Fire fighting obstacle avoidance robot

Background :

This project was completed as a part of a college course and is a prototype of a obstacle avoidance fire fighting robot which performs the following functions

- Obstacle avoidance
- Human identification and alarm
- Fire sensing and extinguishing

What changes can be made to better the prototype

The entire project can be completed in either a raspberry pi or an arduino. My rasberry pi malfunctioned and therefore I had to do half the project on an uno half on a pi. The following changes are recommended

- Use a flame sensor instead of a MQ2. It wasn't very accurate
- Use 2 pumps and 2 sensors one on each side
- Use a wifi module to make it wireless if you're using an UNO
- Add a servo to the ultrasonic sensor so it could scan atleast 180 degrees around itself rather than just straight ahead
- Use tensorflow's autodetection which detects all objects for more accurate results
- You can add a messaging feature which messages the fire department of the fire

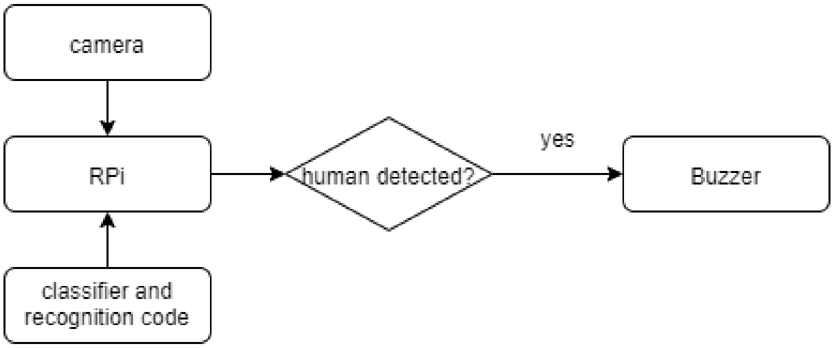
Components Used

1	Web camera
2	Buzzer
3	Raspberry pi
4	Python IDE and open CV

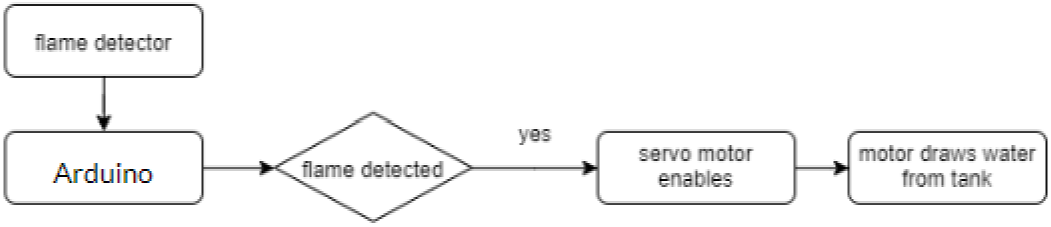
5	Flame sensor
6	5 V pump
7	DC motors
8	Breadboard and Wires
9	9 V batteries
10	Chassis
11	Water tank
12	L293D Motor Drive
13	HCSR04 Ultrasonic Sensor
14	Arduino Uno
15	Relay

Electrical System Flow Diagrams

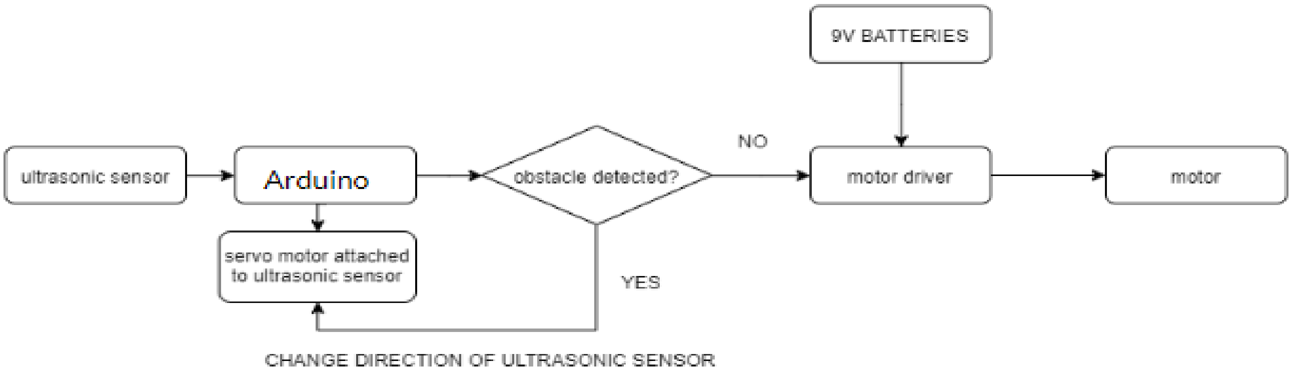
Subsystem 1: Human Recognition



Subsystem 2: Fire Extinguishing



Subsystem 3: Navigation


9V BATTERIES
ultrasonic sensor
Arduino
obstacle detected?
NO
motor driver
motor
servo motor attached to ultrasonic sensor
YES
CHANGE DIRECTION OF ULTRASONIC SENSOR

Components Used

1. Raspberry Pi

2. HCSR04 Ultrasonic Sensor

3. L293D Motor Drive Module

4. 9V 100RPM 37mm GEARED DC Motor

.

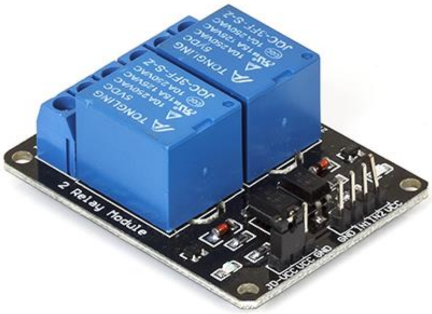
5. Arduino Uno

.

6. MQ2 Gas Sensor

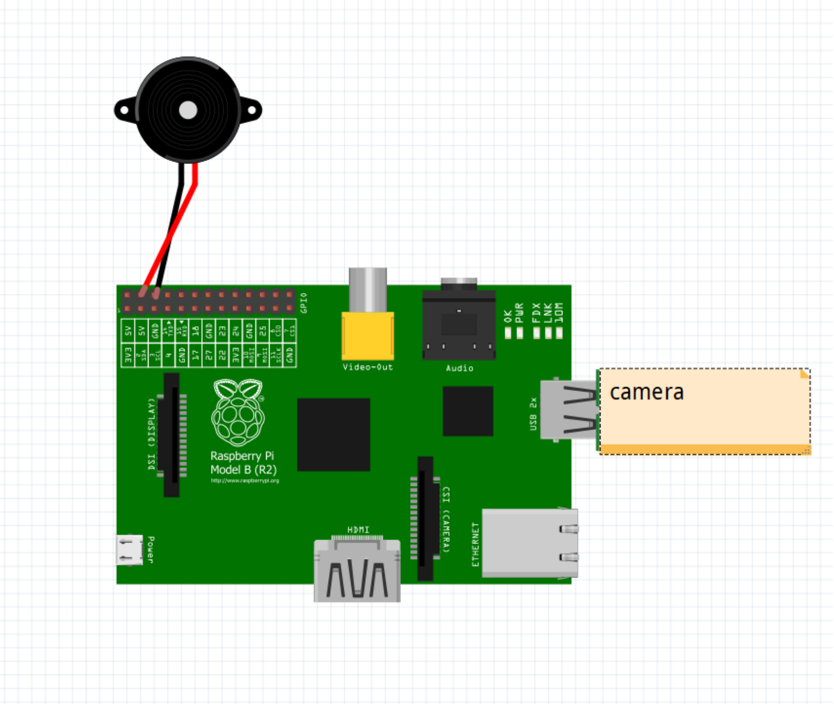
7. 5 V Pump

8. Relay



Electronic Subsystems and Programming

Subsystem 1: camera and buzzer



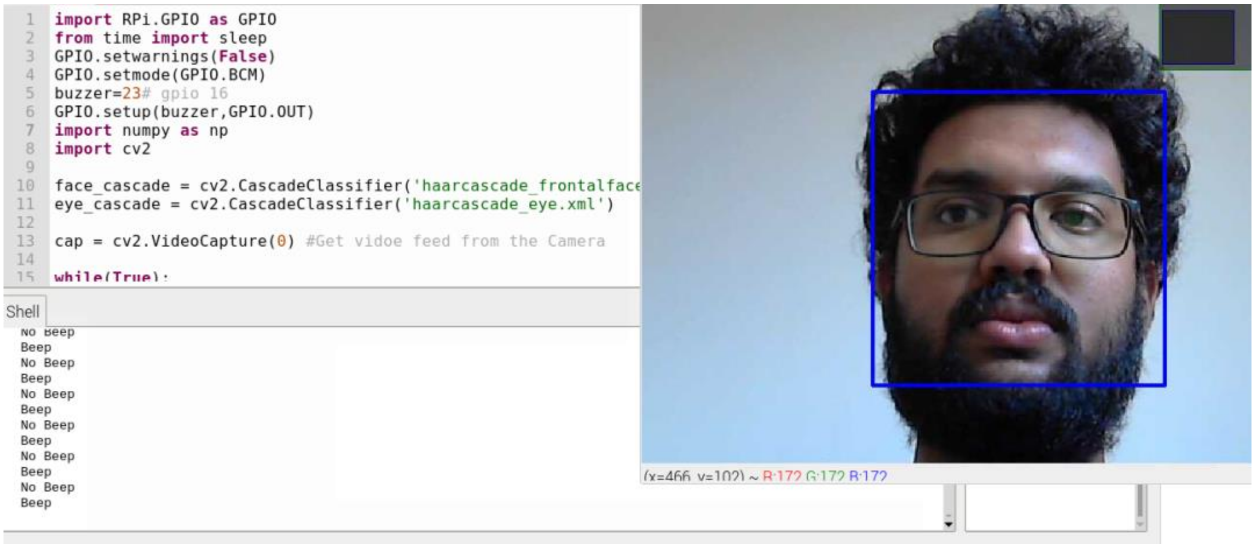
Pin connections

Buzzer pin	**Pin 16**
Buzzer ground	**Pin 6**
Camera	**USB port of raspberry-pi**

Code:

```
import RPi.GPIO as GPIO

from time import sleep

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

buzzer=23 # gpio 16

GPIO.setup(buzzer,GPIO.OUT)

import numpy as np

import cv2


face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')


cap = cv2.VideoCapture(0) #Get video feed from the Camera


while(True):

    ret, img = cap.read() # Break video into frames

    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    for (x,y,w,h) in faces:

        img = cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)

        roi_gray = gray[y:y+h, x:x+w]

        roi_color = img[y:y+h, x:x+w]

        GPIO.output(buzzer,GPIO.HIGH)

        print ("Beep")

        sleep(2)

        GPIO.output(buzzer,GPIO.LOW)

        print ("No Beep")

        sleep(0.5)

    cv2.imshow('camera',img)

    k = cv2.waitKey(10) & 0xff # Press 'ESC' for exiting video

    if k == 27:

        break

print("\n [INFO] Exiting Program and cleanup stuff")

cap.release()

cv2.destroyAllWindows()
```

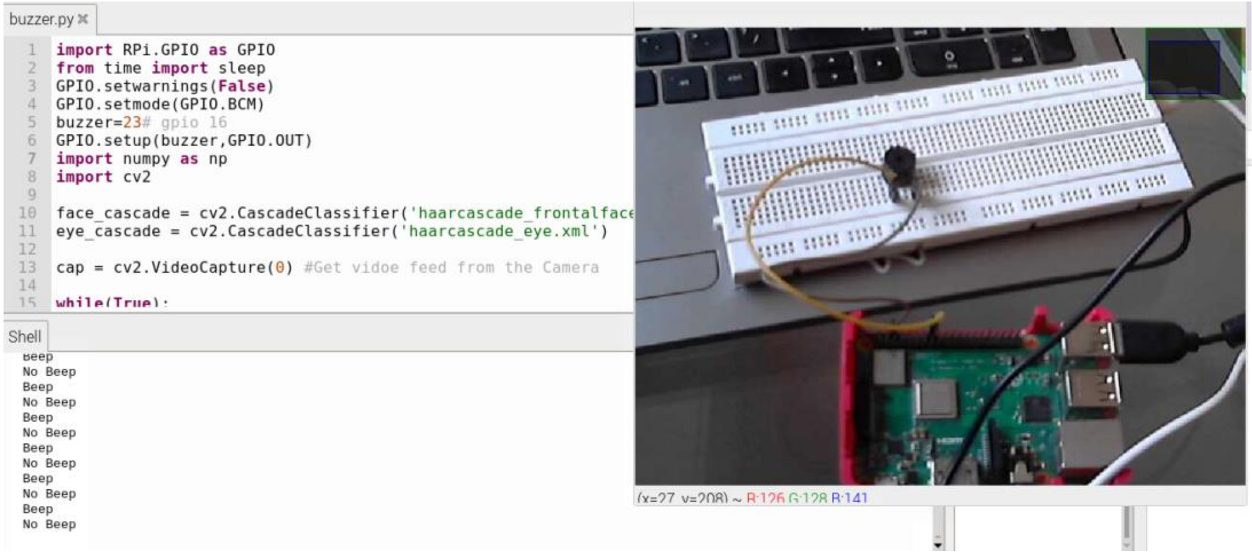
Output:



```
import RPi.GPIO as GPIO
from time import sleep
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
buzzer=23# gpio 16
GPIO.setup(buzzer,GPIO.OUT)
import numpy as np
import cv2

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

cap = cv2.VideoCapture(0) #Get vidoe feed from the Camera

while(True):
```

```
No Beep
Beep
No Beep
Beep
No Beep
Beep
No Beep
Beep
No Beep
Beep
No Beep
Beep
```

When the webcam detects a human, the buzzer is turned on, which alerts the police and firemen nearby

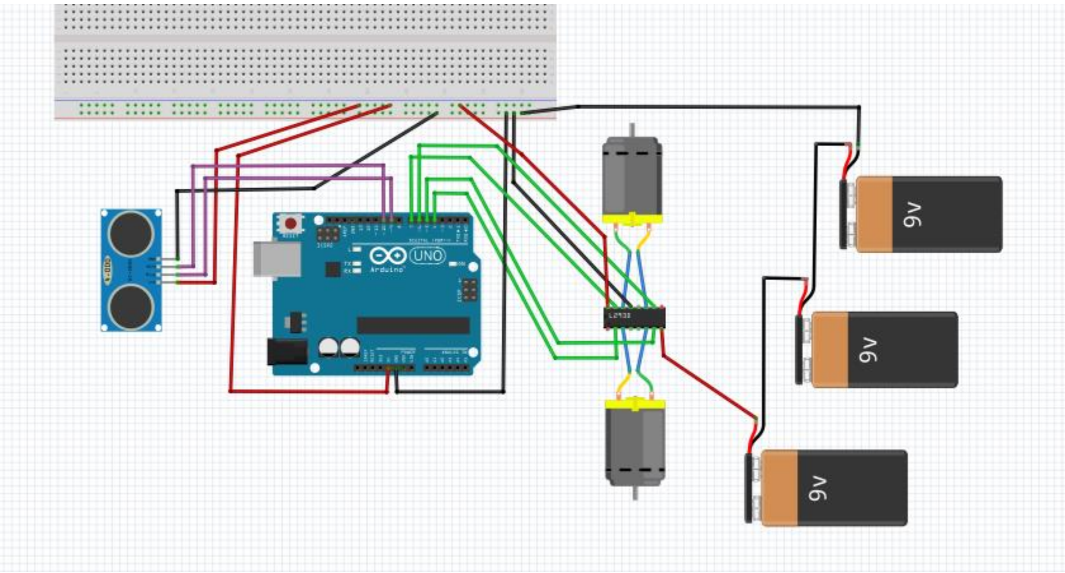


```
import RPi.GPIO as GPIO
from time import sleep
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
buzzer=23# gpio 16
GPIO.setup(buzzer,GPIO.OUT)
import numpy as np
import cv2

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

cap = cv2.VideoCapture(0) #Get vidoe feed from the Camera

while(True):
```

```
Beep
No Beep
Beep
No Beep
Beep
No Beep
Beep
No Beep
Beep
No Beep
Beep
No Beep
```

Buzzer does not beep, when there is no human in the vicinity

Subsystem 2: navigation



Code:

```
int trigPin = 9;      // trig pin of HC-SR04
int echoPin = 10;
int led=13;// Echo pin of HC-SR04


int revleft4 = 4;       //REVerse motion of Left motor
int fwdleft5 = 5;       //ForWarD motion of Left motor
int revright6 = 6;      //REVerse motion of Right motor
int fwdright7 = 7;      //ForWarD motion of Right motor


long duration, distance;


void setup() {
  delay(random(500,2000));   // delay for random time
  Serial.begin(9600);
  pinMode(revleft4, OUTPUT);      // set Motor pins as output
  pinMode(fwdleft5, OUTPUT);
  pinMode(revright6, OUTPUT);
  pinMode(fwdright7, OUTPUT);

  pinMode(trigPin, OUTPUT);         // set trig pin as output
  pinMode(echoPin, INPUT);  // set echo as input
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);     // send waves for 10 us
  delayMicroseconds(10);
  duration = pulseIn(echoPin, HIGH); // receive reflected waves
  distance = duration*0.017; // convert to distance
  Serial.println(distance);
  delay(10);
  if (distance >=15)
  {
forward();
  }
  if (distance < 15)
  {
    stopped();
    delay(500);
    back();
    delay(500);
    stopped();
    delay(100);
    turn();
    delay(500);
  }
}
void forward()
{
  Serial.println("front");
    digitalWrite(fwdright7, HIGH);                    // move forward
    digitalWrite(revright6, LOW);
    digitalWrite(fwdleft5, HIGH);
    digitalWrite(revleft4, LOW);
}
void stopped()
{
  Serial.println("stop");
    digitalWrite(fwdright7, LOW);  //Stop
    digitalWrite(revright6, LOW);
    digitalWrite(fwdleft5, LOW);
    digitalWrite(revleft4, LOW);
}
```

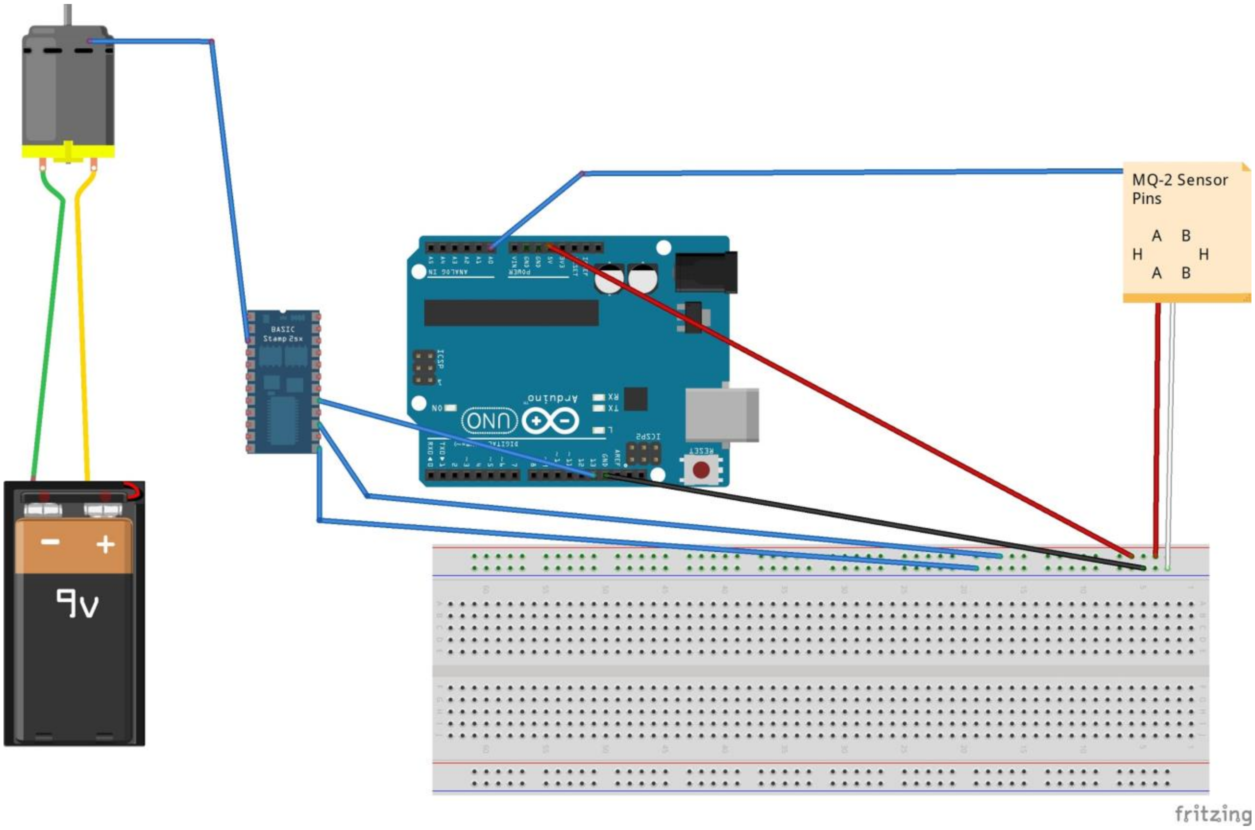
```
void turn()
{
  Serial.println("turn");
    digitalWrite(fwdright7, HIGH);
    digitalWrite(revright6, LOW);
    digitalWrite(revleft4, LOW);
    digitalWrite(fwdleft5, LOW);
}
void back()
{
  Serial.println("back");
    digitalWrite(fwdright7, LOW);     //movebackword
    digitalWrite(revright6, HIGH);
    digitalWrite(fwdleft5, LOW);
    digitalWrite(revleft4, HIGH);
}
```

Source: https://circuitdigest.com/microcontroller-projects/arduino-obstacle-avoding-robot

Subsystem 3: fire extinguishing

Flame sensor and water spraying system.

This was implemented using an Arduino successfully.



Algorithm:

If the input from the flame sensor is high, switch on the pump for 5 seconds.

Code:

```
#include <MQ2.h>

#include <Wire.h>

int Analog_Input = A0;

int lpg, co, smoke;
```

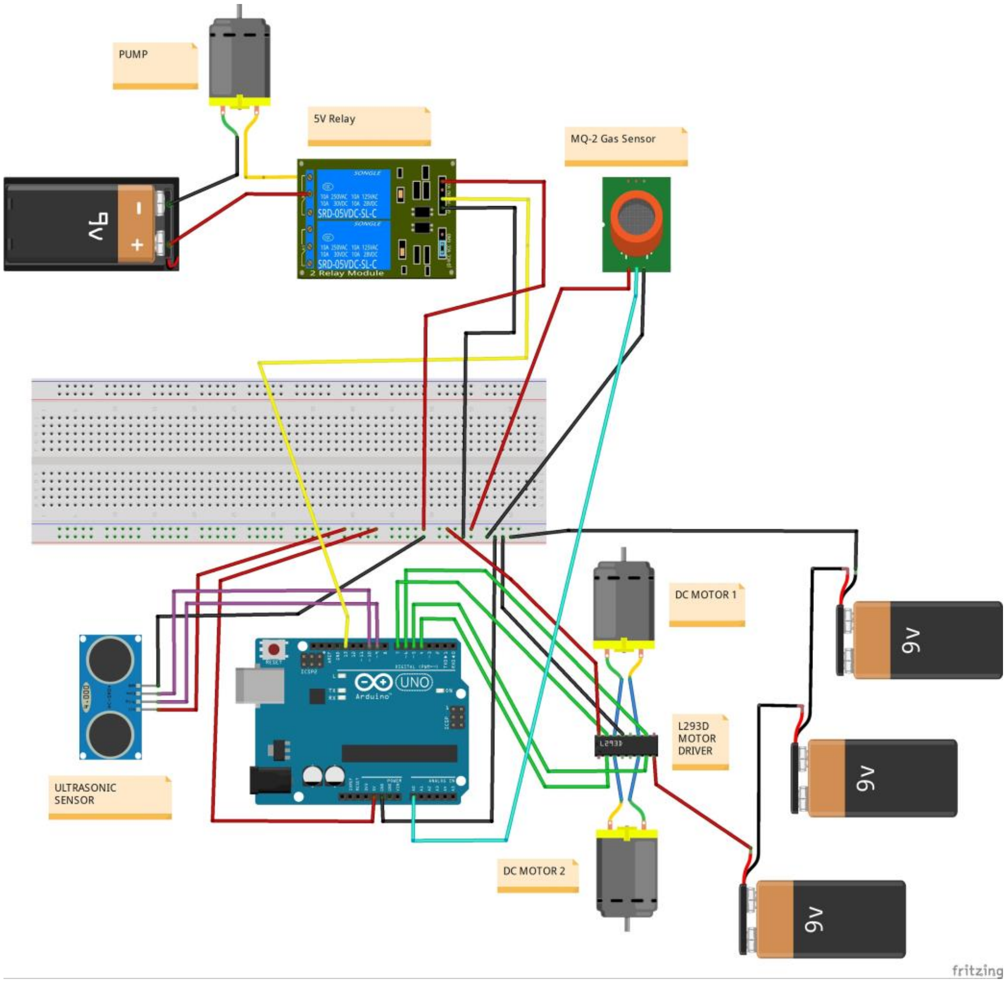
```
MQ2 mq2(Analog_Input);

int pumpPin = 13;

void setup() {

   delay(random(500,2000));   // delay for random time

  Serial.begin(9600);

   pinMode(pumpPin, OUTPUT);

  mq2.begin();

}

void loop() {

  digitalWrite(pumpPin,LOW);

  float* values= mq2.read(false); //set it false if you don't want to print the values in the Serial

  smoke = mq2.readSmoke();

  Serial.println("smoke");

 Serial.println(smoke);

 if(smoke>3000)

 {

  digitalWrite(pumpPin,HIGH);

  delay(5000);

 }

 digitalWrite(pumpPin,LOW);

  delay(1000);

}
```

Final Connections

Arduino pins 4,5,6,7	l293D pins in1,in2,in3,in4
Arduino ground	9V battery ground
+ve of 9v battery	l293D +ve
Arduino ground	l293D ground
Motors +ve and -ve	Motor pins of l293D
Arduino digital pin 10	Ultrasonic echo
Arduino digital pin 9	Ultrasonic trig
Arduino +5v	Ultrasonic vcc
Arduino ground	Ultrasonic ground
Arduino analog A0	Mq A0
Arduino ground	Mq ground
Arduino +5v	Mq vcc
Arduino digital pin 12	In of relay

Arduino ground	Ground of relay
Arduino +5v	Power of relay
NO pin of relay	Pump +ve
-ve of battery	Pump -ve
+ve of battery	Common of relay

Final Circuit Diagram



Final Code

```
#include <MQ2.h>
#include <Wire.h>
int Analog_Input = A0;
int lpg, co, smoke;
MQ2 mq2(Analog_Input);
int pumpPin = 12;
int trigPin = 9;     // trig pin of HC-SR04
int echoPin = 10; //echo pin of HC-SR04
int revleft4 = 4;       //REVerse motion of Left motor
int fwdleft5 = 5;       //ForWarD motion of Left motor
int revright6 = 6;      //REVerse motion of Right motor
int fwdright7 = 7;      //ForWarD motion of Right motor
long duration, distance;

void setup() {

  delay(random(500,2000));   // delay for random time
  Serial.begin(9600);
  pinMode(revleft4, OUTPUT);      // set Motor pins as output
  pinMode(fwdleft5, OUTPUT);
  pinMode(revright6, OUTPUT);
  pinMode(fwdright7, OUTPUT);

  pinMode(trigPin, OUTPUT);         // set trig pin as output
  pinMode(echoPin, INPUT);

  //set echo pin as input to capture reflected waves

  pinMode(pumpPin, OUTPUT);

  mq2.begin();

}


void loop() {

digitalWrite(pumpPin,LOW);

forward();

  float* values= mq2.read(false);

  smoke = mq2.readSmoke();

  Serial.println("smoke");

 Serial.println(smoke);

 if(smoke>3500)

 {

  stopped();

  Serial.println("pump on");

  digitalWrite(pumpPin,HIGH);

  delay(5000);

 Serial.println("pump off");

 digitalWrite(pumpPin,LOW);

  delay(1000);

}

  digitalWrite(trigPin, LOW);

  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);     // send waves for 10 us
  delayMicroseconds(10);
  duration = pulseIn(echoPin, HIGH); // receive reflected waves
  distance = duration*0.017;
  // convert to distance
  Serial.println(distance);
  delay(10);
  if (distance >=15)
  {
forward();
  }
  if (distance < 15)
  {
    stopped();
    delay(500);
    back();
    delay(500);
    stopped();
    delay(100);
    turn();
    delay(500);
  }

}
void forward()
{
  Serial.println("front");
```

```
    digitalWrite(fwdright7, HIGH);                  // move forward
    digitalWrite(revright6, LOW);
    digitalWrite(fwdleft5, HIGH);
    digitalWrite(revleft4, LOW);
}
void stopped()
{
  Serial.println("stop");
    digitalWrite(fwdright7, LOW);  //Stop
    digitalWrite(revright6, LOW);
    digitalWrite(fwdleft5, LOW);
    digitalWrite(revleft4, LOW);
}
void turn()
{
  Serial.println("turn");
    digitalWrite(fwdright7, HIGH);
    digitalWrite(revright6, LOW);
    digitalWrite(revleft4, LOW);
    digitalWrite(fwdleft5, LOW);
}
void back()
{
  Serial.println("back");
    digitalWrite(fwdright7, LOW);     //movebackword
    digitalWrite(revright6, HIGH);
    digitalWrite(fwdleft5, LOW);

    digitalWrite(revleft4, HIGH);

  }
```