Design and Development of the LiDAR Cane with

Performance Evaluation and Key Insights

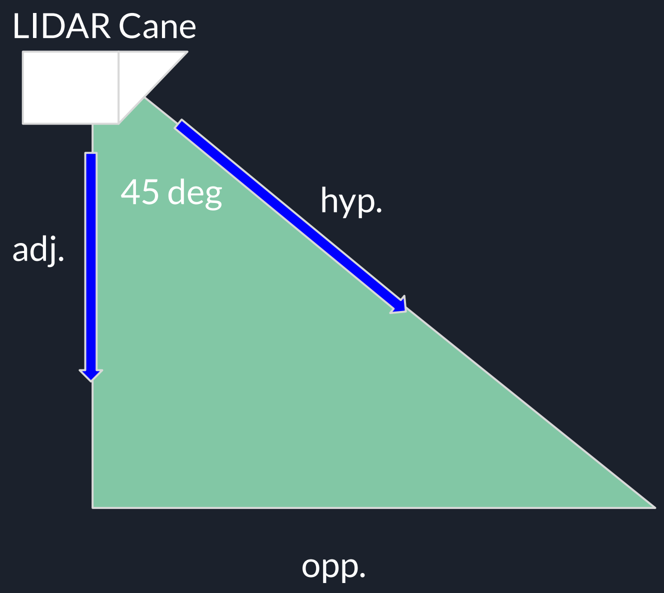
By Abhisek Limbu, Ousmane Barry, and Felix Tavarez

Objective

- Bring LiDAR cane into the healthcare market for visually impaired individuals
- Innovative assistive technology with LiDAR for high-accuracy obstacle detection
- Cost-effective solution
- Reliability and durability for everyday use

Core Design

- We positioned the two LIDAR modules to create a right triangle
- **Known values:** 45 degree angle, adjacent value from lidar 1, and hypotenuse value from lidar 2
- Opp1 = adj * (tan 45)
- Opp2 = hyp * (sin 45)
- If |Opp1 - Opp2| > 0.3m, then there’s an obstacle



Overview of Source Code

- Written in MicroPython
- Object Oriented programming with threading
- Consists of two scripts: tf_luna.py and main.py
- **`tf_luna.py`**
  - A class for LIDAR module
  - Decodes bitstream into distance
- **`main.py`**
  - Assigns pinouts to LIDAR modules and NPN transistor
  - Creates independent threads for LIDAR module 1 and 2 to run in parallel
  - Runs obstacle detection algorithm by computing OPP1 and OPP2 every second

Source Code

- Conversion of bitstream into decimal values
- The function processes raw data and processes bytes into an integer values (EX: Dist_L and Dist_H into distance)
- Frames contain integer representation of distance, amp (signal strength indicator), and temperature values
- Distance is extracted from the most recent frame in the list

Byte	0	1	2	3	4	5	6	7	8
Description	0x59	0x59	Dist_L	Dist_H	Amp_L	Amp_H	Temp_L	Temp_H	Check_sum

```
class TFLuna:

    def process_lidar_raw_data(self, raw_data):
        frames = []
        i = 0

        while i <= len(raw_data) - 9:
            # Search for the correct frame header (0x59 0x59)
            if raw_data[i] == 0x59 and raw_data[i+1] == 0x59:
                frame = raw_data[i:i+9]

                # Check if the frame length is correct
                if len(frame) == 9:
                    # Extract data from the frame
                    dist = frame[2] | (frame[3] << 8)   #shift frame[3] 8-bits to the left and combine with frame[2] into a single 16-bit integer
                    amp = frame[4] | (frame[5] << 8)
                    temp = frame[6] | (frame[7] << 8)
                    checksum = frame[8]

                    # Compute checksum
                    computed_checksum = sum(frame[:8]) & 0xFF

                    if checksum == computed_checksum:
                        # Convert temperature to Celsius
                        temp_celsius = temp / 8 - 256

                        # Check if distance measurement is reliable
                        if amp < 100 or amp == 65535:
                            dist = None  # unreliable measurement
                            print("Unreliable measurement")

                        frames.append((dist, amp, temp_celsius))
                    else:
                        print("Checksum error in frame:", frame)

                # Move to the next frame
                i += 9
            else:
                i += 1  # Move forward to find the correct start of the frame

        return frames
```

Source Code (Continued)

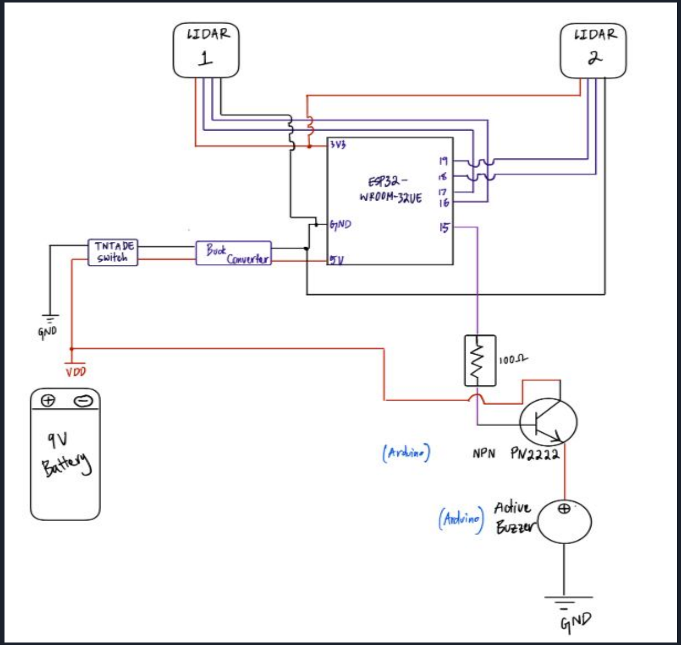
- Obstacle detection algorithm
- Lidar 1 measures height to floor
- Lidar 2 measures diagonal distance
- X1 and X2 (OPP1 and OPP2) are used for obstacle detection
- If X1 is less than X2, there is a structure
- If X1 is greater than X2, then there is a drop

```python
# Perform distance and obstacle checks only if both lidar readings are available
if lidar1_height_to_floor and lidar2_diagonal_distance:
    print("Theta (radians):", theta_rad)
    X1 = lidar1_height_to_floor                        #since tan(45) = 1
    X2 = lidar2_diagonal_distance * sin(theta_rad)
    print(f"X1: {X1:.10f}, X2: {X2:.10f}")

    #Compare with x_threshold: if below threshold, then no obstacles
    if abs(X1 - X2) > x_threshold:
        if X1 < X2:
            print("Obstacle: structure detected")

        elif X2 > X1:
            print("Obstacle: drop detected")

        buzzer.value(1)     #Activate buzzer
    else:
        buzzer.value(0)
else:
    buzzer.value(0)         #Ensure buzzer is off if the distance is invalid
```

Schematic Analysis

- Buck converter is used to step down from 9V to 5V
- Current through the resistor:
  - I = V/R
  - I = 5V/ 100 Ω = 0.05 Amp
- NPN transistor acts as a closed switch when it receives 0.05 Amp
- This allows the buzzer to trigger
- Note: NPN transistor is a current-driven

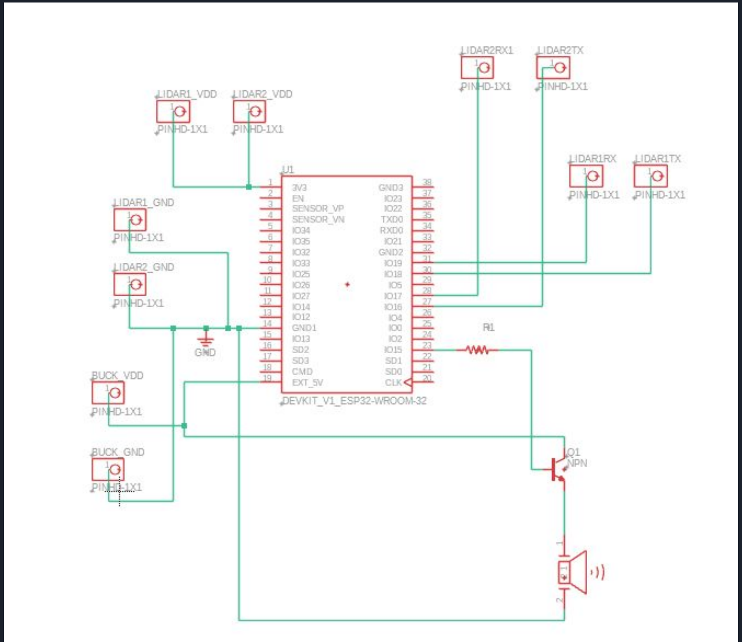
  BJT



PCB Wiring

Design and Development:

- Hardware design services including schematic design and PCB layout.
- Software development services for the ESP32 microcontroller.
- Integration of LiDAR module and feedback system into the cane design.



PCB/ wiring

<u>PCB:</u> Electronic Components Overview

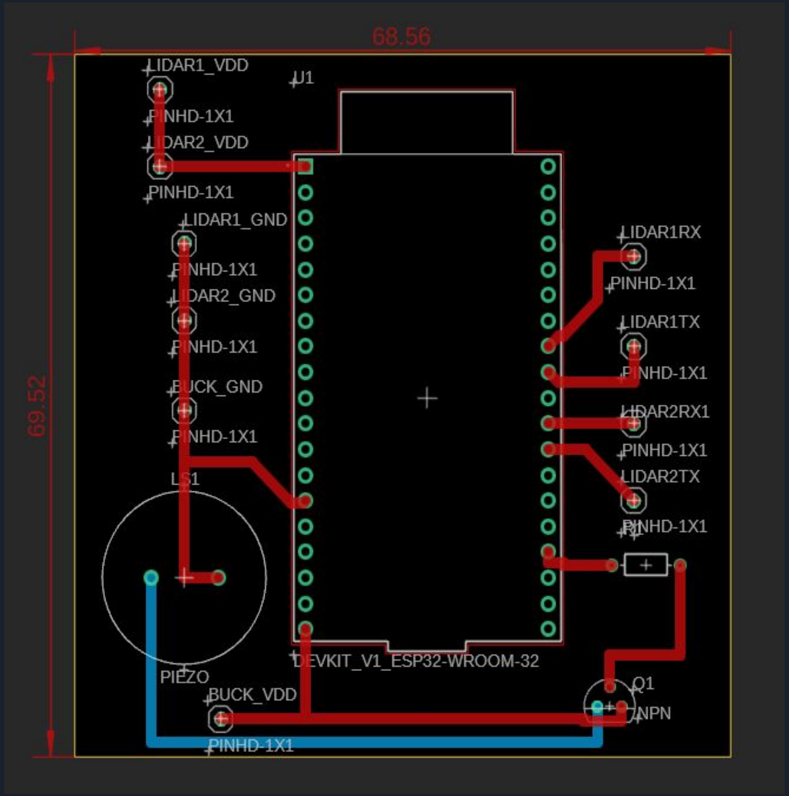
• PCBs are core components of electronic gadgets.
• Typically, a fiberglass substrate or FR4 provides thickness and structure.
• Copper foil layers are bonded to create conductive channels.
• A solder mask protects copper traces against short circuits.
• Silkscreen layers mark components and other information.
• Surface treatments like HASL, ENIG, OSP, and immersion silver/tin offer solderable surfaces and shield copper from oxidation.

·PCB Design Utilization:
• Utilizes Autodesk Fusion 360 software for synchronizing Schematics and PCBs.
• Provides a seamless design experience by maintaining a real-time link between schematic design and PCB layout.
• Instantly reflects changes in schematic design in PCB layout, ensuring consistency and preventing conflicts.
• Enables efficient work, reduces errors, and speeds up the design process.

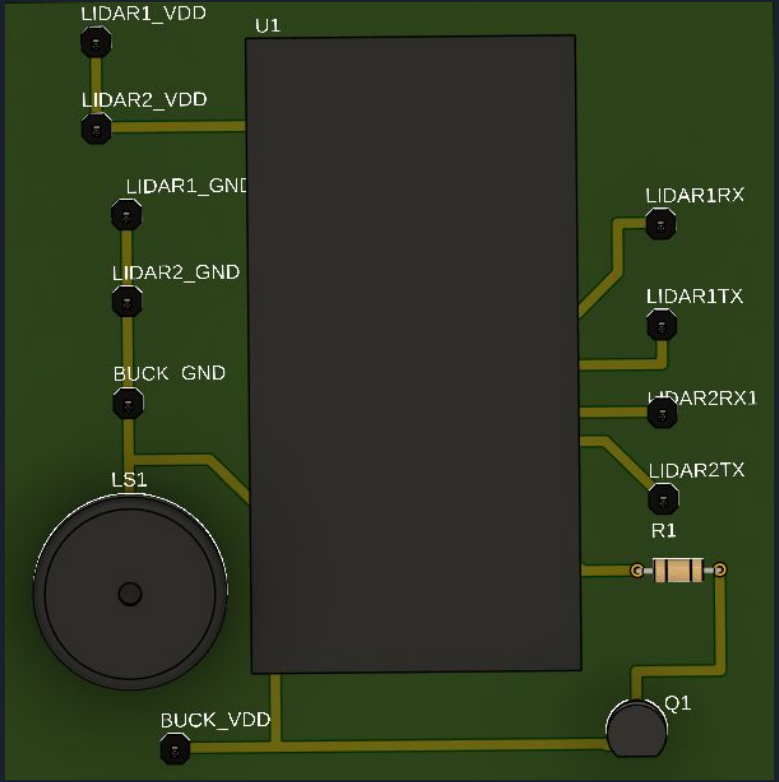
Soldering Techniques: Preheating and Soldering Circuit Board
• Preheat printed circuit board and soldering iron.
• Position components on PCB through holes or fold leads.
• Apply flux paste to solder pads to prevent air and water entrapment.
• Apply solder to the joint, press down firmly, and let it spread.
• Remove soldering iron and wire.
• Resolder defective joints.
• Allow solder to cool before relocating the joint.
• Sanitize the printed circuit board.

PCB

PCB Design Layout



3D PCB



Hardwares

Microcontroller:

LiDAR Data Interpretation in System Operation

• LiDAR data interpretation is crucial for microcontroller operation.

• Distance calculation is done using the distance formula: Distance = (speed of light × time taken) / 2.

• The speed of light is used to evaluate LiDAR measurements.

• Feedback signals are produced based on LiDAR readings, triggering reactions when distance falls within predefined parameters, usually about 4 meters.

ESP32-WROOM-32 UE

TF-Luna LiDAR Module

TF-Luna LiDAR Module Incorporation

• Low-cost, small-size apparatus.

• Meets project requirements.

• excels at level measurement.

• Low power consumption.

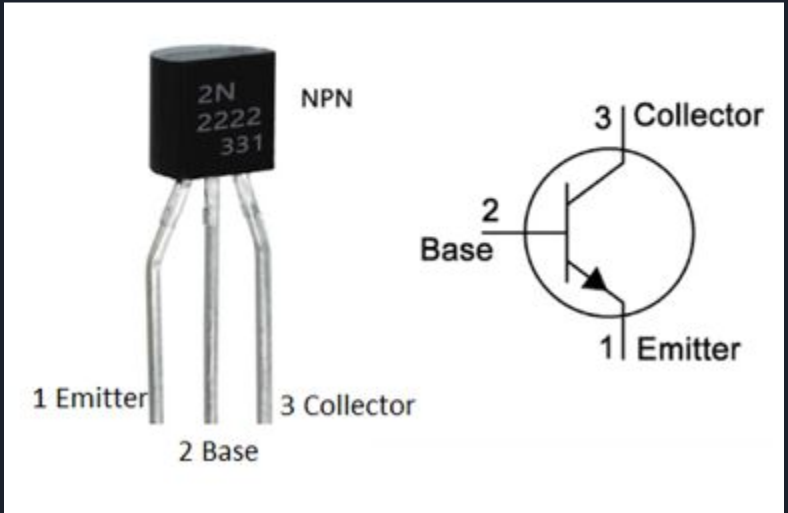
• Lightweight.

Hardwares

Piezoelectric Buzzer: The piezoelectric buzzer provides audible alerts or signals based on the microcontroller's logic, such as proximity warnings from the LIDAR data.

Specifications:
Operating Voltage: 3-5V
Sound Output: 85 dB at 10cm

PN2222: The **2N2222** is a common **NPN** bipolar junction transistor (BJT) used for general purpose low-power amplifying or switching applications.



Hardwares

Buck Converter: Buck Converter Overview
• DC-to-DC converter.
• Decreases voltage while increasing current.
• Class of switched-mode power supply.

Buck Converter

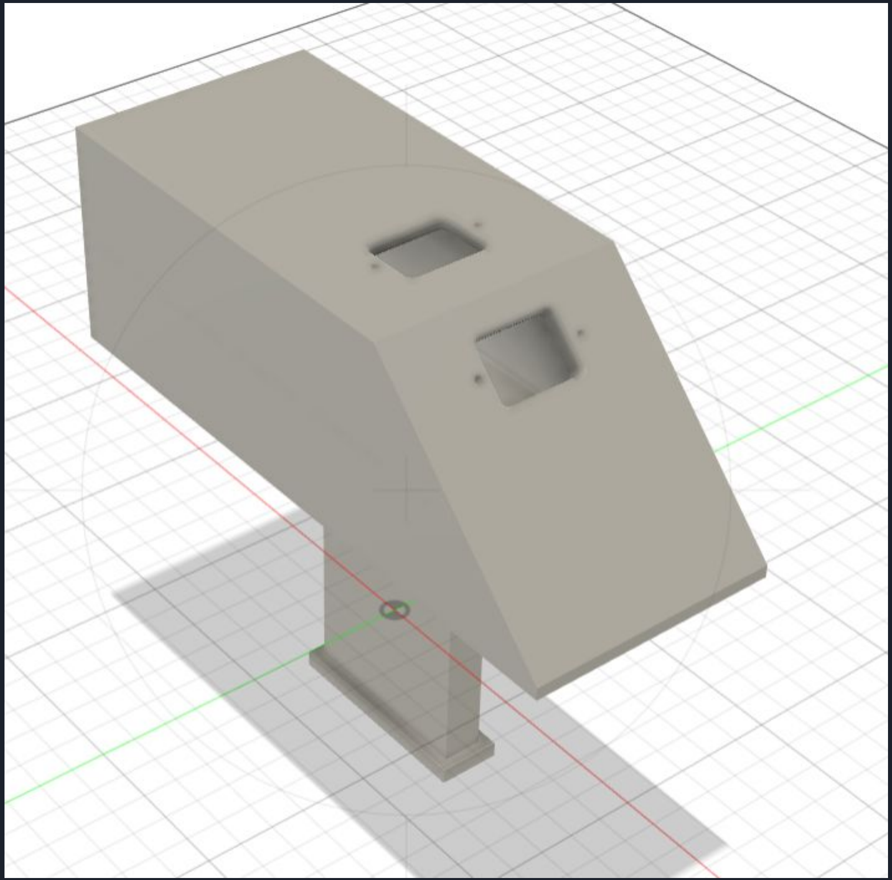
Switches:
• Single button manages battery power consumption.
• Switch controls primary power supply.
• Functional purpose: conserve battery power.

Switch Button

Power Consumption Models Overview
• Represent network equipment power consumption based on traffic load.
• Traffic load refers to user resource consumption.
• Power module: 9V-9800MAH lithium battery.

Enclosure

- For the device to function the enclosure needs to be constructed with:
- A fixed 45 degree angle between the LiDAR modules
- It must be handheld/ easy to hold
- Printing was done in the makerspace with assistance from Super Users to set proper density and build orientation.



Earlier design iteration (version 20)

Test Report 1: Single LiDAR Module

- Objective: Test a single LiDAR module’s distance measurement accuracy.
- Results: Average operation point was 0.99m for a target of 1m.
- Key Insight: Unable to detect altitude changes, necessitating redesign.

Test Report 2: Dual LiDAR Module

- Objective: Measure height and distance simultaneously.
- Results: Modules operated successfully; 45° angle required.
- Insight: Fixed enclosure needed to maintain alignment.

Test

- Device performance was tested inside the Grove School of Engineering
- Accurately detects obstacles including stairs

Test Report 3: Enclosure and Outdoor Testing

- Enclosure Test: Verify alignment and stability in the enclosure.
- Outdoor Test: Evaluate object detection (e.g., curbs, obstacles).
- Goal: Assess real-world functionality and limitations.

Deliverables

- A fully functional LiDAR Cane
- Final Report regarding design and development of LiDAR Cane