB integrated the sensor breakout header and connectors for streamlined wiring.
- The PCB was soldered manually; as this was the first soldering experience, the solder joints were initially non-circular, requiring careful reheating and adjustment. Despite challenges, all components were successfully soldered with no cold joints.

2.3 Case Design and Assembly

- The sensor was mounted inside a custom-designed case created using **Inkscape** for layout design.
- The case served primarily to position the sensor correctly near the air purification system.
- The sensor breakout board was glued securely inside the case, with openings to allow unobstructed IR proximity sensing.
- The design ensured stable placement and protection while maintaining sensor functionality.

2.4 Firmware Development and Use

- Firmware was developed in **Python**, running on the Raspberry Pi.
- The **Adafruit VCNL4040 Python library** was used to interface with the sensor over I^2C.
- Sensor readings were collected continuously and sent to the project's **Firebase** database over Wi-Fi.
- The Android app subscribed to proximity updates and triggered interactive animations (waves on water, glowing effects) when motion was detected.
- Additionally, if motion was detected continuously for over one minute, the app sent a "Thanks for dropping by" notification as part of the gamification experience.

3.0 Testing and Results

- Testing included distance measurement, hand-waving, and object detection near the sensor.
- The VCNL4040 reliably detected objects up to approximately **20 cm** distance.
- Ambient lighting effects on accuracy were not fully quantified but the sensor's IR emission helped reduce interference in typical indoor conditions.
- Firebase logs and Android app behavior confirmed that proximity data was transmitted correctly and the intended visual and notification effects triggered as expected.

4.0 References

- Adafruit VCNL4040 Proximity Sensor Datasheet: https://cdn-learn.adafruit.com/downloads/pdf/adafruit-vcnl4040-proximity-sensor.pdf
- Adafruit VCNL4040 Python Library: https://github.com/adafruit/Adafruit_CircuitPython_VCNL4040
- KiCad PCB Design Software: https://kicad.org/
- Raspberry Pi Official Documentation: https://www.raspberrypi.org/documentation/