Project 1 - Bidirectional People Counter

Robert Allen

Ananta Thapaliya

Mobile & Pervasive Computing

PROBLEM STATEMENT

In national parks and hiking trails where it is unreasonable to deploy humans at various checkpoints to track visitors, some automated mechanism for counting people is crucial. This lets the rangers know if people have been left behind after park hours. This applies to buildings too, especially at times of danger (fire, earthquake) when it is important to know that everyone has evacuated. In short, keeping track of the flow of people is useful in many circumstances.

SOLUTION

What metric would reflect the flow of people? First, we need to know if we must track who entered and exit. If yes, it would require a camera or barcode reader. If we are just interested in the "how many" question and not "who," a simpler sensor would suffice.

User Story	**Acceptance Criteria**
`As a head ranger of XYZ national park, I need to know everyday if all three visitors have gotten out of the park before the park closes, so that we track individuals who might have been left behind and plan our search operation accordingly.`	`A small battery powered deployable device that keeps track of the people who have entered but not exited the national park area.`

So, a device capable of detecting and counting people will fulfill the acceptance criteria.

PROTOTYPE

Apparatus Required

1. Arduino Board (x 1)
2. Ultrasound Sensors (x 2)

3. LED Lights (x 2)
4. Jumper Wires
5. Breadboard (x 2)

Theory

Humans can hear sounds of frequencies ranging from 20 Hz 20khz. Ultrasound sensors emit sound waves of frequency beyond this range. When the sound wave faces an obstacle, it reflects back to the sensor. It then measures the time of flight (T in microseconds).

velocity of sound (v_s) = distance(d) / time taken(t)
t = T/2, because the sensor measures the round trip time

So, distance (d) = v_s * (T/2) = 0.034 cm/microseconds * T

Arduino Code:

```
int dur = pulseIn(echoPin,HIGH);
int dist = dur*0.034/2;
```

Methodology

Measuring Distance

1. Set the trig pin to LOW
2. Delay for 2 microseconds to let the sensor detect the low state of the transmitter
3. Set the trig pin to HIGH, allowing the transmitter to emit ultrasound waves
4. Delay for 10 microseconds, allowing the transmitter to emit enough pules
5. Set the trig pin to LOW, stopping the emission
6. Set the echo pin to HIGH and record the duration captured by the echo pin
7. Apply the physics explained in the theory section
8. Obtain the distance

Detecting an Event:

- Check if the last distance reading is below a certain threshold for a specified amount of time.
- If that condition is met, then there is an object passing in front of the sensor, which starts an event for that sensor.

During an Event:

- Once an event is triggered for one sensor, the direction is known (since the code is sequential). So when the second sensor event is triggered, the counter is incremented or decremented depending on which sensor was first.
- The events for both sensors are then reset.

Resetting an Event

- Since an object could remain in the sensor range after the counter is updated, they might set off another event when they shouldn't. So we wait for them to leave the sensor range completely before continuing to check for new events.
- Similar to detecting an event, this checks if the last reading is above the threshold (meaning it has returned to normal readings) for a certain amount of time, it continues checking until this condition is met, preventing any unwanted readings if the object remains in the sensor too long.