Child In Car Reminder

Aim

The purpose of this invention is to help parents remember that their child is in the car, to avoid children being left in cars, which could lead to death.

Introduction

The idea of this project came from when I was watching the news one time, and there was an accident where a child's dad had forgotten his 3 year old son in the car on a 35-degree day. As a result, his son has died from the heat [1]. Each year in Australia, over 5000 children are rescued from an unattended car. According to statistics, Ambulance Victoria responded to callouts for people locked in cars across Victoria between 1st September 2017 to 31st August 2018. Most of these cases involve toddlers and babies [2].

I thought it would be helpful to create an invention that could remind drivers when their child is still in the car as soon as they leave the driver's seat. A way to do this is to install a sensor on the child car seat to detect when the child is sitting on it, another sensor to detect whether the driver is in the car seat, and then connect it to an alarm to the front of the car where the driver can be alerted when they leave the car while child is still in the seat.
While there are similar reminders in the market, such as the Clever Elly that work as a generic reminder in the car [3], this is an invention that will actually detect when the child is in the car seat and also detect if the driver is in the car or not.

Instructions

1. Place the Arduino and power bank somewhere safe at the front of the car where it cannot be exposed to sunlight
2. Stick the ultrasonic sensor on the car ceiling above the driver's seat, facing down (don't cover the sensor, otherwise the buzzer won't go off)
3. Place the padded pressure sensor underneath the child car seat cover
4. Hide the wires to the ceiling of the car or underneath the carpet of the car
5. Power on the power bank

Materials and Equipment

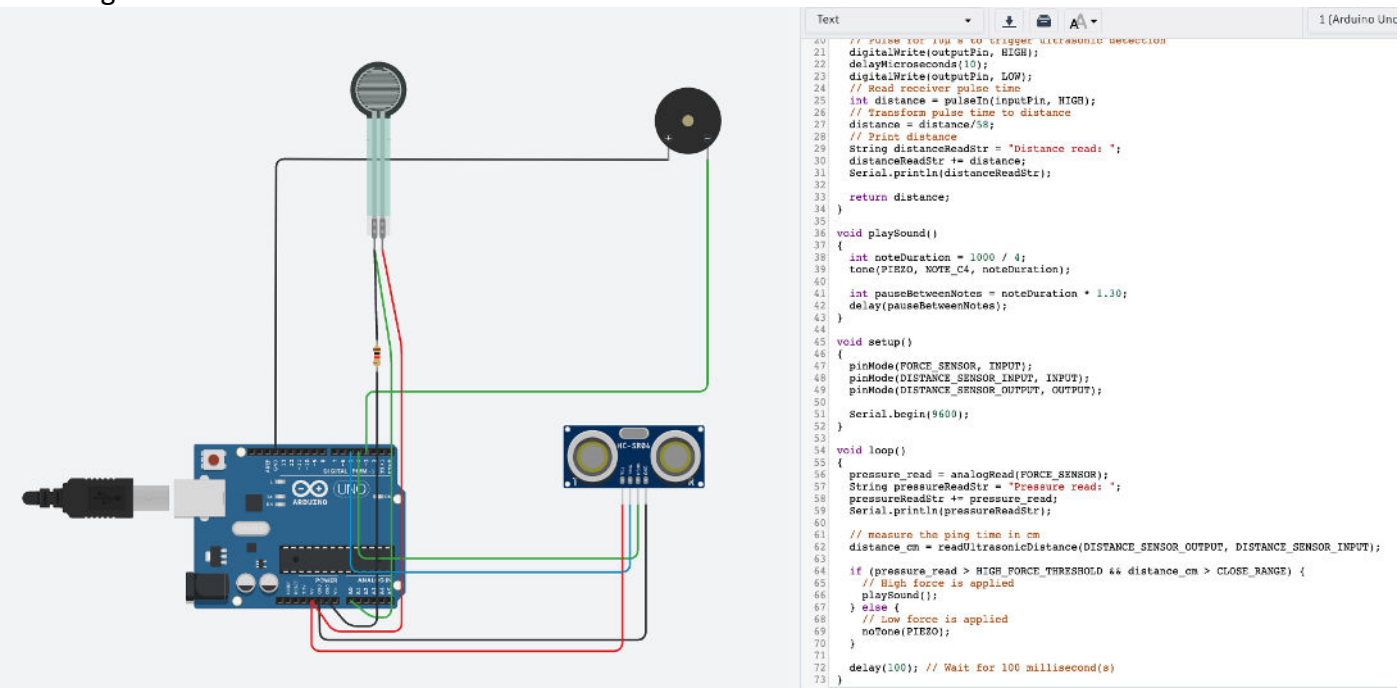
- Arduino
- Colour coded wires
- Pressure sensor
- Ultrasonic distance sensor
- Piezo (buzzer)
- Power bank
- USB cord
- Computer that runs Arduino IDE to compile and upload C++ code to the Arduino device

Design brief

Building process

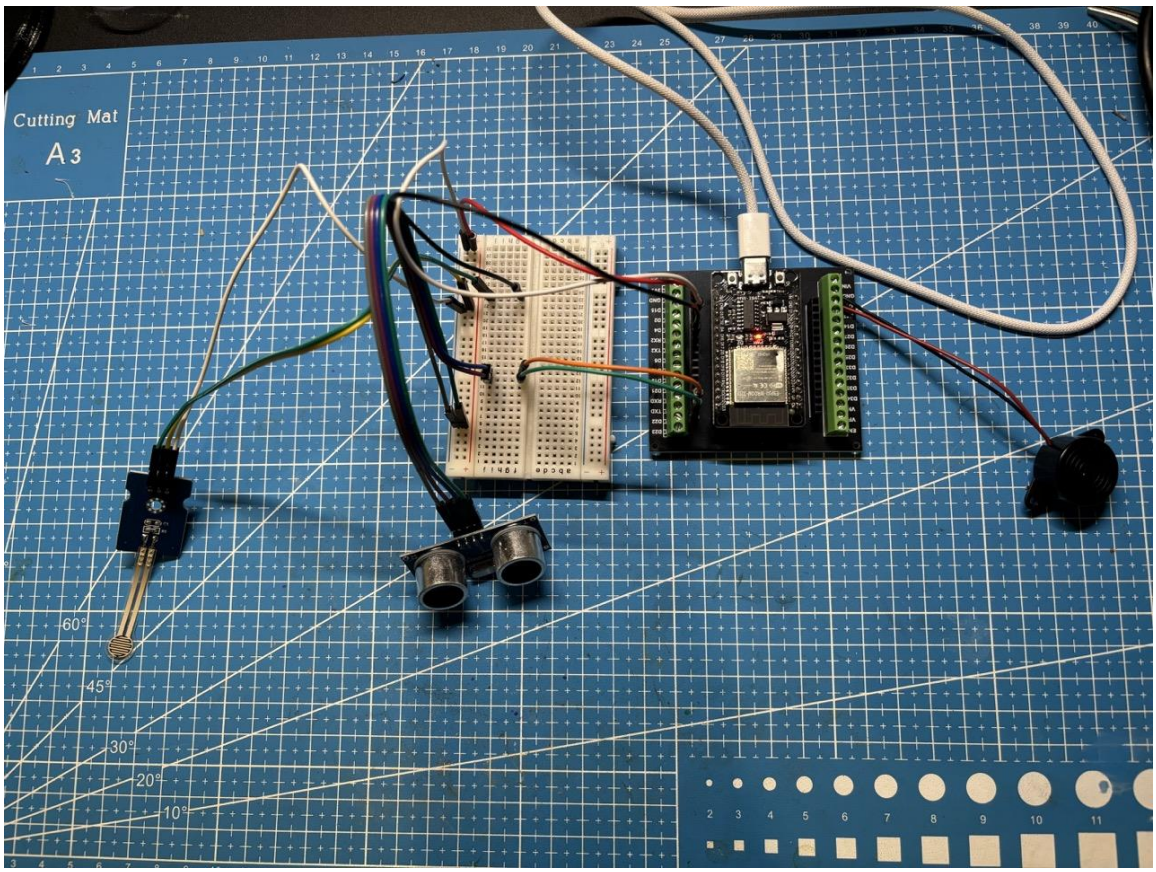
1. First, we need to figure out what components we need and what we need to detect, and then research into what devices and sensors we can use. We need to:
   - Detect if child is sitting in the seat – weight sensor, pressure sensor
     We decided to use force sensor, because weight sensor needs a flat surface to work and can be quite chunky to sit on
   - Detect if driver is sitting in the car – infrared sensor, ultrasonic distance sensor
     We decided to use ultrasonic distance sensor, because infrared sensor can be disrupted by sunlight shining into the car
   - Alert driver when leaving the car while child is still in seat – Piezo buzzer

2. We then used Tinkercad.com to build a virtual prototype of the device with some sample circuit designs.

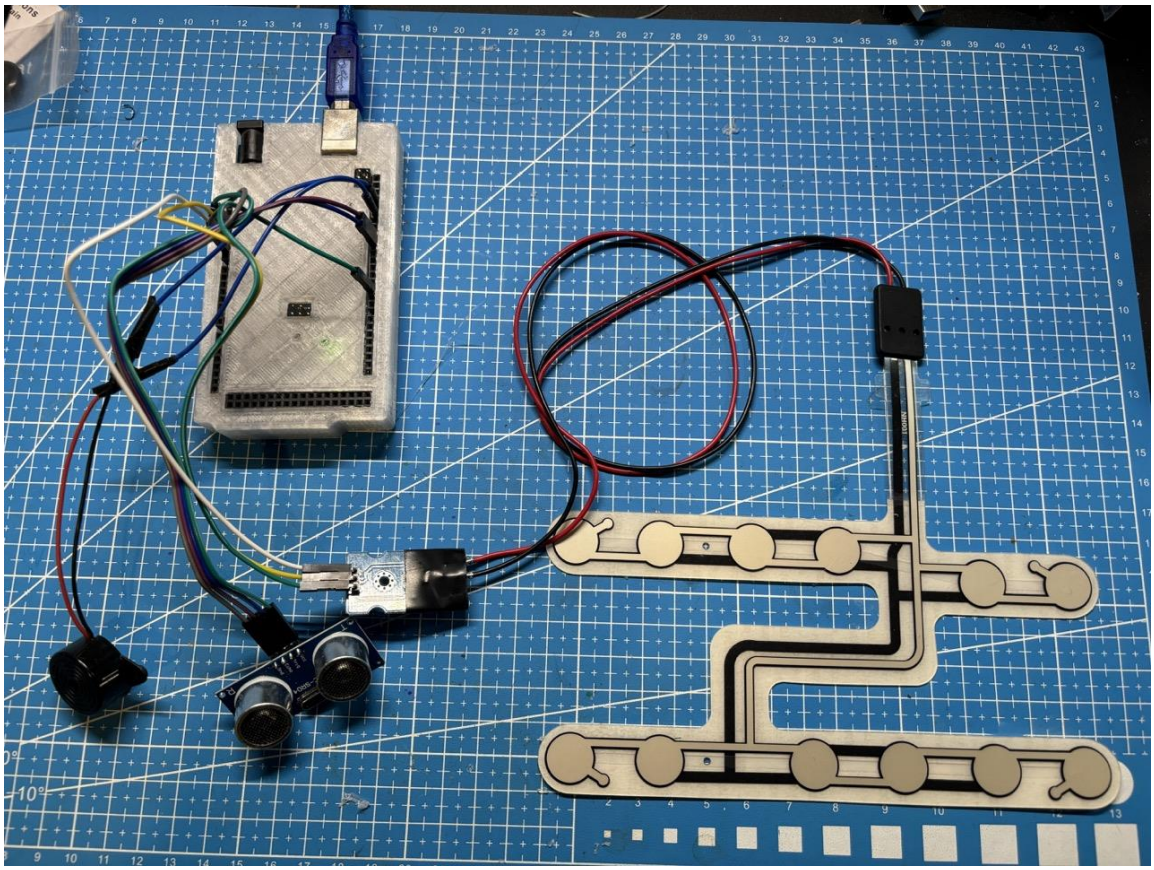


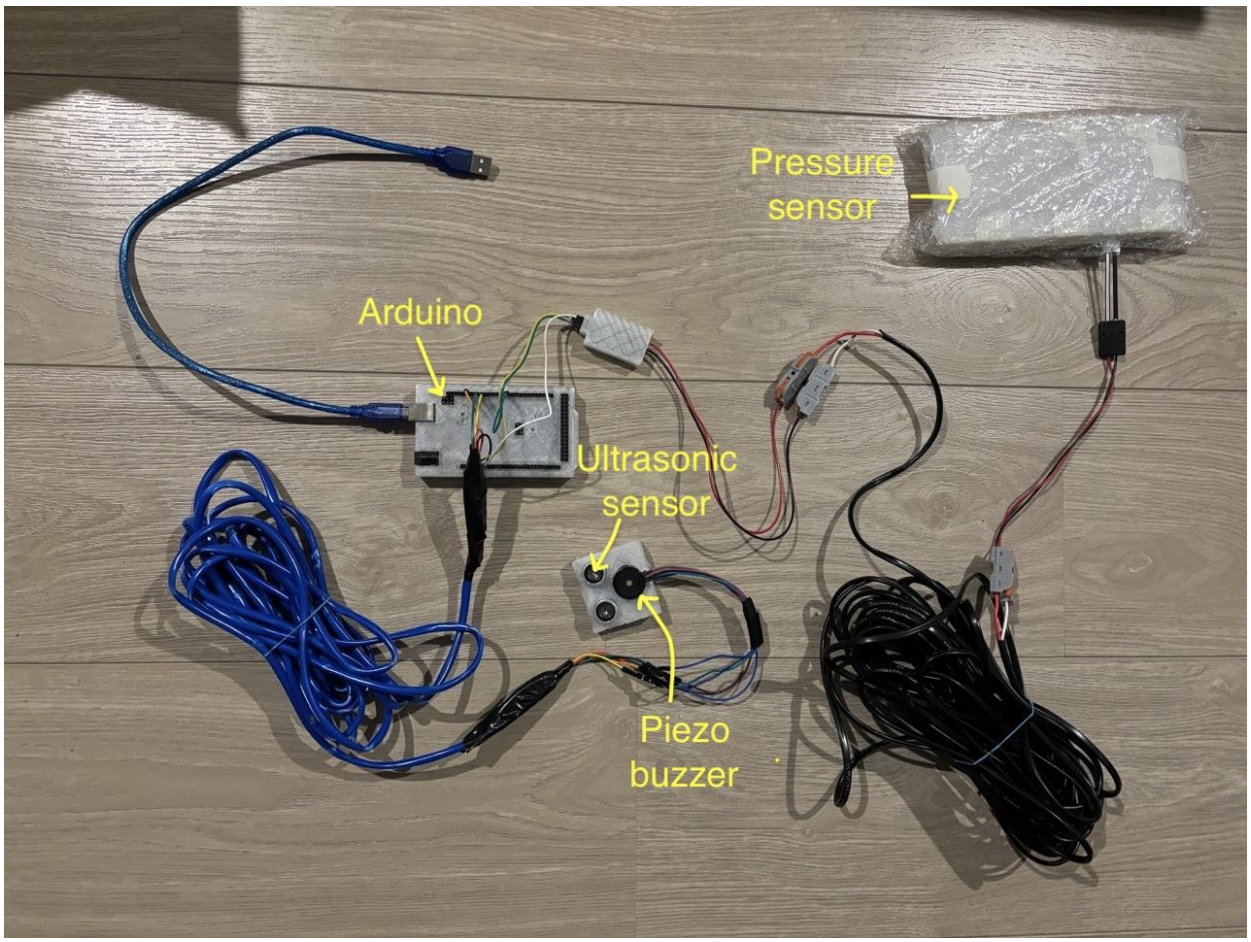
```
  digitalWrite(outputPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(outputPin, LOW);
  // Read receiver pulse time
  int distance = pulseIn(inputPin, HIGH);
  // Transform pulse time to distance
  distance = distance/58;
  // Print distance
  String distanceReadStr = "Distance read: ";
  distanceReadStr += distance;
  Serial.println(distanceReadStr);

  return distance;
}

void playSound()
{
  int noteDuration = 1000 / 4;
  tone(PIEZO, NOTE_C4, noteDuration);

  int pauseBetweenNotes = noteDuration * 1.30;
  delay(pauseBetweenNotes);
}

void setup()
{
  pinMode(FORCE_SENSOR, INPUT);
  pinMode(DISTANCE_SENSOR_INPUT, INPUT);
  pinMode(DISTANCE_SENSOR_OUTPUT, OUTPUT);

  Serial.begin(9600);
}

void loop()
{
  pressure_read = analogRead(FORCE_SENSOR);
  String pressureReadStr = "Pressure read: ";
  pressureReadStr += pressure_read;
  Serial.println(pressureReadStr);

  // measure the ping time in cm
  distance_cm = readUltrasonicDistance(DISTANCE_SENSOR_OUTPUT, DISTANCE_SENSOR_INPUT);

  if (pressure_read > HIGH_FORCE_THRESHOLD && distance_cm > CLOSE_RANGE) {
    // High force is applied
    playSound();
  } else {
    // Low force is applied
    noTone(PIEZO);
  }

  delay(100); // Wait for 100 millisecond(s)
}
```

3. Our next step is to build the physical prototype. We bought all the sensors we need and connected them together with wires and a breadboard, then uploaded the code to the Arduino. We tested it and this was working as expected.
   There is a slight problem, the buzzer started alerting as soon as pressure is placed on the pressure sensor. To avoid alerting before the driver sits in the car, we changed the code in the Arduino to only sound the buzzer after the ultrasonic sensor detects an object in close range and then leaves.



4. To further improve the prototype, we bought another type of pressure sensor that has multiple detection points and use a smaller Arduino Mega and 3D printed cases for vulnerable parts, then change to longer cables to be able to install in a real car.



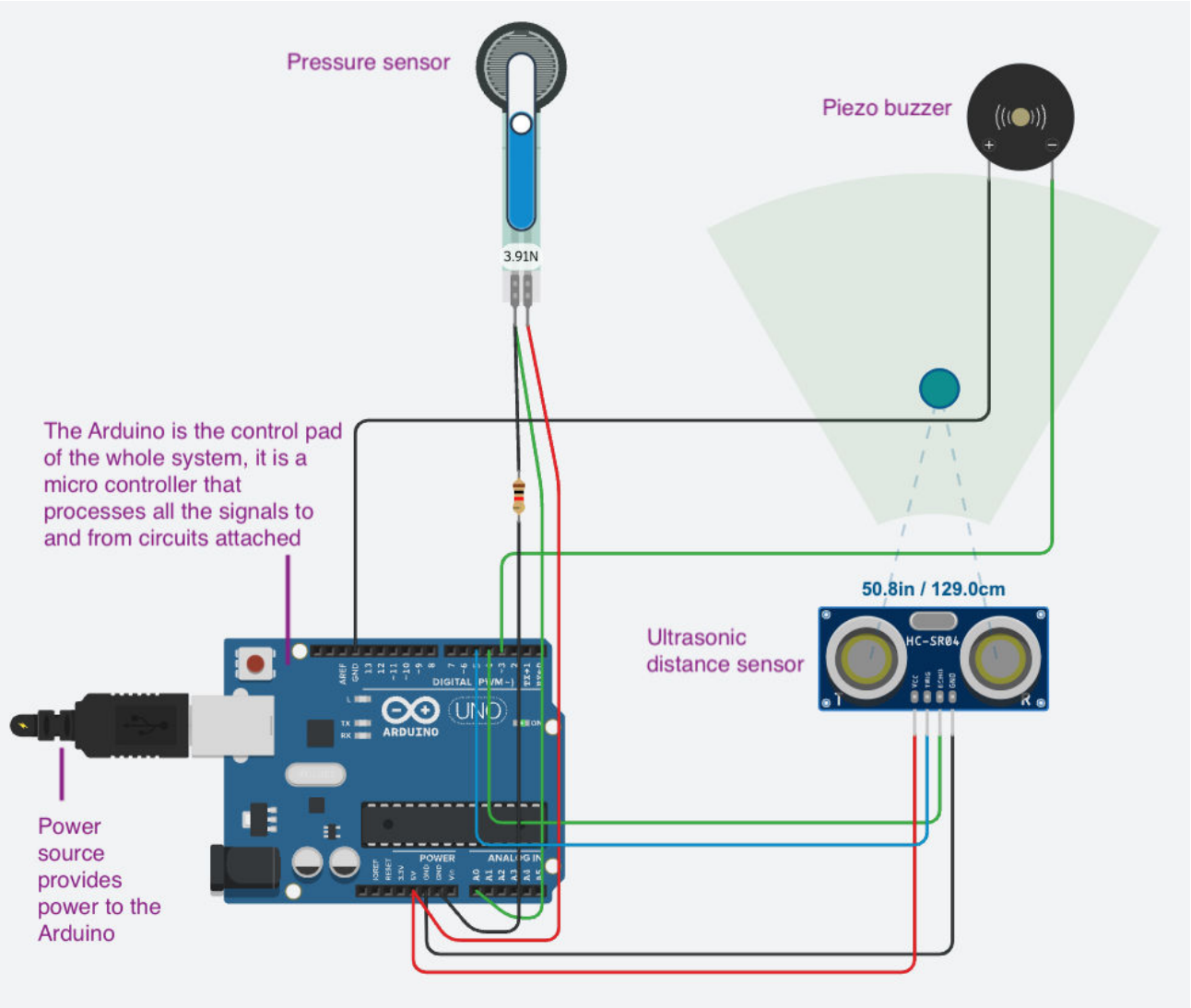


5. Finally, we were ready to install it to the car!
   This prototype is programmed to sound the buzzer when all conditions are met:
   - the pressure sensor detects pressure being applied,
   - the ultrasonic sensor detects object within close range,
   - and then does not detect object within close range

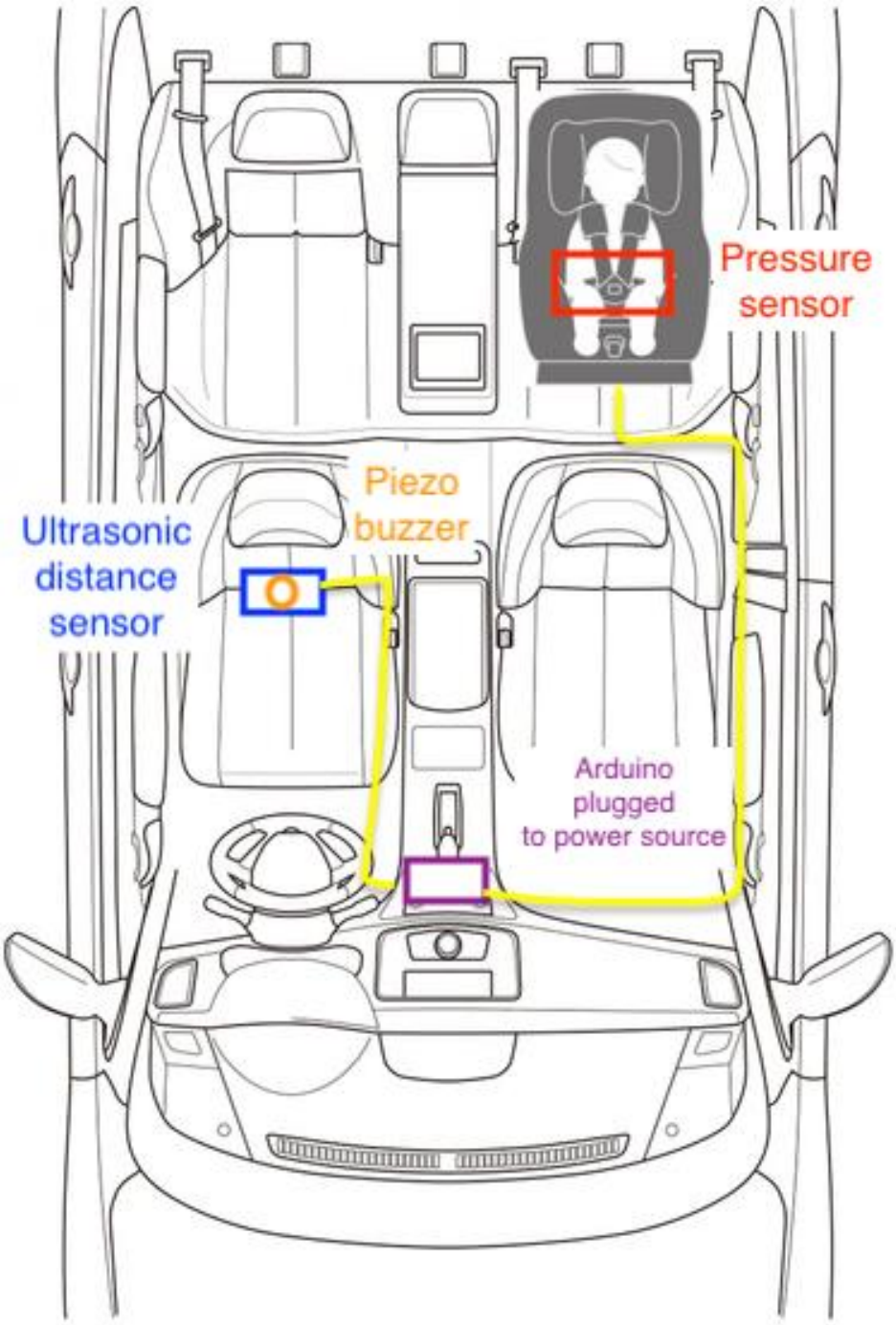
   The buzzer will stop when either condition is met:
   - the ultrasonic sensor detects object within close range
   - the pressure sensor does not detect pressure being applied

Design

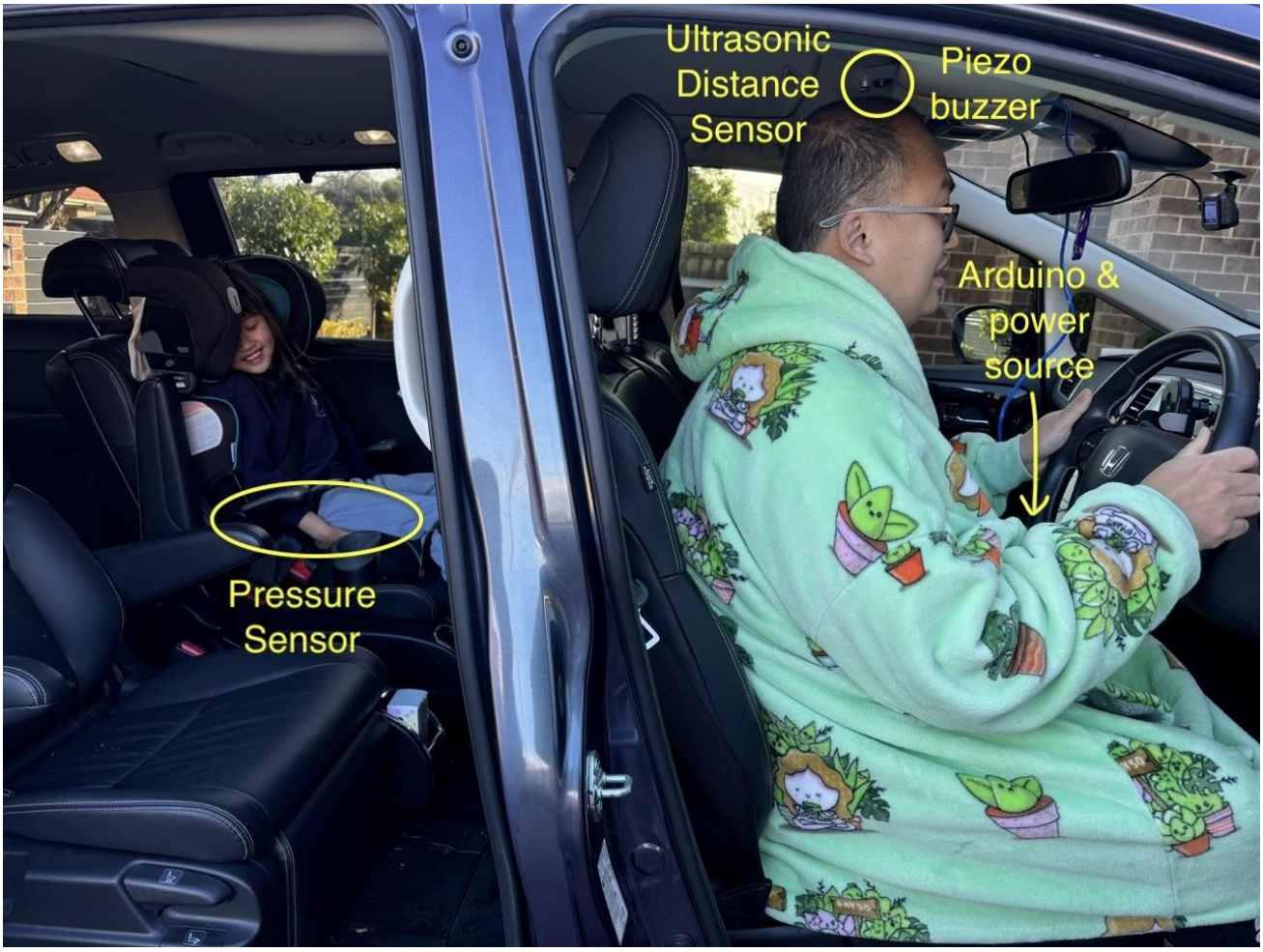
Circuit Design



System Design



Final Product


Ultrasonic
Distance
Sensor
Piezo
buzzer
Arduino &
power
source
Pressure
Sensor

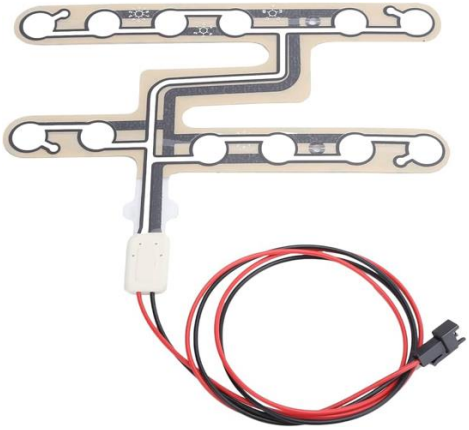
Discussion

The science behind

The science behind each component connected to the Arduino:

- Pressure sensor

  Pressure sensors work when pressure is applied to the surface, this pressure causes the upper and lower circuit layers of the sensor to touch each other, conducting an electrical circuit to generate a signal to the Arduino. [4]



- Ultrasonic Distance sensor

  Ultrasonic sensors work by sending ultrasonic waves in a certain direction. When the waves hit an object, they bounce off moving towards the sensor. The wave is received from the object and the time it took to reach the receiver is measured and calculated telling the Arduino how far the object is. [5]

- Piezo buzzer

  The piezo buzzer works by creating sounds from the flexing and vibrating of a membrane inside the cylinder. This is initiated by the Arduino, which gives a PWM (Pulse Width Modulation) signal to the buzzer, the frequency then converts into sound waves by the membrane. [6]

Field test observations

Scenario 1	Expected Outcome	Actual Outcome
1. Child sits in car seat	No alarm	✓
2. Driver sits in car seat	No alarm	✓
3. Driver leaves car seat	Sound alarm	✓
4. Child leaves car seat	No alarm	✓
5. Repeat 1 - 4	Work same as above	✓

Scenario 2	Expected Outcome	Actual Outcome
1. Child sits in car seat	No alarm	✓
2. Driver sits in car seat	No alarm	✓
3. Child leaves car seat	No alarm	✓
4. Driver leaves car seat	No alarm	✓

Scenario 3	Expected Outcome	Actual Outcome
1. Child sits in car seat	No alarm	✓
2. Driver sits in car seat	No alarm	✓
3. Driver leaves car seat	Sound alarm	✓
4. Driver sits in car seat	No alarm	✓
5. Driver leaves car seat	Sound alarm	✓
6. Child leaves car seat	No alarm	✓

Scenario 4	Expected Outcome	Actual Outcome
1. Driver sits in car seat	No alarm	✓
2. Driver leaves car seat	No alarm	✓

Scenario 5	Expected Outcome	Actual Outcome
1. Driver sits in car seat	No alarm	✓
2. Child sits in car seat	No alarm	✓
3. Driver leaves car seat	Sound alarm	✓
4. Child leaves car seat	No alarm	✓

What we learned

- During testing, we found the buzzer going off randomly when both child and driver are both sitting in the car. We improved it by adding some threshold in the distance reading
- We realised that we cannot rely on the car power in a real-life scenario, because the driver will turn off the car before leaving, so the device won't have power to sound the buzzer when leaving the car

What we can improve on

- We could use a rechargeable backup battery that can be charged from the car for more efficiency
- We could use a wireless system for easier install and safer with children

Acknowledgements and References

Acknowledgements

- Mum – helped put this report together and coding the Arduino.
- Dad – helped build the device together and coding the Arduino.
- My sister – helped with testing and acting for the presentation video.
- Skye Paice (Science teacher) – helped giving advice on the project.

References

1. https://www.kidspot.com.au/news/father-of-boy-3-who-died-in-a-hot-car-reveals-how-the-tragedy-occurred/news-story/4eedcab37d8daf5c2944f06bde78daed (Accessed: 09-06-2024, 10:00am)
2. https://www.kidsafevic.com.au/road-safety/hot-cars/#:~:text=Every%20year%20in%20Australia%2C%20over,cases%20involving%20toddlers%20and%20babies (Accessed: 09-06-2024, 10:20am)
3. https://www.kidsafevic.com.au/shop/3798/ (Accessed: 09-06-2024, 11:00am)
4. https://www.jcftechnology.com/products_detail.php?menuid=5&id=48 (Accessed: 16-06-2024, 10:10am)
5. https://wiki.keyestudio.com/KS0328_Keyestudio_HR-SR04_Blue_Ultrasonic_Module (Accessed: 16-06-2024, 10:00am)
6. https://www.youtube.com/watch?v=kqv-aan4pBU (Accessed: 16-06-2024, 10:30am)

Appendix

Arduino code (in C++)

```
#include "pitches.h"

#define BUZZER_PIN 12
#define LED_PIN 13 //pin 13 to activate the LED when the baby sits on the pressure sensor
#define PRESSURE_SENSOR_PIN 2
#define DISTANCE_SENSOR_TRIGGER 8
#define DISTANCE_SENSOR_ECHO 9

const int CLOSE_DISTANCE_MM = 100;

// variables will change:
int pressureSensorState = 0;  //baby is on - 0, baby is off - 1
float detectedDistanceMM;
bool isSensorActivated = false;
int activeCounter = 0;

float readDistance(int triggerOutput, int echoInput) {
  digitalWrite(triggerOutput, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerOutput, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerOutput, LOW);
  int duration = pulseIn(echoInput, HIGH);

  return (duration*.0343)/2; // Returns distance in mm
}

void playBuzzer() {
  delay(500);
  for (int thisNote = 0; thisNote < 4; thisNote++) {
    // to calculate the note duration, take one second divided by the note type.
    //e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
    int noteDuration = 1000 / 4;
    tone(BUZZER_PIN, NOTE_C4, noteDuration);

    // to distinguish the notes, set a minimum time between them.
    // the note's duration + 30% seems to work well:
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
  }
  // stop the tone playing:
  noTone(BUZZER_PIN);
}

void setup() {
  // initialize the LED pin as an output:
  pinMode(LED_PIN, OUTPUT);
  // initialize the pushbutton pin as an input:
```

```
  pinMode(PRESSURE_SENSOR_PIN, INPUT);
  pinMode(DISTANCE_SENSOR_TRIGGER, OUTPUT);
  pinMode(DISTANCE_SENSOR_ECHO, INPUT);
  Serial.begin(9600);
}

void loop() {
  // read the state of the pressure value:
  pressureSensorState = digitalRead(PRESSURE_SENSOR_PIN);
  detectedDistanceMM = readDistance(DISTANCE_SENSOR_TRIGGER, DISTANCE_SENSOR_ECHO);
  int personInRange = detectedDistanceMM <= CLOSE_DISTANCE_MM;
  int personNotInRange = detectedDistanceMM > CLOSE_DISTANCE_MM;

  // check if the baby sit on the seat. If it is, the pressureSensorState is HIGH(1):
  if (pressureSensorState == HIGH) {
    // turn LED on:
    digitalWrite(LED_PIN, HIGH);

    if (personInRange)
    {
      isSensorActivated = true;
      activeCounter = 0;
    }

    if (personNotInRange && isSensorActivated == true)
    {
      activeCounter++;
      delay(200);
      if (activeCounter >= 5) // object not found in close range consistently 5 times
      {
        playBuzzer();
      }
    }
  } else {
    activeCounter = 0;
    // turn LED off:
    digitalWrite(LED_PIN, LOW);
    if (personNotInRange)
    {
      isSensorActivated = false;
    }
  }

  Serial.print("Distance: ");
  Serial.println(detectedDistanceMM);
  delay(100);
}
```