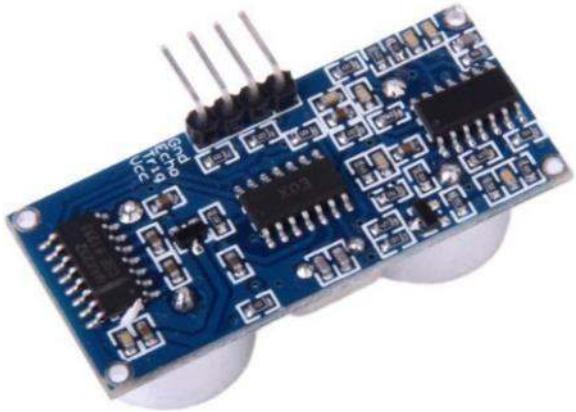
Arduino Coding – HC-SR04 Distance Sensors

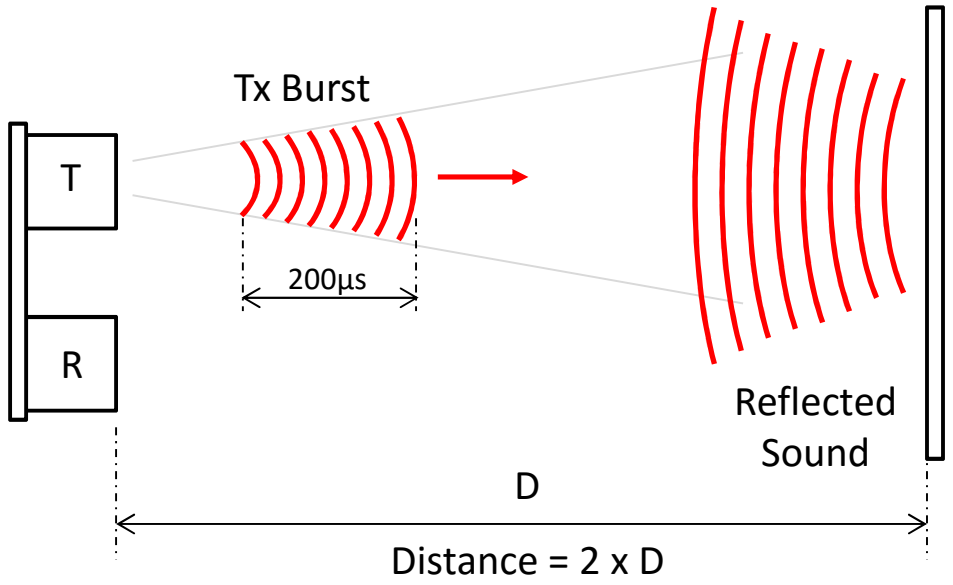
Contents:

- Sensor Features
- Sensor Connections
- Sample Sketch
- Questions

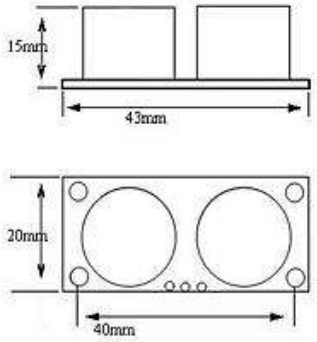


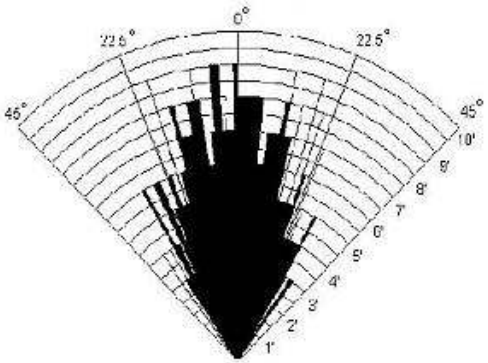
Sensor Features

- Low cost < £2.00
- Low power 2mA static current
- 8 cycle 40 kHz sound burst
- Detection range 2cm – 450cm
- Sensor angle 15 degrees
- High precision +/- 3mm
- Easy to use?



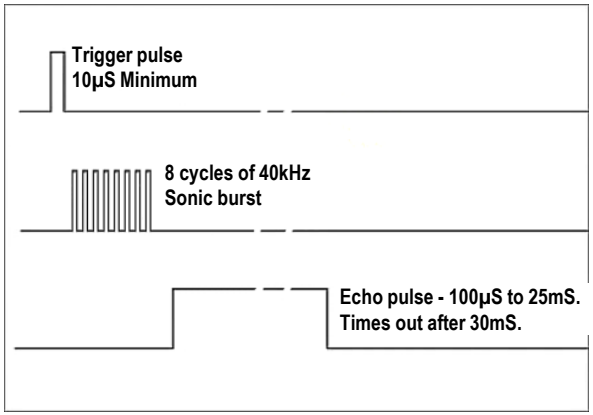
- Provide 10µs HIGH trigger pulse
- Measure width of returned 'Echo' pulse





Practical test of performance,
Best in 30 degree angle

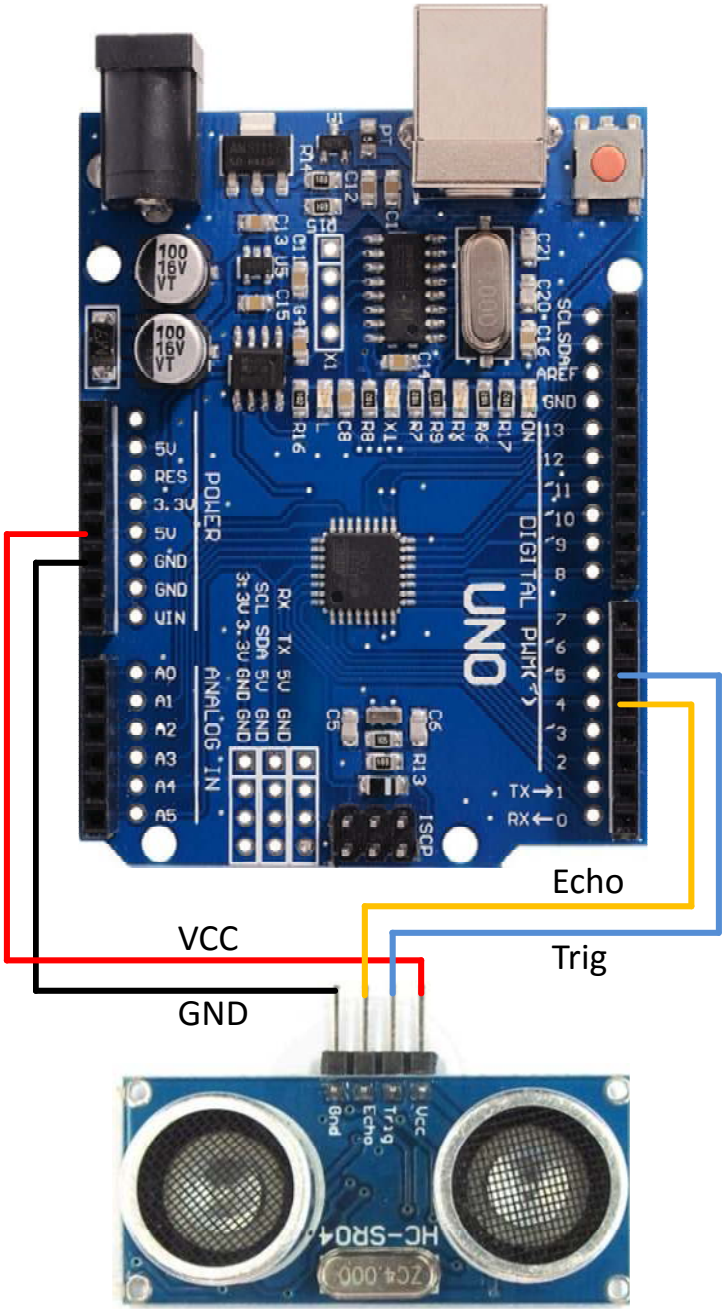
Speed of sound:	340 m/s
Distance	= speed x Time / 2
Distance in cm	= echo pulse width in µS / 58
Distance in inches	= echo pulse width in µS / 148
Range:	2cm min – 4.5m max (t = 13ms).



Sensor Connections

- Wipe Arduino with ‘New Sketch’ code
- **REMOVE** power before wiring
- Connect GND to Gnd
- Connect 5v to Vcc
- Connect Pin 5 to Trig
- Connect Pin 4 to Echo

Low power so use with USB power.



Sample Sketch

SR04_Ping_Example

```
// ################################################################################
//
//  SR04 Ping Example v0.1 Beta
//
//  Released:  15/06/2015
//
// ################################################################################
/*
  Test HC-SR04 distance sensor by returning distance to a PC screen.

  Trigger distance sensor.
  Wait for returned 'echo' pulse
  Calculate distance measured by echo pulse
  If within limits send distance and turn OFF LED
  If outside limits send -1 and turn ON LED
  Wait 50ms between measurements
 */

#define echoPin 4 // Echo Pin
#define trigPin 5 // Trigger Pin
#define LEDPin 13 // Onboard LED

int maximumRange = 200; // Maximum range needed
int minimumRange = 0; // Minimum range needed
long duration; // Duration used to calculate distance
long distance; // Distance calculated

void setup() {
 Serial.begin (9600);
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);
 pinMode(LEDPin, OUTPUT); // Use LED indicator (if required)
}
```

```
void loop() {
/* The following trigPin/echoPin cycle is used to determine the
 distance of the nearest object by bouncing soundwaves off of it. */
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);

 digitalWrite(trigPin, HIGH);
 delayMicroseconds(10);
 digitalWrite(trigPin, LOW);
 duration = pulseIn(echoPin, HIGH);

 //Calculate the distance (in cm) based on the speed of sound.
 distance = duration/58.2;

 if (distance >= maximumRange || distance <= minimumRange){
  // Send a negative number to computer and Turn LED ON
  // to indicate "out of range"
  Serial.println("-1");
  digitalWrite(LEDPin, HIGH);
 }
 else {
  // Send the distance to the computer using Serial protocol, and
  // turn LED OFF to indicate successful reading.
  Serial.println(distance);
  digitalWrite(LEDPin, LOW);
 }

  //Delay 50ms before next reading.
  delay(50);
}
```

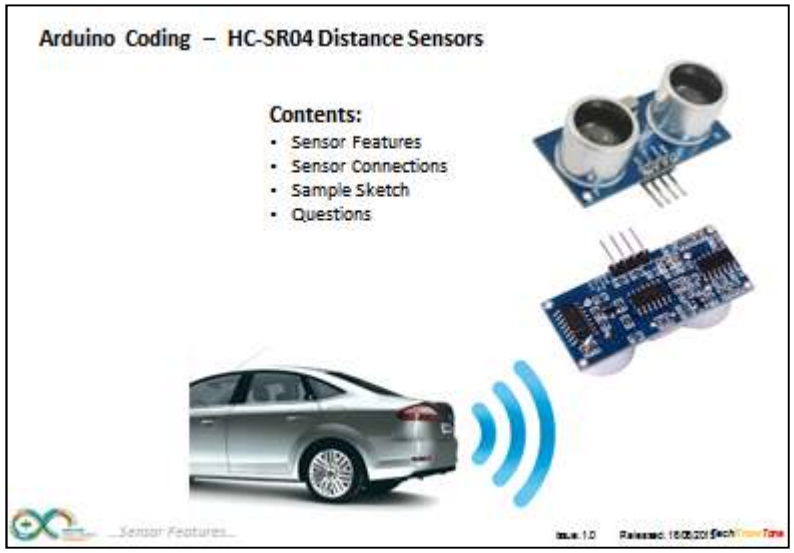
Note:

- Allow 50 ms between trigger pulses
- Avoids false echoes

Questions

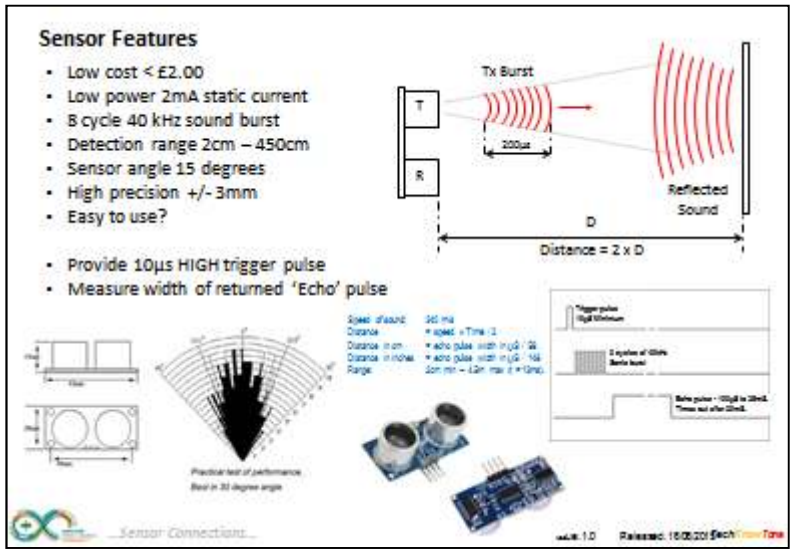
?

Teaching Notes:



- Who has been in a car with parking sensors?
- Can you describe what they do?
- Here we will explain how distance sensors work
- We will write code in an Arduino sketch

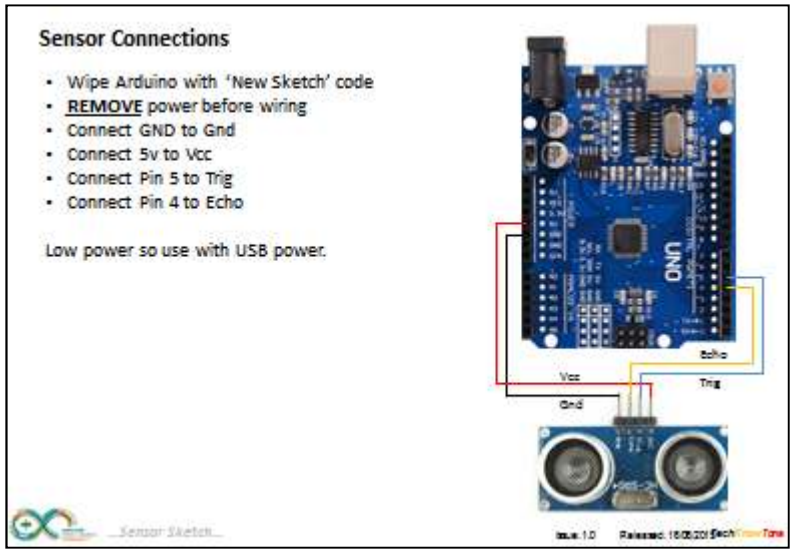
- Can anyone tell me how a bat sees in the dark?
- Yes, it uses sound echoes from the ‘clicks’ it makes.
- Parking sensors use bursts of sound waves too.



- The sensor has a sound transmitter and a sound receiver.
- It sends out a burst of sound at frequencies higher than you can hear.
- The burst of sound bounces of distance objects and is reflected back.
- As sound travels at a constant speed, time taken == distance travelled.
- As it receives an echo, distance travelled is 2 x distance to the object.
- In use we simply trigger the sensor by providing it with a logic ‘1’ pulse.
- In code we then wait for the ‘echo’ response pulse.
- We then ½ the time taken to calculate the distance to the object.

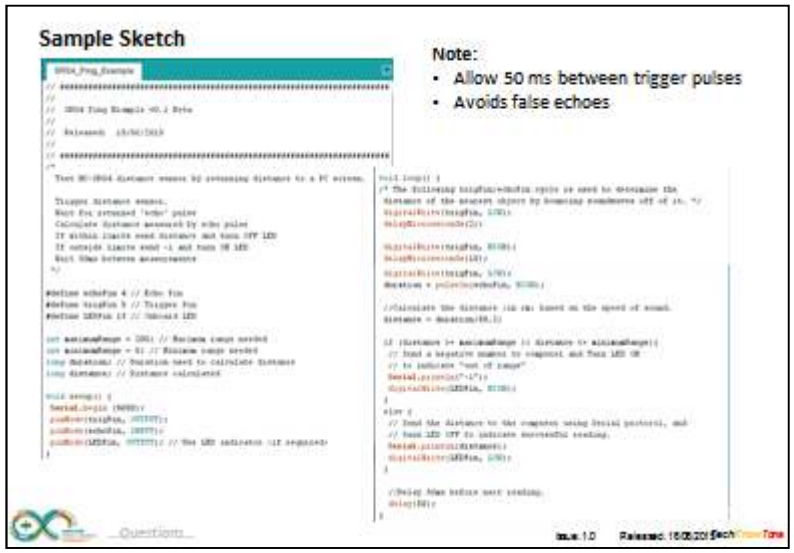
- Note that the sound travels outwards in the form of a cone.
- There may be multiple echoes depending on what is in front of it.
- The reflected sound energy is greater for larger objects.
- Round or angled objects, like chair legs, may not reflect enough sound.

Teaching Notes:



- This is the connection diagram for an Arduino UNO.
- As the sensor is low power you can run it off the USB connection.
- If your UNO has been used before, wipe it with a 'New' sketch first.
- Always remove USB power when making/changing connections.
- Incorrect connections can damage the sensor or the UNO.
- Always double check your wiring before reconnecting USB power.
- If working with someone it's best to ask them to check it too.

- With the USB connection made we can now write and load the sketch.



- In the sketch we define the pins we have used and variables.
- Variables must be declared before they are used in your code.
- The 'type' of variable is declared as part of this definition.
- We are using the serial monitor to display the distance values.
- See how we convert the echo pulse (time) into distance (cm).
  - The speed of sound in air is 340 m/s.
  - Sound will travel 1 cm in 29 µs.
  - With an echo, Distance = Distance to object / 2
  - Hence distance to object = Pulse time / 58
- To ignore multiple echoes we wait 50ms between triggers.

ARDUINO OPEN-SOURCE COMMUNITY

...End...