Nidhi Roshan Mundhada
Email id: nidhirmundhada@gmail.com
Contact no: 8379926546

Range Finder using Raspberry Pi

PROBLEM STATEMENT

To design and implement an ultrasonic distance measurer using Raspberry Pi.

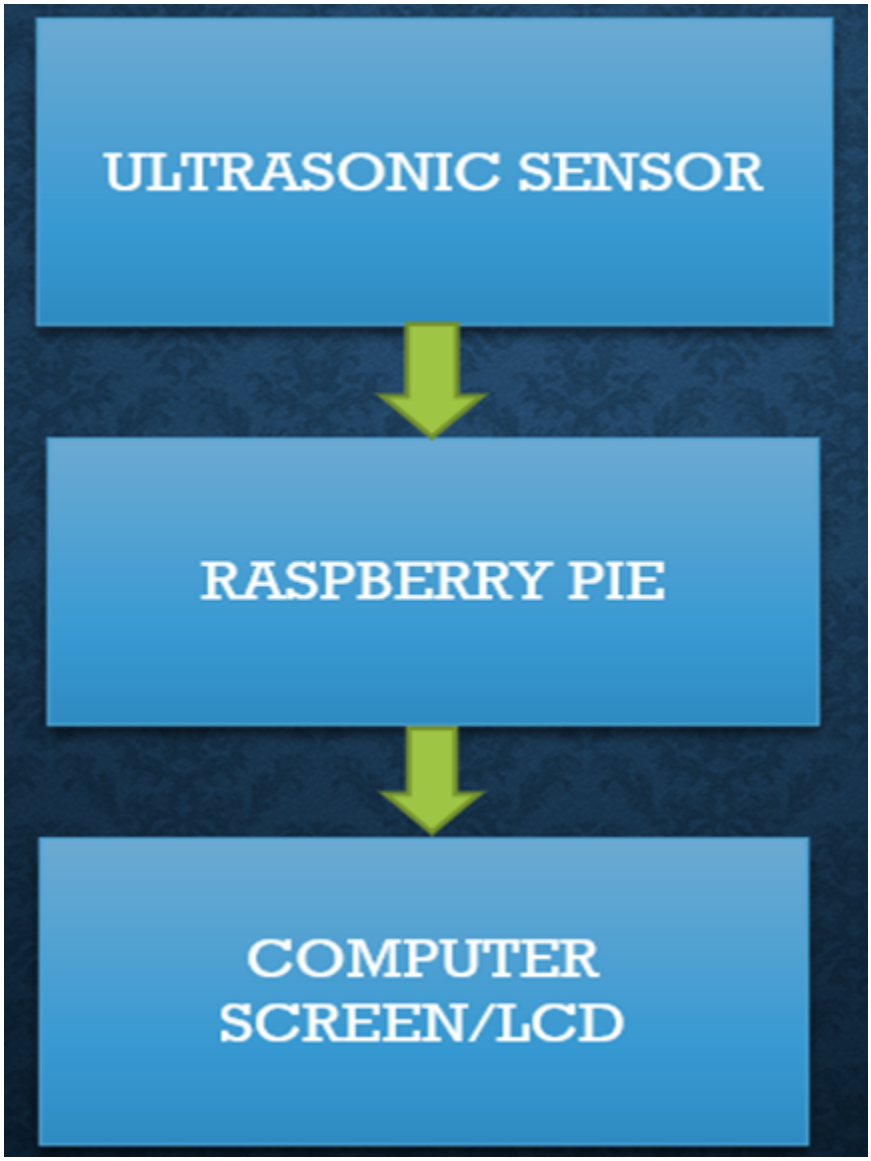
SOLUTION

Ultrasonic Sensor is used for distance measurement and gives fairly accurate readings as its using ultrasonic sound to find the distance. We have used python program for setting the algorithm.

COMPONENTS USED

- Raspberry Pi 3B: Raspberry Pi is a Single Board Computer or SBC. It is based on Broadcom SoC, an ARM Cortex A Series Microprocessor. It controls ultrasonic sensor in this project.
- Ultrasonic sensor: It is used to transmit and receive ultrasonic waves.
- 1k & 1.5k: It is used for voltage divider in this project.
- GPP: Implementation purposes.
- Jumper wires: For connections.

BLOCK DIAGRAM

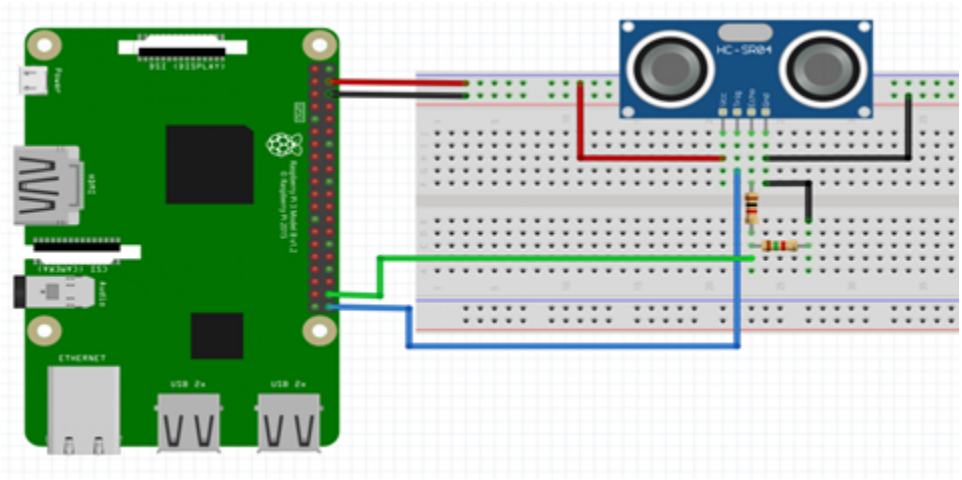


BLOCK DIAGRAM EXPLANATION

The ultrasonic sensor works on reflection phenomenon. Transmitter transmits a short audio bust and when it is reflected back, it is received by the receiver. Based on time of transmission and reception, we can calculate how much distance travelled using a simple formula. We have designed it in a way, such that the echo pin works as input

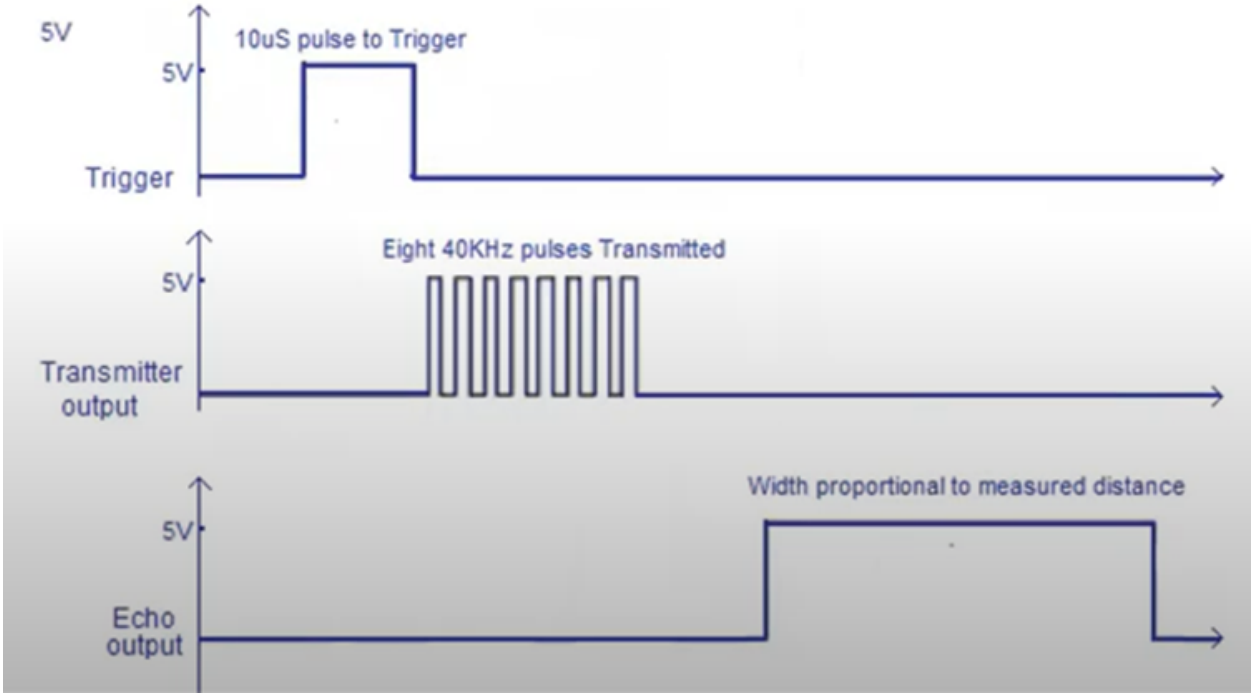
and the trigger pin works as output and these have been connected to the raspberry pi. The Rpi is then interfaced with the LCD/computer screen then we obtain the distance displayed on the screen.

CIRCUIT DIAGRAM AND WORKING



We have connected the ultrasonic sensor on the breadboard. The ultrasonic sensor's pins which are Vcc, Trigger, echo and ground have been connected to the respective terminals. The entire operation is based on the reflection phenomenon of the ultrasonic sensor. The ultrasonic sensor has two important parts: transmitter and receiver. As soon as a short audio bust of 10us is given to the trigger pin, a simple audio bust of ultrasound is transmitted, and as soon as this happens, the echo pin becomes high. And when audio reflects back, it is received by the receiver and the echo pin becomes low. The total duration of the echo pin corresponds to the total time taken by ultrasound to transmit and reflect back to the sensor. But when the sound is travelling, it is covering the same distance twice, and hence while calculating the distance travelled, it will be divided by 2. Using the formula for distance and taking speed of sound at sea level to be 343 m/s or 34300 cm/s, we obtain an expression for distance, Distance = 17150*Time

The echo pin works as input to the rpi, and hence, we use resistors to reduce voltage to 3.3V obtained from the Vcc which is +5V. For this, firstly, we keep the trigger pin low, then it is switched to high for only 10us, and then finally low towards the end. While this happens, the echo pin also switches between high and low. When it is low, we start a timer and then it goes high, the counter gets updated. This counter is updated using function time. This function gives the epoch time when the trigger pin becomes high and when finally becomes low. The difference between the high epoch time the low epoch time gives the entire duration of time of the pulse which can then be used to obtain the distance using the formula mentioned above.



Speed of Sound = 343 m/s = 34300 cm/s

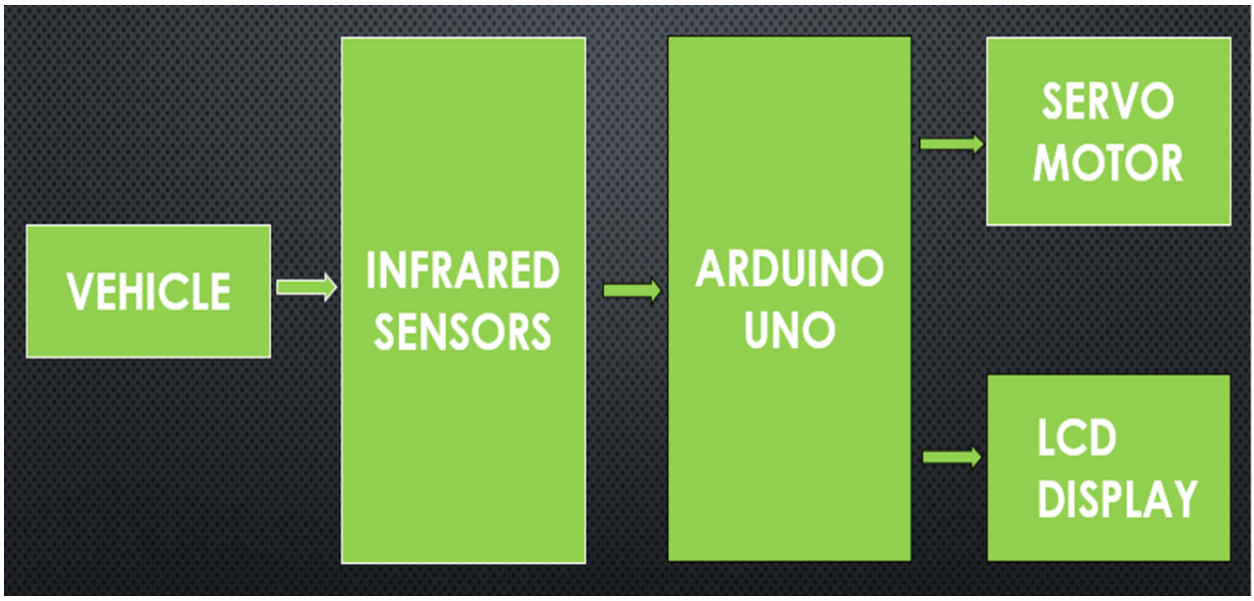
Distance = Time x Speed

But, the time required to travel is twice the required time.

Distance = (Time x Speed)/2

= 17150 x Time in cm

BLOCK DIAGRAM

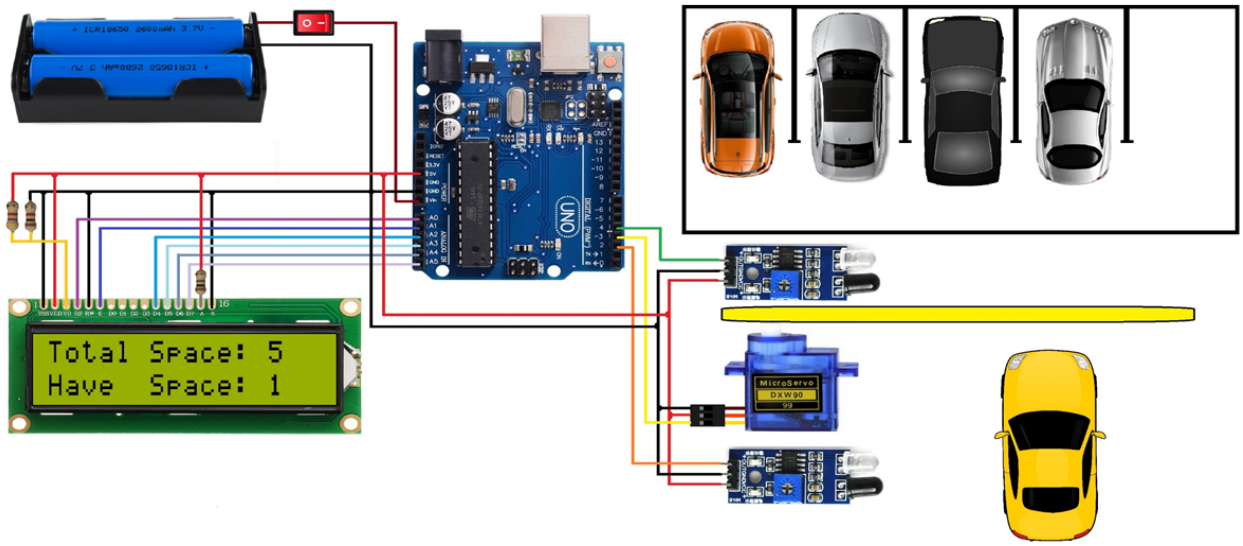


BLOCK DIAGRAM DESCRIPTION

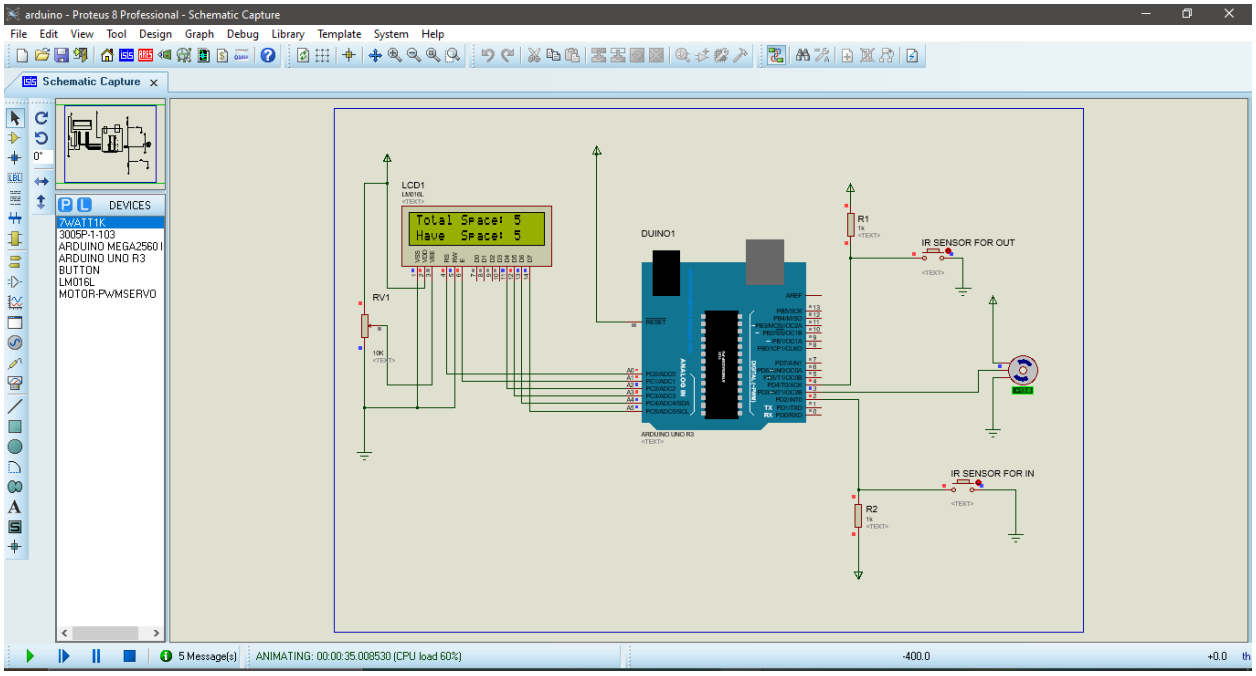
To design this car parking system circuit using ARDUINO UNO, we have placed IR transmitter, receiver pair at the initial position in the parking slot. IR transmitter transmits Infrared signal or rays into the environment continuously. As soon as a vehicle/ obj is encountered by the IR rays, there is a reflection and as IR receiver receives it, a digital value (0/1) is generated at the output pin which is then connected to the Arduino. The Arduino stores the signal received. And as a result, sends a signal to the servo motor via the signal wire, which decides by how much angle the servo motor has to be rotated. The servo motor rotations decides whether the barrier will open/close for the vehicle to enter/exit from the parking lot. Output from the Arduino is also sent to the LCD display. If the RS pin of the LCD is made high, and is feed an input at its data lines, the input is treated as data to display on the LCD screen. Similarly, if the RS pin is made low,

and is feed an input at the data lines, then this will be treated as a command to be written to LCD controller. In this way, the LCD shows the number of vacant and occupied slots in the parking lot.

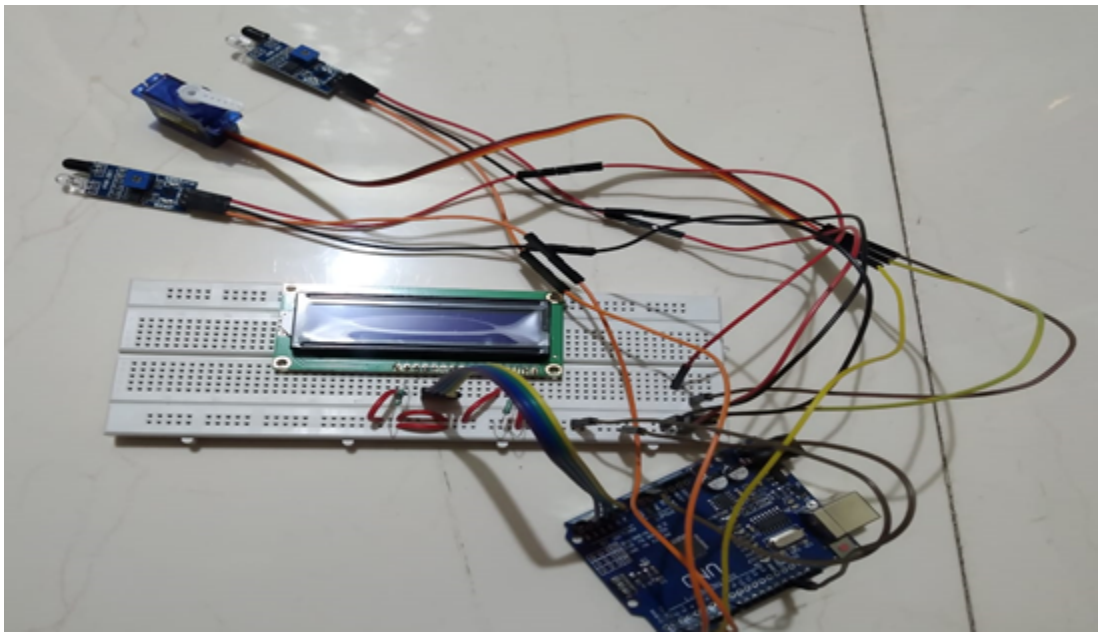
CIRCUIT DIAGRAM



PROTEUS SOFTWARE SIMULATION



HARDWARE IMPLEMENTATION



WORKING

We have used 2 IR sensors, out of which one is at the entry point. An IR Transmitter continuously transmits IR rays into the environment, the rays reflect back to the receiver after striking any vehicle trying to get in/out of the parking area. When a vehicle approaches the first IR sensor, a signal is sent to the Arduino and in turn the Arduino sends a signal to the servo motor which in turn rotates, and the barrier thus opens/closes depending upon the respective angles of rotation.

Push buttons have been used instead of IR transmitter receiver pair while performing in the PROTEUS SOFTWARE, because we cannot use IR transmitter receiver pair in the PROTEUS SOFTWARE. The designing has been in a way such that the servo motor rotates from 0 degrees to 100 degrees. As soon as a car approaches the first IR sensor, a high output signal is sent from the IR sensor to the Arduino, which in turn sends a signal to the servo motor and thus the servo motor rotates from 0 to 100 degrees, and the barrier opens up for the vehicle to get in. As soon as the car enters, the Have Space variable decrements by 1. After this as the car is out of the IR sensor 2, after a delay of 1 second, the barrier would close again. Now the next time, another vehicle approaches, the same procedure continues. Simultaneously, the Have Space variable would go on decrementing until it reaches the value 0. At this moment, the LCD would display the message “SPACE NOT AVAILABLE” and the barrier won’t open. Now as the vehicles would start exiting out of the parking area, they would first approach the IR sensor 2, by which the barrier would open, incrementing the Have Space variable by 1 and as soon as the vehicle would be out of IR sensor 1, the servo motor would rotate from 100 to 0 degrees, closing back the barrier. Similar procedure would continue.

APPLICATIONS

The system uses Arduino UNO to handle chaos- free and guided parking.

It is so handy that it can be installed at places like:

Parking lots of offices, malls, toll plazas, commercial buildings, etc.

DRAWBACK

The sensor cannot differentiate between a vehicle and any other object i.e. the system would record the entry of even a human passing through the sensor.

FUTURE SCOPE

The system can be upgraded by adding image recognition software which would basically detect the shape and size of the object and differentiate between any other object and a vehicle. Thus the drawback introduced earlier can also be removed using this.