Name : Vishal Kumar Mahatha
Reg. No. : 20BRS1168

LAB 3

Smart Parking System

Aim – To create a smart parking system using Arduino, sensors, and Node-RED for monitoring and notification.

Components -

IR sensor
LED
HC-SR04
Arduino UNO
Male-Female Jumper cables
Male – Male USB A to B cable for data transfer and power

Procedure –

- Take the LED, IR sensor, HC-SR04 and Arduino UNO (ensure there is no power supply to Arduino)
- Connect VCC and GND pin of both IR and ultrasonic sensor to the 5V and ground pin of Arduino respectively using jumper cables.
- Connect Trig and Echo pin of HC-SR04 and out pin of the IR sensor to any digital pin on Arduino and note it for coding.
- Connect LED's longer pin to any digital pin of Arduino and the other one to gnd.
- Connect USB A to B cable and code the Arduino to show an output whenever a signal is received from the sensor.
- Stream the message on serial port on a baud rate of 9600.
- Upload the code and observe changes in on serial monitor and the LED.
- Get the values on serial monitor on Node-Red through COM7
- Check for the values and output desirable notification using notification node.

Sample code –

- Arduino

```
//UR sensor

const int trigPin = 9;

const int echoPin = 10;

long duration;
```

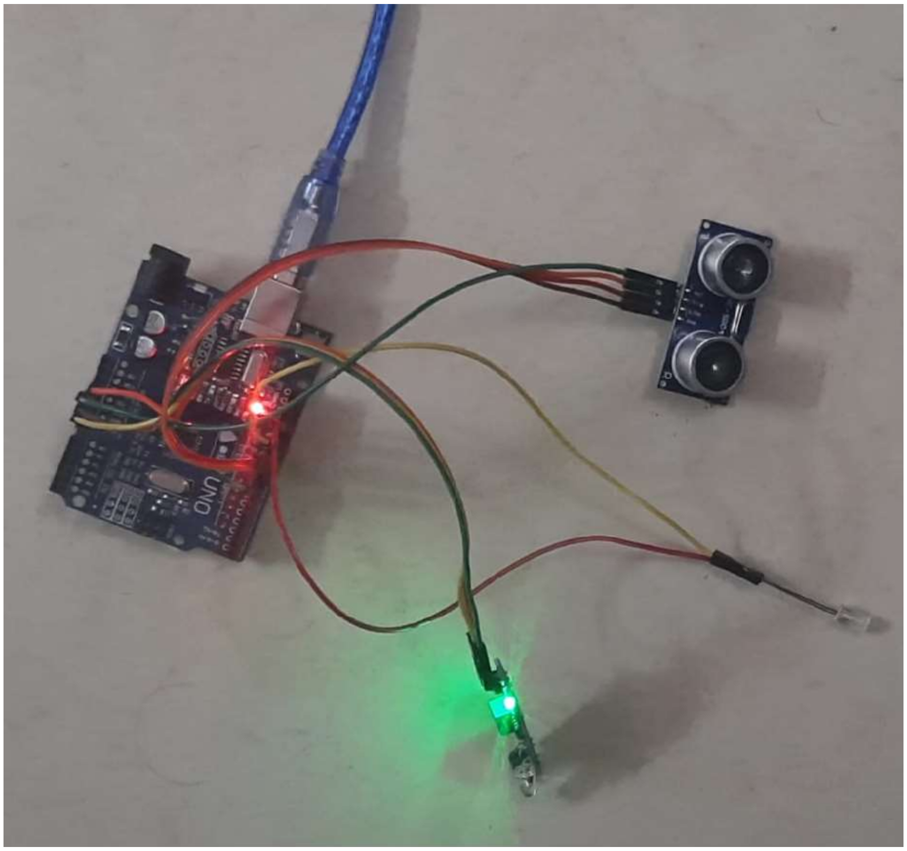
```
int distance;



//LED and IR sensor
int LEDpin = 12;
int obstaclePin = 7;
int hasObstacle = HIGH;

void setup() {
        pinMode(trigPin,OUTPUT);
         pinMode(echoPin,INPUT);
        pinMode(LEDpin, OUTPUT);
        pinMode(obstaclePin, INPUT);
        Serial.begin(9600);
}

void loop() {
        digitalWrite(trigPin,LOW);
        delayMicroseconds(2);
        digitalWrite(trigPin,HIGH);
        delayMicroseconds(10);
        digitalWrite(trigPin,LOW);
        duration = pulseIn(echoPin,HIGH);
        distance = duration * 0.034/2;
        Serial.println(String(distance) + "," + String(hasObstacle));
        if(distance <= 200){
                digitalWrite(LEDpin, HIGH);
                delay(4000);
                digitalWrite(LEDpin, LOW);
        }

        hasObstacle = digitalRead(obstaclePin);

        if (hasObstacle == LOW){

                digitalWrite(LEDpin, HIGH);

                delay(4000);

                digitalWrite(LEDpin, LOW);

        }

        delay(2000);

}
```

- Function node

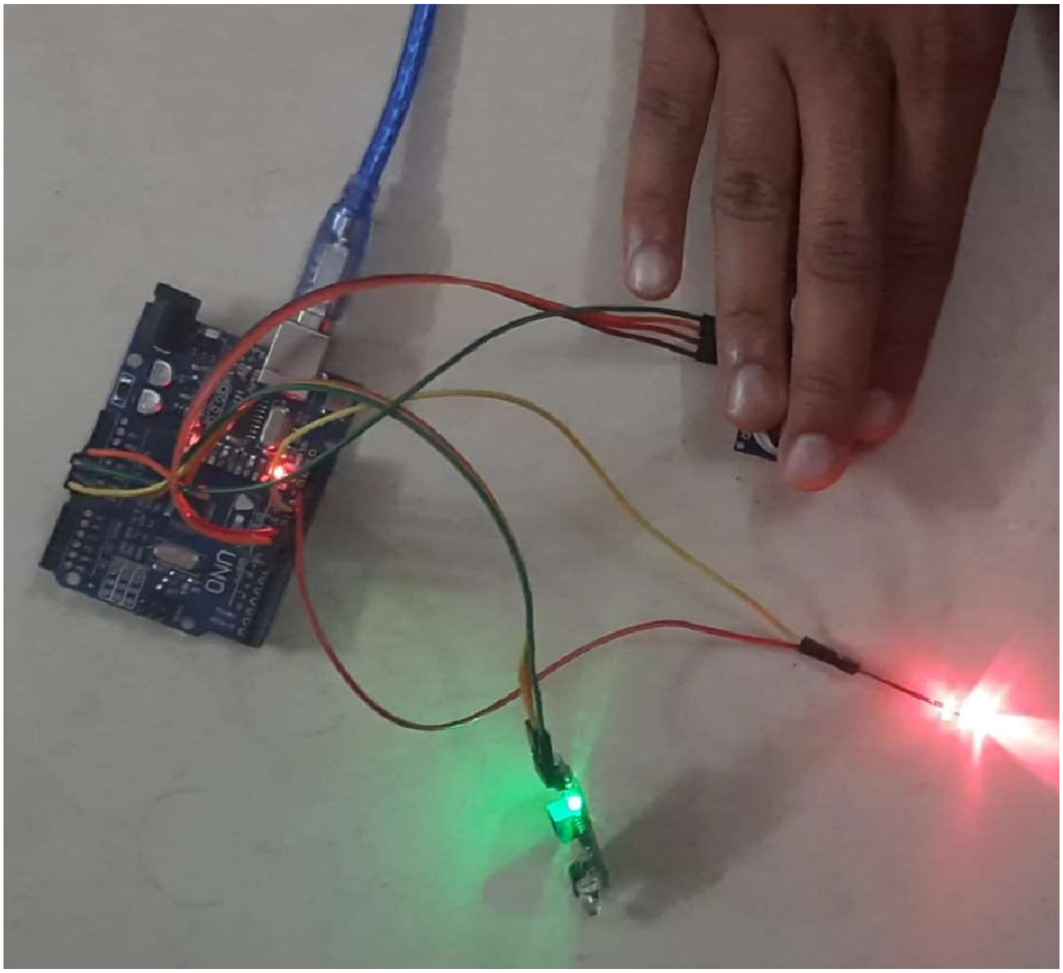
```
var m = msg.payload.split(',');
var Ultrasonic = { payload: parseFloat(m[0]) };
var IR = { payload: parseFloat(m[1]) };
return [Ultrasonic, IR];
```

Output –

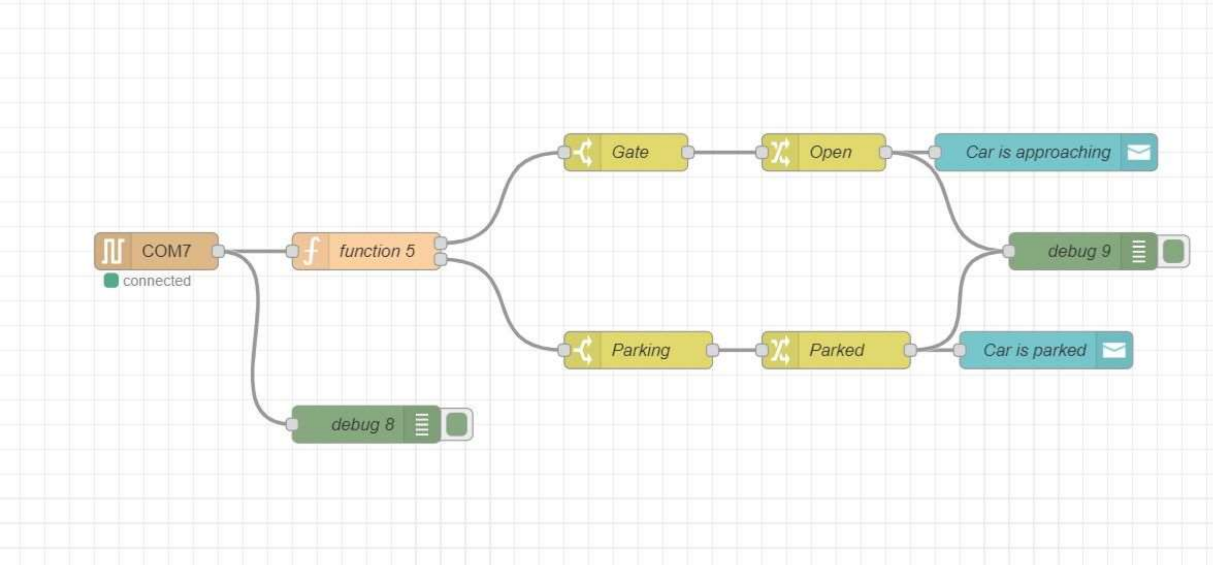
Circuit



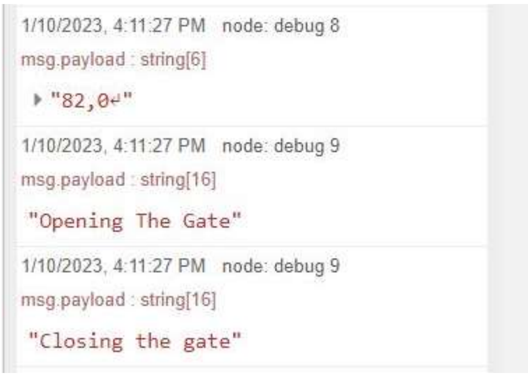
Circuit (Gate is opening/closing)



Node-red (Flow)

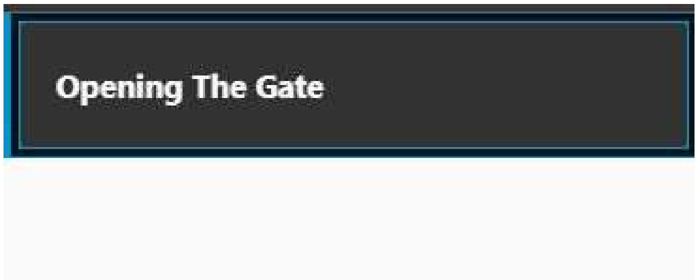


Node-red (Debug)

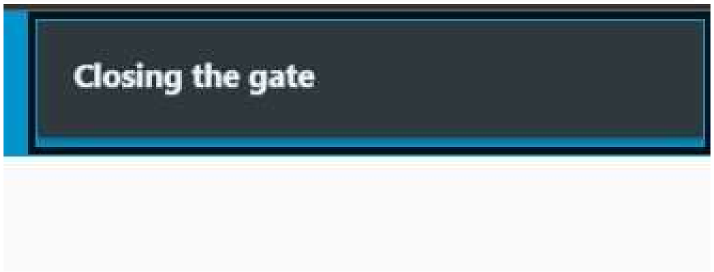


```
1/10/2023, 4:11:27 PM   node: debug 8
msg.payload : string[6]
 ▸ "82,0↵"
1/10/2023, 4:11:27 PM   node: debug 9
msg.payload : string[16]
 "Opening The Gate"
1/10/2023, 4:11:27 PM   node: debug 9
msg.payload : string[16]
 "Closing the gate"
```

Notifications



Opening The Gate



Closing the gate