Arduino Code for Measuring Distance with Ultrasonic Sensor

Bard

This code uses an ultrasonic sensor to measure the distance to an object and displays the measurement on the serial monitor.

1 Code Explanation

```
const int trigPin = 12; // Trigger pin of ultrasonic sensor
const int echoPin = 11; // Echo pin of ultrasonic sensor

void setup() {
  Serial.begin(9600); // Initialize serial communication
  pinMode(trigPin, OUTPUT); // Set trigger pin as output
  pinMode(echoPin, INPUT); // Set echo pin as input
}

void loop() {
  long duration, distance;

  digitalWrite(trigPin, LOW); // Set trigger pin low for 2 microseconds
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH); // Set trigger pin high for 10 microseconds
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW); // Set trigger pin low

  duration = pulseIn(echoPin, HIGH); // Read the duration of the echo pulse
  distance = duration * 0.0343 / 2; // Calculate distance in centimeters

  if (distance > 150) { // Check for maximum distance
    distance = 150;
  }

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(50); // Delay for 50 milliseconds
}
```

1.1 Pin Definitions

trigPin: This constant stores the pin number connected to the trigger pin of the ultrasonic sensor. It is set to pin 12.

echoPin: This constant stores the pin number connected to the echo pin of the ultrasonic sensor. It is set to pin 11.

1.2 Setup Function

`Serial.begin(9600)`: Initializes serial communication at a baud rate of 9600, allowing data to be sent to the computer's serial monitor.

`pinMode(trigPin, OUTPUT)`: Sets the trigger pin as an output pin. This means that the Arduino can send a signal to the trigger pin.

`pinMode(echoPin, INPUT)`: Sets the echo pin as an input pin. This means that the Arduino can receive a signal from the echo pin.

1.3 Loop Function

The loop function executes repeatedly, constantly measuring the distance.

Sending an Ultrasonic Pulse:

`digitalWrite(trigPin, LOW)`: Sets the trigger pin low for 2 microseconds.

`delayMicroseconds(2)`: Delays for 2 microseconds.

`digitalWrite(trigPin, HIGH)`: Sets the trigger pin high for 10 microseconds.

`delayMicroseconds(10)`: Delays for 10 microseconds.

`digitalWrite(trigPin, LOW)`: Sets the trigger pin low again.

This sequence generates a short pulse on the trigger pin, which initiates the ultrasonic sensor to send a sound wave.

Receiving the Echo Pulse:

`duration = pulseIn(echoPin, HIGH)`: This line reads the duration (in microseconds) of the echo pulse received on the echo pin. The ultrasonic sensor sends a sound wave that travels to an object and reflects back. The echo pin detects this reflected sound wave, and the `pulseIn()` function measures the time it takes for the sound wave to travel to the object and back.

Calculating Distance:

`distance = duration * 0.0343 / 2`: This line calculates the distance to the object using the formula:

$$\text{distance} = \frac{\text{duration} \times \text{speed of sound}}{2}$$

The speed of sound in air is approximately 343 meters per second, which is 0.0343 centimeters per microsecond.

We divide by 2 because the measured time (duration) is for the round trip of the sound wave (from the sensor to the object and back).

Limiting Distance:

`if (distance > 150)` { `distance = 150;` }: This line checks if the calculated distance is greater than 150 cm. If it is, it sets the distance to 150 cm, as the sensor likely has a limited range.

Printing Distance to Serial Monitor:

`Serial.print("Distance:  ");`
`Serial.print(distance);`
`Serial.println(" cm");`: This prints the measured distance in centimeters to the serial monitor.

Delay:

`delay(50);`: This line delays for 50 milliseconds before the loop repeats, measuring the distance again.