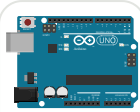


Parking assist: beep before you bump · START A BIT

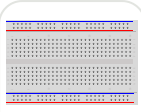


Scan to open

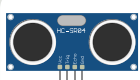
Parts you need



Arduino
UNO



Breadboard



Ultrasonic
sensor (HC-
SR04)



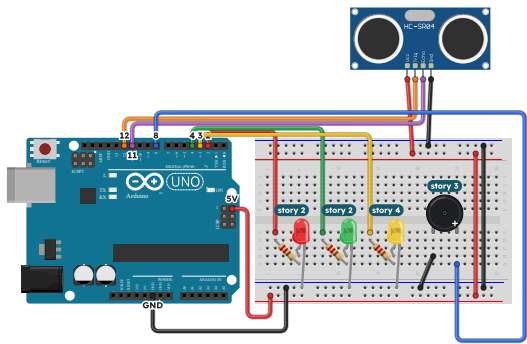
LEDs



220 Ω
resistors



Buzzer



pin 12 → Trig

pin 11 → Echo

5V → rails → VCC

GND → – rails

pin 2 → red LED

pin 4 → green LED

pin 8 → buzzer +

pin 3 → yellow LED

story 2

story 2

story 3

story 4

Remember

Ultrasonic distance sensor

distance in cm = time in μs ÷ 2 ÷ 29.1

Serial Monitor

Serial.begin(9600) in setup,
Serial.println(value) wherever you want to
see something. The monitor must be set
to 9600 too.

Arduino libraries

Install it once, include it at the top, make
an object, call its functions.

Buzzer

The leg under the + mark to a pin, the
other to GND. tone(pin, 1000) sounds
until noTone(pin).

Story 1: See the distance

```
#include <HCSR04.h>

const int triggerPin = 12;
const int echoPin = 11;
UltrasonicDistanceSensor sensor(triggerPin, echoPin);

void setup() {
  Serial.begin(9600);
}

void loop() {
  float distance = sensor.measureDistanceCm();
  Serial.print(distance);
  Serial.println(" cm");
  delay(200);
}
```

Story 2: Red when too close

```
#include <HCSR04.h>

const int triggerPin = 12;
const int echoPin = 11;
const int redLedPin = 2;
const int greenLedPin = 4;
const int tooClose = 30; // cm
UltrasonicDistanceSensor sensor(triggerPin, echoPin);

void setup() {
  Serial.begin(9600);
  pinMode(redLedPin, OUTPUT);
  pinMode(greenLedPin, OUTPUT);
}

void loop() {
  float distance = sensor.measureDistanceCm();
  Serial.println(distance);
  if (distance > 0 && distance < tooClose) {
    digitalWrite(redLedPin, HIGH);
    digitalWrite(greenLedPin, LOW);
  } else {
    digitalWrite(redLedPin, LOW);
    digitalWrite(greenLedPin, HIGH);
  }
  delay(100);
}
```

Story 3: Beep faster, closer

```
#include <HCSR04.h>

const int triggerPin = 12;
const int echoPin = 11;
const int redLedPin = 2;
const int greenLedPin = 4;
const int buzzerPin = 8;
const int tooClose = 30; // cm: from here
UltrasonicDistanceSensor sensor(triggerPin, echoPin);

void setup() {
  pinMode(redLedPin, OUTPUT);
  pinMode(greenLedPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
}

void loop() {
  float distance = sensor.measureDistanceCm();
  if (distance > 0 && distance < tooClose) {
    digitalWrite(redLedPin, HIGH);
    digitalWrite(greenLedPin, LOW);
    int gap = map((int) distance, 0, tooClose, 50, 100);
    tone(buzzerPin, 1000); // a beep of 1000 Hz
    delay(60);
    noTone(buzzerPin); // ... for 60 ms
    delay(gap);
  } else {
    digitalWrite(redLedPin, LOW);
    digitalWrite(greenLedPin, HIGH);
    delay(100);
  }
}
```

Story 4: Park like a real car

```
#include <HCSR04.h>

const int triggerPin = 12;
const int echoPin = 11;
const int redLedPin = 2;
const int yellowLedPin = 3;
const int greenLedPin = 4;
const int buzzerPin = 8;
const int slowDown = 40; // cm: yellow
const int tooClose = 20; // cm: red and
const int stopHere = 8; // cm: red and
UltrasonicDistanceSensor sensor(triggerPin, echoPin);

void setup() {
  pinMode(redLedPin, OUTPUT);
  pinMode(yellowLedPin, OUTPUT);
  pinMode(greenLedPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
}

void lights(bool red, bool yellow, bool green) {
  digitalWrite(redLedPin, red);
  digitalWrite(yellowLedPin, yellow);
  digitalWrite(greenLedPin, green);
}

void loop() {
  float distance = sensor.measureDistanceCm();
  if (distance < 0 || distance >= slowDown) {
    lights(false, false, true);
    noTone(buzzerPin);
    delay(100);
  } else if (distance >= tooClose) {
    lights(false, true, false);
    noTone(buzzerPin);
    delay(100);
  } else if (distance >= stopHere) {
    lights(true, false, false);
    int gap = map((int) distance, stopHere, tooClose, 100, 50);
    tone(buzzerPin, 1000);
    delay(60);
    noTone(buzzerPin);
    delay(gap);
  } else {
    lights(true, false, false);
    tone(buzzerPin, 1500);
    delay(100);
  }
}
```