Angled ToF Laser-Based Non-Contact Height Measurement

Course: Embedded Systems (CS336) | **Platform:** AVR ATmega328P

Group Members:

- Faisal Iqbal (478364)
- Hassan Shahid (464485)
- Syed Muneeb ur Rehman Bukhari (465315)
- Talha Shafat (460861)

Table of Contents

Problem Statement

Measuring dimensions in real-time industrial environments is challenging using traditional ultrasonic or IR sensors due to beam spread and slow response.

> **Many applications require a compact and fast height measurement system that can detect an object's height instantly as it enters the sensing region.**

Why is this needed?
Modern automated lines (like conveyor belts) move fast.
Applications like: Automatic vehicle classification at tolls, parcel dimensioning in logistics, and quality control on assembly lines all require instant, non-contact height data to function efficiently.

Proposed Solution

> **What is ToF?**
> **Time-of-Flight (ToF)** is a method for measuring distance by calculating the precise time it takes for a light signal (laser) to travel to an object and reflect back to the sensor.

A laser-based system using the **VL53L0X Time-of-Flight sensor** mounted at a fixed angle, interfaced with an **Arduino UNO**.

Key Features:

- **Long Range:** Reliable detection up to 1 meter.
- **Instant Feedback:** No missed entries; < 30ms response.
- **Dual Units:** Height displayed in cm and feet/inches on LCD.

Block Diagram

Methodology

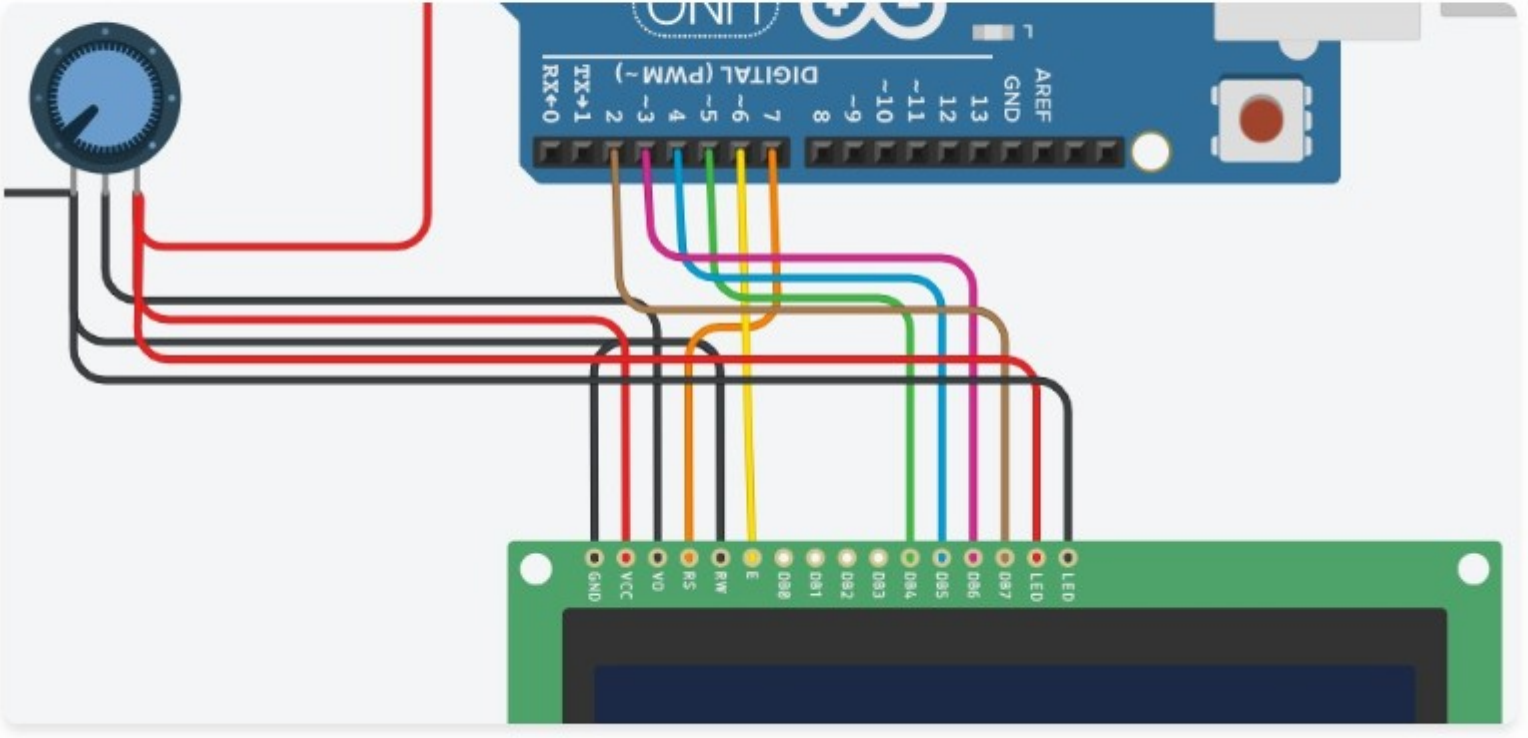
1. Hardware Setup

- Sensor mounted at 45° angle.
- I²C bus (SDA/SCL) for sensor data.
- 4-bit parallel mode for LCD.

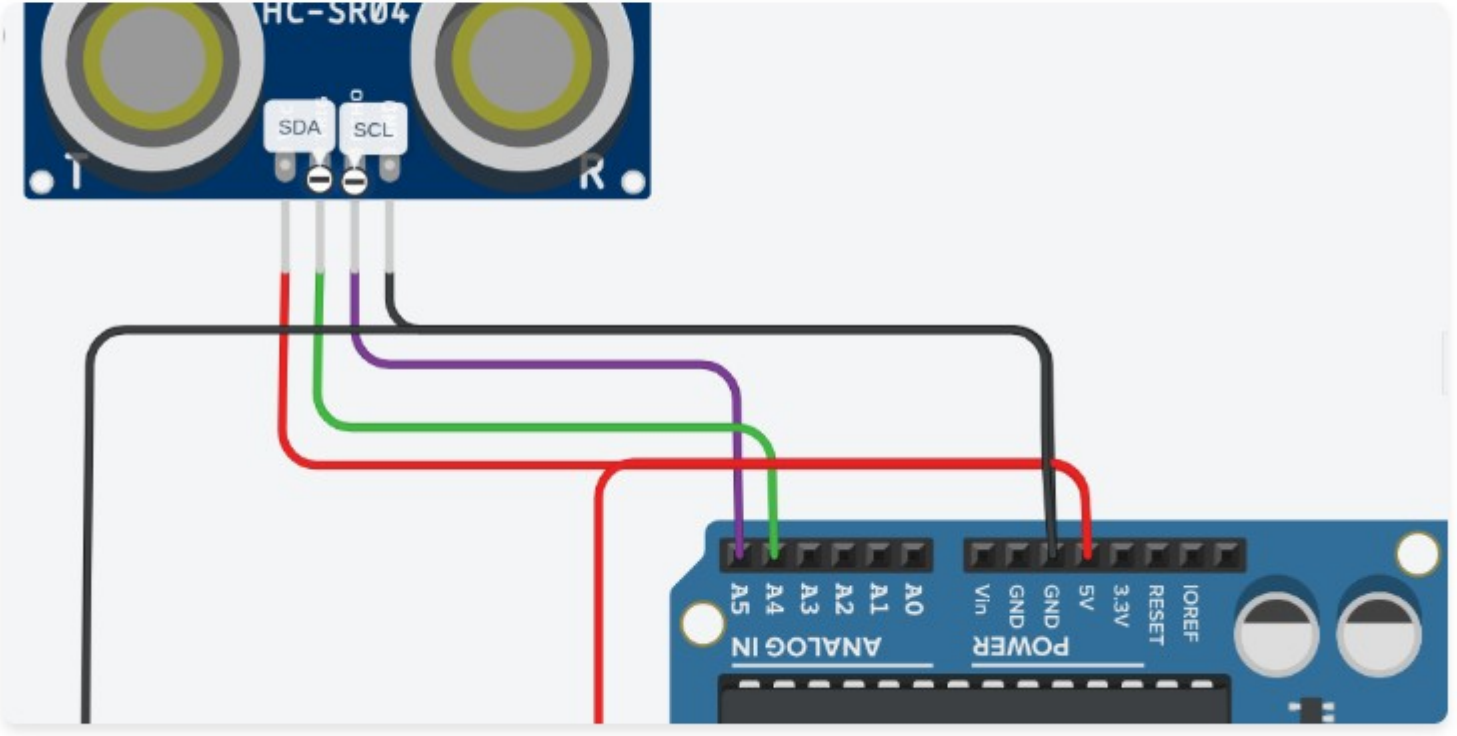
2. Software Logic

1. **Init:** Calibrate sensor & LCD.
2. **Scan:** Read distance continuously.
3. **Detect:** If reading < Baseline, Object detected.
4. **Calc:** Trigonometry to find vertical height.

Circuit Diagram



Circuit Diagram



Code Implementation

```
#include <LiquidCrystal.h>

// LCD PIN MAPPING
// RS, E, D4, D5, D6, D7 to arduino PORT D

// VL53L0X SENSOR OBJECT
Adafruit_VL53L0X lox = Adafruit_VL53L0X();

// SENSOR HEIGHT FROM FLOOR IN CM
const float SENSOR_HEIGHT_CM = 200.0;

void setup() {

    // Initialize the LCD
    lcd.begin(16, 2);
    lcd.print("Height Meter");

    // Initialize the VL53L0X distance sensor
    lox.begin();
}

void loop() {

    // Structure to store distance measurement data
    VL53L0X_RangingMeasurementData_t measure;

    // Check if the measurement is valid from the previously returned object
```

Code Implementation

```
// Structure to store distance measurement data
VL53L0X_RangingMeasurementData_t measure;

// Check if the measurement is valid from the previously returned object
if (measure.RangeStatus != 4) {

    // DISTANCE CALCULATION
    float distance_mm = measure.RangeMilliMeter;
    float distance_cm = (distance_mm / 10.0) - 1.9;

    // HEIGHT CALCULATION
    float height_cm = (distance_cm * 0.70710678) + SENSOR_HEIGHT_CM;

    // Noise filtering: height should never be negative
    if (height_cm < 0) height_cm = 0;

    // CONVERSION FROM CM TO FEET & INCHES
    int total_inches = height_cm / 2.54;
    int feet = total_inches / 12;
    int inches = total_inches % 12;

    // LCD DISPLAY
    lcd.setCursor(0, 0);
    lcd.print("Dist: ");
    lcd.print(distance_cm, 1);
    lcd.print(" cm  ");

    lcd.setCursor(0, 1);
```

Code Implementation

```
        // LCD DISPLAY
        lcd.setCursor(0, 0);
        lcd.print("Dist: ");
        lcd.print(distance_cm, 1);
        lcd.print(" cm  ");

        lcd.setCursor(0, 1);
        lcd.print("H:");
        lcd.print(height_cm, 0);
        lcd.print("cm ");

        lcd.print(feet);
        lcd.print("'");
        lcd.print(inches);
        lcd.print("\" ");

    } else {
        lcd.setCursor(0, 1);
        lcd.print("Out of Range  ");
    }

    // Small delay to actually realize the next measurement
    delay(1000);
}
```

Calculations & Visualization

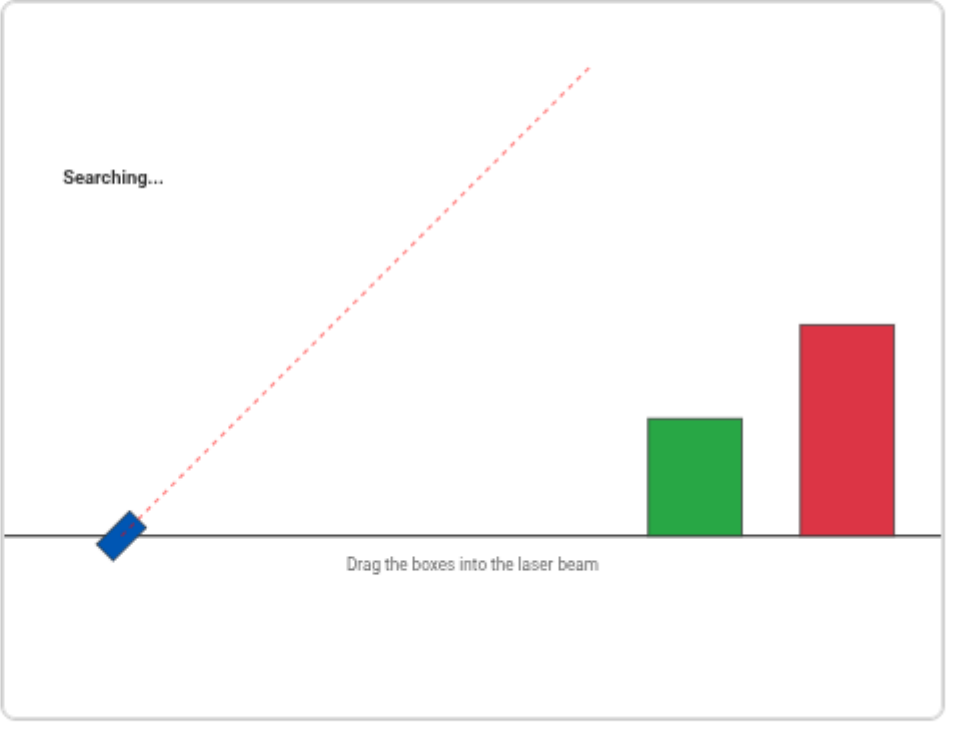
The system uses trigonometry to derive vertical height from the angled distance reading.

The Math

Hypotenuse: Raw distance reading.

Offset: Sensor height from floor.

$$\text{Height} = \text{Hypotenuse} \times \sin(45°) + \text{Distance\_of\_sensor\_from\_floor}$$



Calculations & Visualization

The system uses trigonometry to derive vertical height from the angled distance reading.

The Math

Hypotenuse: Raw distance reading.

Offset: Sensor height from floor.

$$\text{Height} = \text{Hypotenuse} \times \sin(45°) + \text{Distance\_of\_sensor\_from\_floor}$$

Results

⚡

Real-Time

Instant detection and LCD update latency under 50ms.

📏

Unit Conversion

Seamless automated conversion between Centimeters, Feet, and Inches.

Applications to Society

Industrial Automation

- **Luggage Sorting:** Airports use height to classify bag sizes.
- **Logistics:** Couriers dimension parcels automatically.

Civil & Safety

- **Vehicle Clearance:** Warning trucks before tunnels/bridges.
- **Robotics:** Obstacle height mapping for autonomous rovers.

Contribution of Group Members

Member	Key Responsibilities
Faisal Iqbal	Sensor integration, Trigonometry logic calibration
Hassan Shahid	Circuit design, Hardware assembly, LCD interfacing
Syed Muneeb	Embedded C++ programming, Debugging
Talha Shafat	Project documentation, Presentation design, Testing

Conclusion

We successfully designed and implemented a non-contact height measurement system.

- **Success:** The device meets the requirement for a compact, low-cost industrial measuring tool.
- **Learning:** Gained deep understanding of I²C communication and Angled ToF physics.
- **Improvement:** We can use a better sensor and better angle to get more accurate results in future iterations.

Any Questions?

Thank you for your attention.

References

1. STMicroelectronics. "VL53L0X Datasheet - Time-of-Flight Ranging Sensor".

2. Arduino.cc. "Arduino UNO R3 Documentation".

3. Lab Helping Material for 'ThinkerCad'"