ESCI

Engineering Staff College of India

Autonomous Organ of The Institution of Engineers (India)

(An ISO 9001:2008 Certified, AICTE & CEA Recognized Institution)

Industry Partner

SMARTBRIDGE

Let's Bridge the Gap

Internship Project Report

Smart Car Accident Prevention (SCAP)

Done by

Raghav Khera, Shubham Sardana, Ankush Kamboj, Deepak Dhanda, Gaurav Raheja

Smart Car Accident Prevention

(S.C.A.P)

Table of Contents

Abstract

The project is about making cars more intelligent and interactive which will notify or resist user under unacceptable conditions, device made by our team will be fitted to car which will provide critical information of real time situations to rescue owner himself. In this documentation, we describe a real-time online safety prototype that prevents the starting of the engine in the case of brake failure and warn the user through SMS and buzzer or LED. The purpose of such a model is to advance a system to detect brake failure cause of accident and control the speed of vehicle to avoid accidents. The main components of the system consist of number of real time sensors like Ultra-Sonic Sensors, to increase more chances of avoiding accidents two IR senssor can be embeded with our circuits at sideways of the car.

Introduction

The **Internet of Things (IoT) is the interconnection of uniquely identifiable embedded computing devices within the existing Internet infrastructure**. Typically, IoT offers advanced connectivity of devices, systems, and services that goes beyond machine-to-machine communications (M2M) and covers a variety of protocols, domains, and applications.The interconnection of these embedded devices (including smart objects), is implemented in nearly all fields of automation enabling advanced applications like a SmartGrid. The term things in the IoT refers to a wide variety of devices such as heart monitoring implants, biochip transponders on farm animals, electric clams in coastal waters, automobiles with built-in sensors, or field operation devices that assist fire-fighters in search and rescue. Current market examples include thermostat systems and washer/dryers that utilize Wi-Fi for remote monitoring.

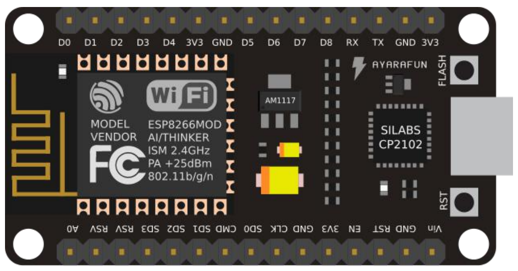
We are using Node MCU as the main control unit in this project project. When the system is switched on, LED will be ON indicating that power is supplied to the circuit. In case of the brake failure the car isn't allowed to move on further and this information will be sent to a mobile number through a text message and in case of network problem the user will be warned by LED or Buzzer. This message will be received against the unique ID embeded in the program that is dumped into Node MCU unit to the cloud, and the cloud will process the data against that registered ID and send SMS to the Number through any messaging server or any API. The message will give the information of brake failure.

Implementation

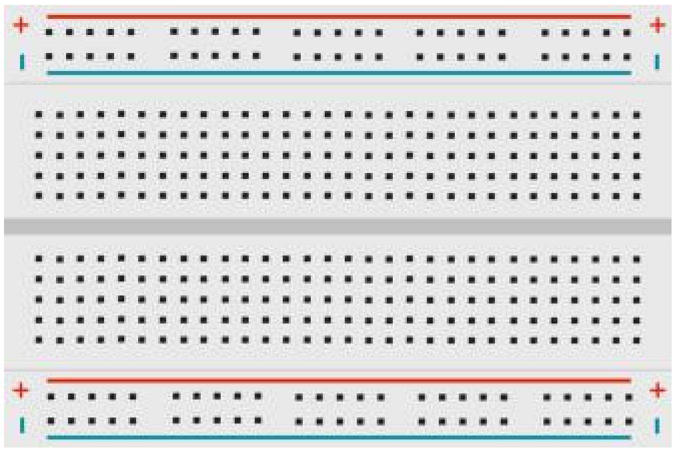
Various Major Hardware Components that are used in the circuits are:

- **ULTRASONIC SENSOR:** The HC-SR04 ultrasonic sensor uses sonar principle to determine the distance to an object. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package from 2cm to 400 cm or 1inch to 13 feet.

- **NODE-MCU:** Node MCU development board is an OPEN- SOURCE IOT development kit. It is the low cost hardware module which is used to connect IOT Application ex - Ultrasonic Sensor.


D0 D1 D2 D3 D4 3V3 GND D5 D6 D7 D8 RX TX GND 3V3
AYARAFUN
FLASH
MODEL ESP8266MOD
VENDOR AI/THINKER
ISM 2.4GHz
PA +25dBm
802.11b/g/n
AM1117
SILABS CP2102
RST
A0 RSV RSV SD3 SD2 SD1 CMD SD0 CLK GND 3V3 EN RST GND Vin

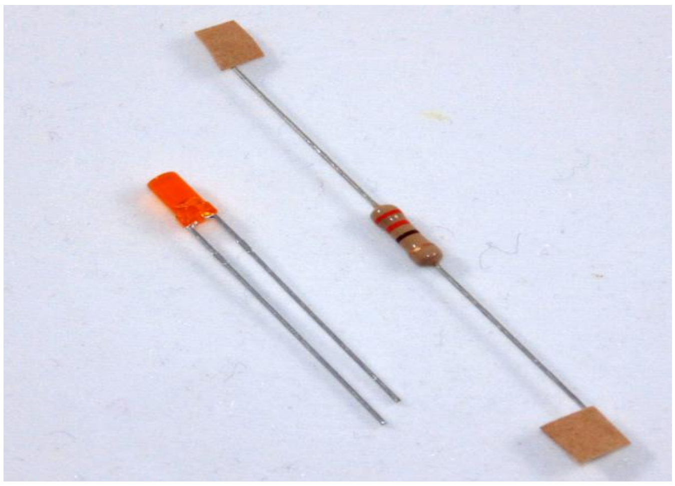
- **BREADBOARD:** A breadboard is a solder less device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate.



LED with a Resistor: You may take any LED for demonstrating about the brake failure, but you need to take a suitable resistor that is to be connected to the ground pin in the series with LED. Each type of LED have a fairly constant voltage across them.If you supply 3V to this LED without series resistor the LED will try to set for a voltage/current combination for this 3V. There's no current that goes with this kind of voltage, theoretically it would be 10s, maybe 100s of amperes, which would destroy the LED. And that's exactly what happens if your power supply can supply enough current.
So the solution is a series resistor. If your LED needs 20mA you can calculate for the red LED in the example

$$R=\Delta V/I=3V-2.2V/20mA=40\Omega$$

You may think that supplying 2.2V directly will also work, but that's not true. The slightest difference in LED or supply voltage may cause the LED to light very dim, very bright, or even destroy. A series resistor will ensure that slight differences in voltage have only a minor effect on the LED's current, provided that the voltage drop across the resistor is large enough.

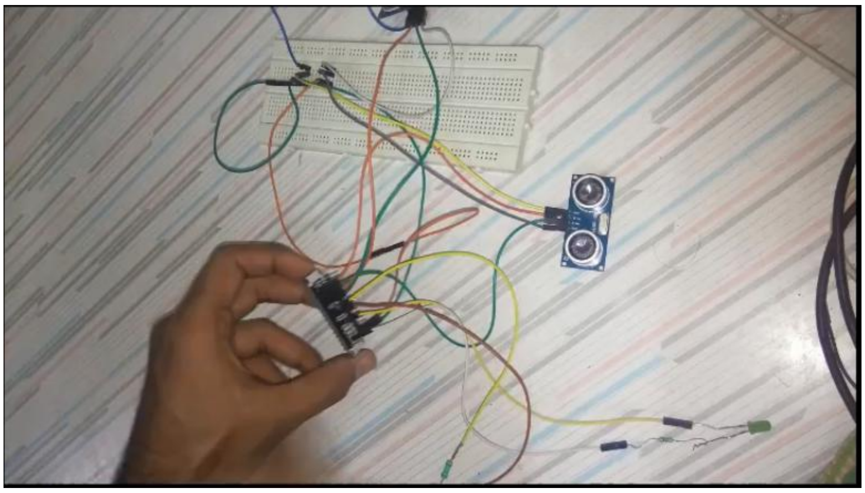


Connect the node MCU with ultrasonic sensor pins as Mentioned Below

Node MCU	**Ultrasonic Sensor Front**
D1	Trigger
D2	Echo
Vin	Vcc
GND	GND

Node MCU	**Ultrasonic Sensor Back**
D3	Trigger
D4	Echo
Vin	Vcc
GND	GND

Since there is only one Vin Pin on the node MCU but we need to conect two HC-SR04 sensor to the Node MCU. So we need to connect the two sensor through breadboard to Node MCU. The trigger Pin and Echo Pin of each of the sensor can be connected to any of the digital I/O pin of the Node MCU. But we are using pin mentioned in the above table. Connect these pin with each other through jumper wires. Then connect LED to any Digital Input Output Pin and our circuit is ready and will look as shown below:



For the processing of data and sending sms, and demonstrating the change in speed as the object comes near or away from the object, we use **IBM bluemix**

Then we need to dump the program that is mentioned in the appendix, but before that we need to change some credentials that are mentioned in the program

Look at the following segment in the code, there replace orgid, DeviceType, DeviceID, Authentication Token with your device created in IBM Blemix.

```
#define ORG orgid
#define DEVICE_TYPE DeviceType
#define DEVICE_ID DeviceID
#define TOKEN AuthenticationToken
```

Then You need to setup the SSID and Password of wifi and the Hostname You of the server you are using. Since we are using IBM Cloud our HostName Main URL will look like "device-name.eu-gb.mybluemix.net" There might be any other if you are using some other cloud:

```
      const char* gssid     = YourSSID;
      const char* password = Password;
      const   char*   host   =   YourHostNameMainURL;   //"device-name.eu-
gb.mybluemix.net"
```

Note: User should be provided an automatic wifi connectivity client since user can change the password or device most oftenly

If you are using other pin then the pin mentioned in the above table change the pin D1, D2, D3, D4, D6 , D7 accordingly in the following Snippets

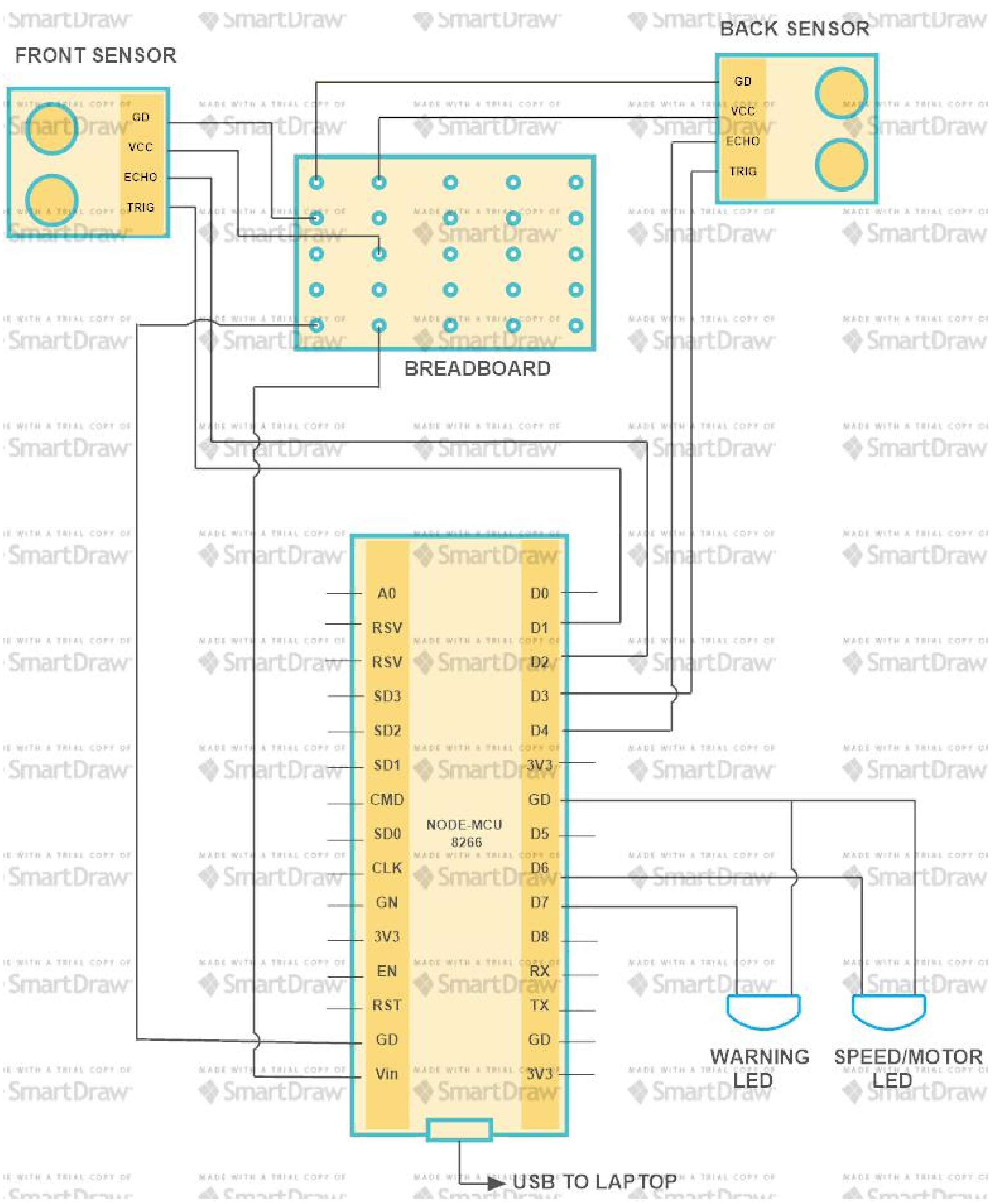
```
            //Sensor Pin Number and Node MCU conncetion
            const int trigPinFront = D1;
            const int echoPinFront = D2;
            const int trigPinBack= D3;
            const int echoPinBack = D4;
            int motor = D6;
            int alert_led = D7;
```

After that you need to save the registered _id against that client to the program so that the cloud can process the data stored against that id and can extract the phone number and other required credentials. To warn the user against some inappropriate conditions:

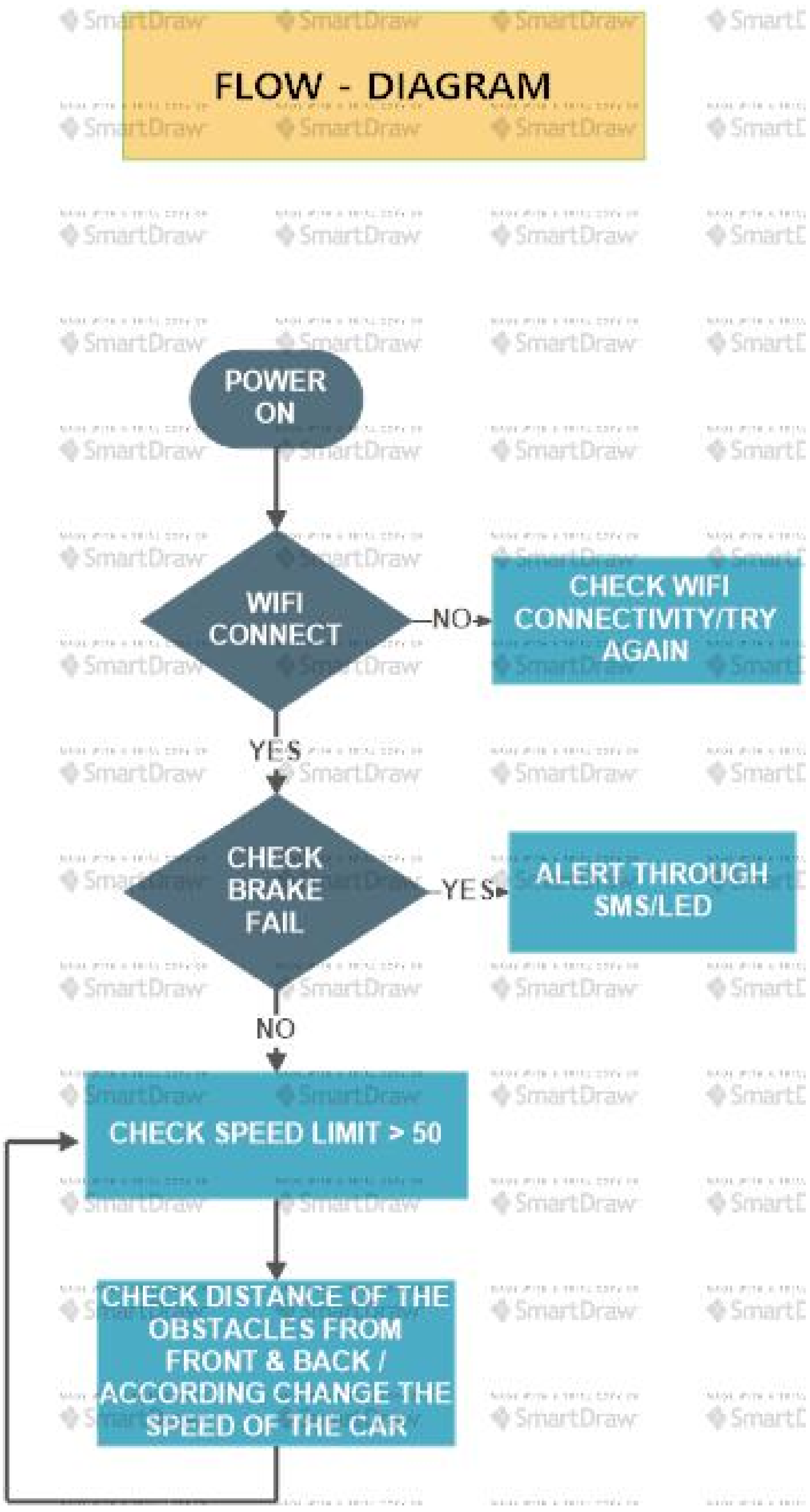
```
      char* _id = OwnerUniqueID;
```

Now change the **brakeFailed** variable to True or False accordingly if you want to see the output in the case of brake failed or working properly, and dump the program to the device using arduino IDE and an USB Cable.

BLOCK Diagram



FLOW Diagram

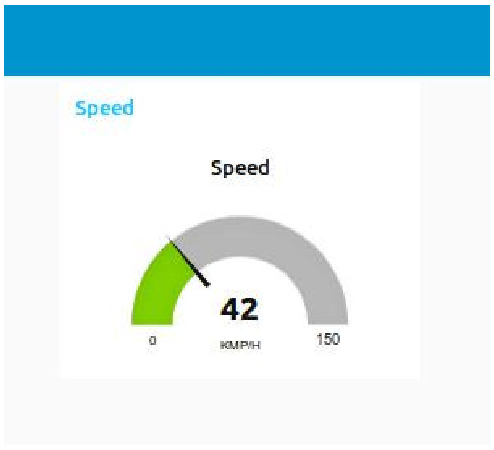

FLOW - DIAGRAM
POWER ON
WIFI CONNECT
NO
CHECK WIFI CONNECTIVITY/TRY AGAIN
YES
CHECK BRAKE FAIL
YES
ALERT THROUGH SMS/LED
NO
CHECK SPEED LIMIT > 50
CHECK DISTANCE OF THE OBSTACLES FROM FRONT & BACK / ACCORDING CHANGE THE SPEED OF THE CAR

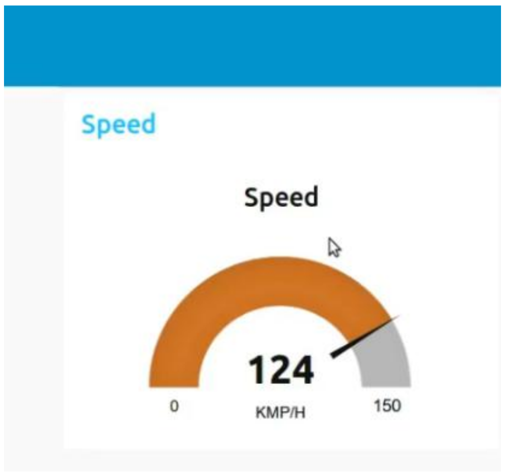
Result & Discussion

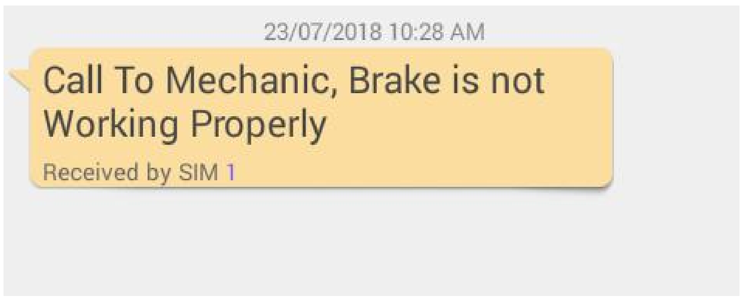
Our Sensor starts to operate at speed greater than or equal to speed limit fixed in the program. The two picture shown below, reveal the two cases

1. Decrease in speed when the vehicle comes too closer from the front.

2. Increase in the speed as the vehicles approaches too near from the rear.



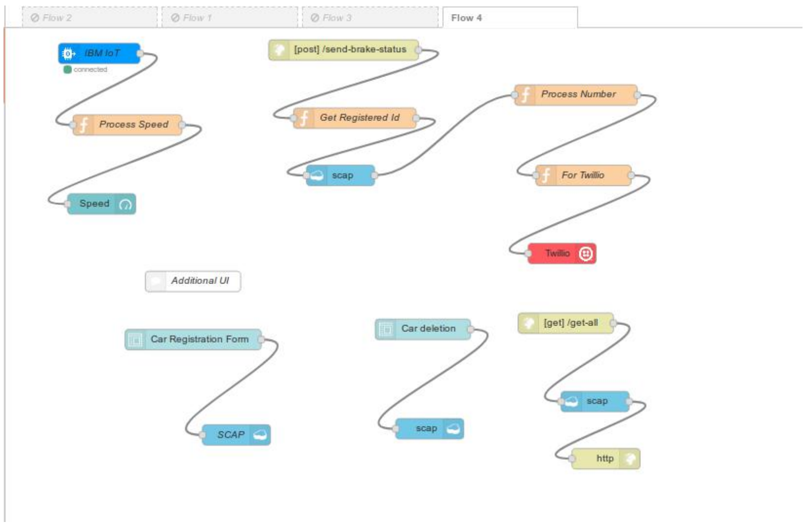



23/07/2018 10:28 AM

Call To Mechanic, Brake is not Working Properly

Received by SIM 1

The picture shown on the left side represent the message sent to the owner of the car in case of brake failure



The picture shown on the right hand side represent the node red flow that operate all of the above functioning like change in speed or sending of warning sms.

Advantages

- Our main motive to make this project is to reduce road accidents and save life not only the mankind but also the stray animals which accidentally come on the roads.
- This will help us in increasing road safety.
- This will also reduce the accidents in bad environmental conditions.
- Alerts will also helping gaining the drivers attention back on the road.
- Medical and other helps can reach the spot better.
- In case of brake failure there are to alerts being provided in the system one being the sms alert other being the LED or buzzer alert which decrease the case of any mishappening.

Disadvantages

- Drivers may show negligence while driving by over reeling on the sensors.
- Sudden Brakes may lead to internal injuries.
- Failure of sensors may lead to accidents .

Future Scope

1. **Vehicle-to-Vehicle Communication:** In future, the self-driving cars must communicate with each other for a smooth operation. Understanding the GPS coordinates and the speed to play a major role to facilitate this and will depend a lot on the new age sensor technology which will focus on the cars to communicate. The sensors also detect pedestrians, bicycles within its proximity and adjusts the car's speed accordingly.
2. **Traction control system (TCS):** The traction control system prevents wheel spin when starting off or accelerating, particularly on a slippery or wet road surface. While the antilock braking system (ABS) prevents the wheels from locking during braking by reducing the braking pressures, TCS ensures that the wheels do not spin when driving off or accelerating. To do this, the drive torque at each driven wheel is reduced correspondingly. TCS improves the traction of the vehicle and increases vehicle safety by avoiding unstable driving situations within the limits of physics.

Appendix

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266HTTPClient.h>
#include <PubSubClient.h>


//For PUBSUB - just used for demonstration of device
#define ORG orgid
#define DEVICE_TYPE DeviceType
#define DEVICE_ID DeviceID
#define TOKEN AuthenticationToken


char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char topic[] = "iot-2/evt/send-speed/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

const char* gssid = YourSSID;
const char* password = YourPassword;
const char* host = YourHostNameMainURL;
//"device-name.eu-gb.mybluemix.net"

//Sensor Pin Number and Node MCU conncetion
const int trigPinFront = D1;
const int echoPinFront = D2;
const int trigPinBack= D3;
const int echoPinBack = D4;
int motor = D6;
int alert_led = D7;
char* _id = OwnerUniqueID;


//other variables defining states of car
boolean brake_failed=true, brake_failed_status;
int distFromFront,distFromBack;
int very_near = 5, near = 10, mid = 50, far =100;
int spd,speed_after_which_sensor_operates, max_range;


//Method Declaration
int get_distance(int, int);
boolean check_status_of_brake(boolean);
void post_brake_status();
void wifiConnect(const char *,const char *);
WiFiClient wifiClient;
PubSubClient client(server, 1883, wifiClient);

void setup() {
    Serial.begin(115200);
    delay(1000);
    pinMode(trigPinFront, OUTPUT);
    pinMode(echoPinFront, INPUT);
    pinMode(trigPinBack, OUTPUT);
    pinMode(echoPinBack, INPUT);
    pinMode(motor, OUTPUT);
    pinMode(alert_led, OUTPUT);

    brake_failed_status= check_status_of_brake(brake_failed);
    wifiConnect(gssid,password);
    mqttConnect();
    delay(10);
    speed_after_which_sensor_operates = 50;
    max_range = 150;
    spd=0;
    if (!brake_failed_status){
      for(int i =0; i<100; i++)
      {
          spd +=1;
          analogWrite(motor,spd*2);
          delay(100);
       }
    }
    else{
        post_brake_status();
    }
    Serial.println(spd);

}

void loop() {
    while (!brake_failed_status){

        delay(100);
        while(spd > speed_after_which_sensor_operates){
            distFromFront = get_distance(trigPinFront, echoPinFront);
            delay(100);
            distFromBack = get_distance(trigPinBack, echoPinBack);
            delay(100);
            while (isnan(distFromFront) || isnan(distFromBack))
            {
              Serial.println("Your Sensor aren't working Properly, Connect
Your Sensor Properly");
              delay(1000);
            }
            if(distFromFront < very_near){
                  spd -= 12;
                  Serial.println("The Vehicle Ahead You is too closer,
Stopping your vehicle and dropping Speed down to " + String(spd) );
            }
            else if (distFromFront<= near){
                if (distFromBack <= near)
                {
                    spd -= 10;
                    Serial.println("The Vehicle ahead You and behind You
are approaching near, So decrasing speed down to " +String(spd));
                }
                else
                {
                    spd -= 8;
                    Serial.println("The  Vehicle  ahaed  You  and  behind  You
are approaching near, So decrasing speed down to "+ String(spd));
                }
            }
            else if(distFromFront<=mid && distFromFront>near){

               if (distFromBack<near){
                  spd += 10;
                  Serial.println("The  Vehicle  behind  you  is  approaching
towards You so increasing speed by "+ String(spd));
               }
               else{

                  spd+=7;
               }

            }
            else if (distFromFront<=far && distFromFront>mid) {
               if (distFromBack<near){
                  spd += 12;
                  Serial.println("The  Vehicle  behind  you  is  approaching
towards You so increasing speed by " + String(spd));
               }
               else{
                  spd+=10;
               }
            }
            else{
              //spd=60;

            }
            if (spd<0)
            {
                spd =0;
            }
            if (spd > max_range){

              spd = max_range;
            }
            PublishData(spd,distFromFront,distFromBack, _id);
            if (!client.loop()) {
                mqttConnect();
                delay(100);
            }
            analogWrite(motor,spd*4);
        }
        delay(100);
        brake_failed_status = check_status_of_brake(brake_failed);
    }
    Serial.println("Brake Failed Call Mechanic");
    delay(100);
    while(brake_failed_status){
        digitalWrite(alert_led,HIGH);
        delay(100);
        digitalWrite(alert_led,LOW);
        delay(100);
    }



}


int get_distance(int t_pin, int e_pin){
    int distance;
    int duration;
    digitalWrite(t_pin, LOW);
    delay(2);
    digitalWrite(t_pin, HIGH);
    delay(10);
    digitalWrite(t_pin, LOW);
    duration = pulseIn(e_pin, HIGH);
    distance= duration*0.034/2;
    return distance;

}

boolean check_status_of_brake(boolean brake){

    if(brake_failed){
        spd=0;
        return true;
    }
    else{
        return false;
    }
}

void wifiConnect(const char *ssid,const char *passw) {

    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, passw);

    while (WiFi.status() != WL_CONNECTED) {
        delay(100);
        Serial.print(".");
        if (brake_failed_status){
                  digitalWrite(alert_led,HIGH);
        delay(100);
        digitalWrite(alert_led,LOW);
        delay(100);
          }
    }

    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}
```

```
void post_brake_status()
{
    Serial.print("Connecting to ");
    Serial.println(host);

    String postData = "_id="+String(_id);
    HTTPClient http;
    http.begin("http://beast-app.eu-gb.mybluemix.net/send-brake-status");
    http.addHeader("Content-Type",    "application/x-www-form-urlencoded");
//Specify content-type header
    int httpCode = http.POST(postData);   //Send the request
    String payload = http.getString();    //Get the response payload

    Serial.println(httpCode);   //Print HTTP return code
    Serial.println(payload);    //Print request response payload

    http.end();

}
void mqttConnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting MQTT client to ");
    Serial.println(server);
    while (!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }
    initManagedDevice();
    Serial.println();
  }
}
void initManagedDevice() {
    if (client.subscribe(topic)) {
        Serial.println("subscribe to cmd OK");
    } else {
        Serial.println("subscribe to cmd FAILED");
    }
}
void PublishData(int spd,int frontDist,int backDist,char *id){
   if (!!!client.connected()) {
      Serial.print("Reconnecting client to ");
      Serial.println(server);
      while (!!!client.connect(clientId, authMethod, token)) {
          Serial.print(".");
          delay(500);
      }
      Serial.println();
   }
   String payload = "{\"d\":{\"speed\":";
   payload += spd;
   payload += ",\"frontDist\":";
   payload += frontDist;
   payload += ",\"backDist\":";
   payload += backDist;
   payload += ",\"_id\":\"";
   payload += id;

   payload += "\"}}";
   Serial.print("Sending payload: ");
   Serial.println(payload);
   if (client.publish(topic, (char*) payload.c_str())) {
      Serial.println("Speed :"+ String(spd));
   }
   else {
      Serial.println("Publish failed");
   }
}
```