Hardware Simulation

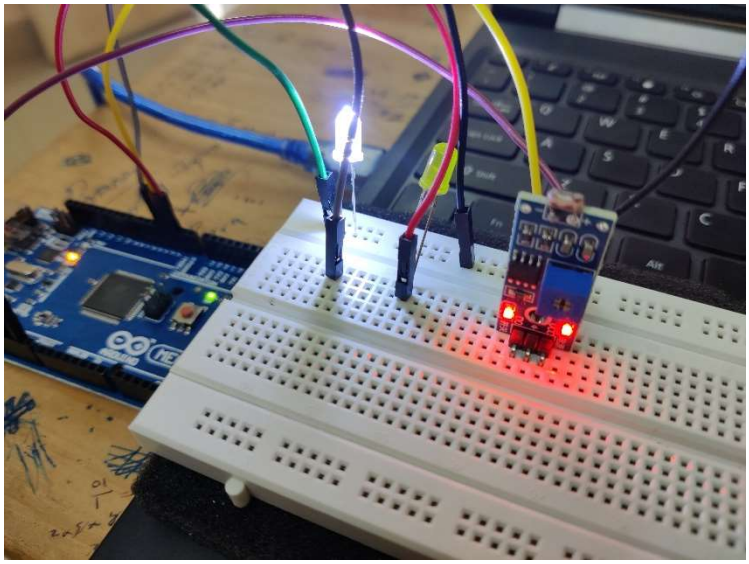
Group 31 – Advanced Auto

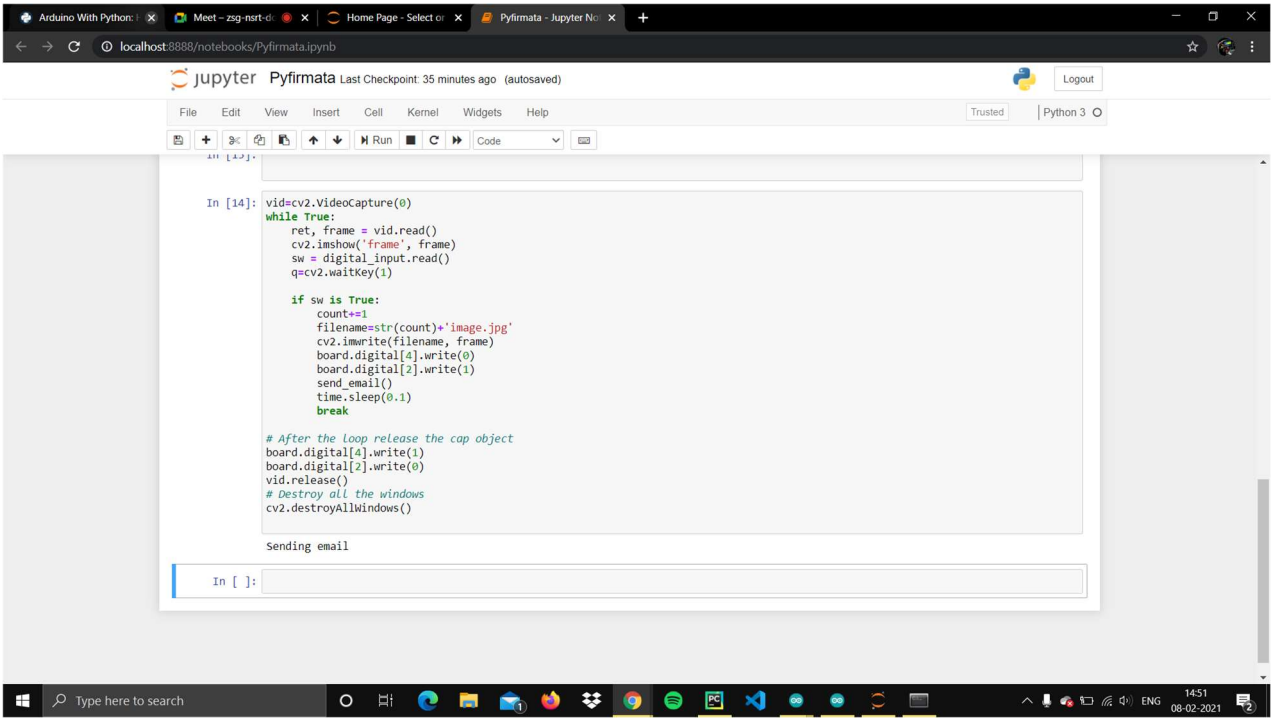
DJ Strike 2020-21

1. Garage Gate (Frame Capture in Python with IR sensor)

When the car approaches the garage gate, an IR sensor triggers frame capture function in the TensorFlow object classification model and processes the parameters pertaining to the car and saves the frame in local folder and a registration email is sent.

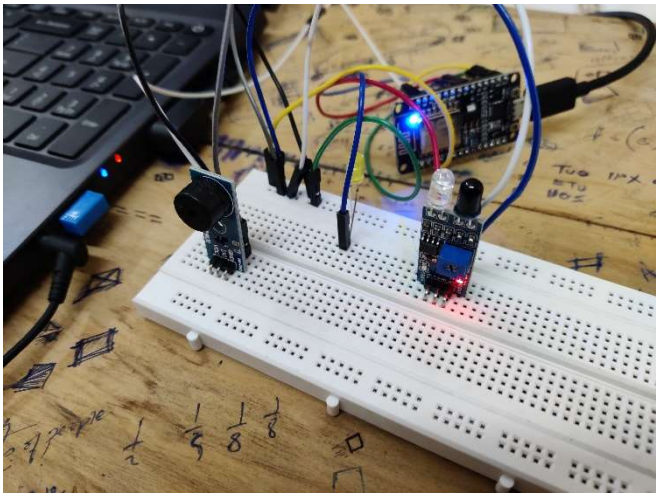
This has been implemented using standardFirmata for Arduino and pyfirmata library to control an Arduino Board through a python code.

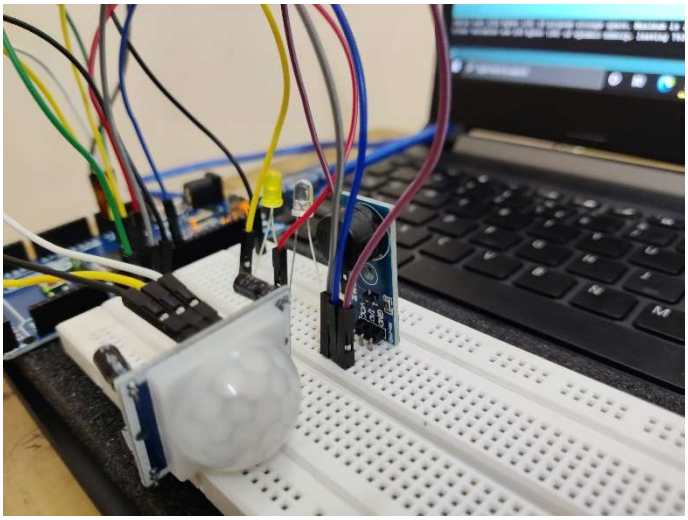


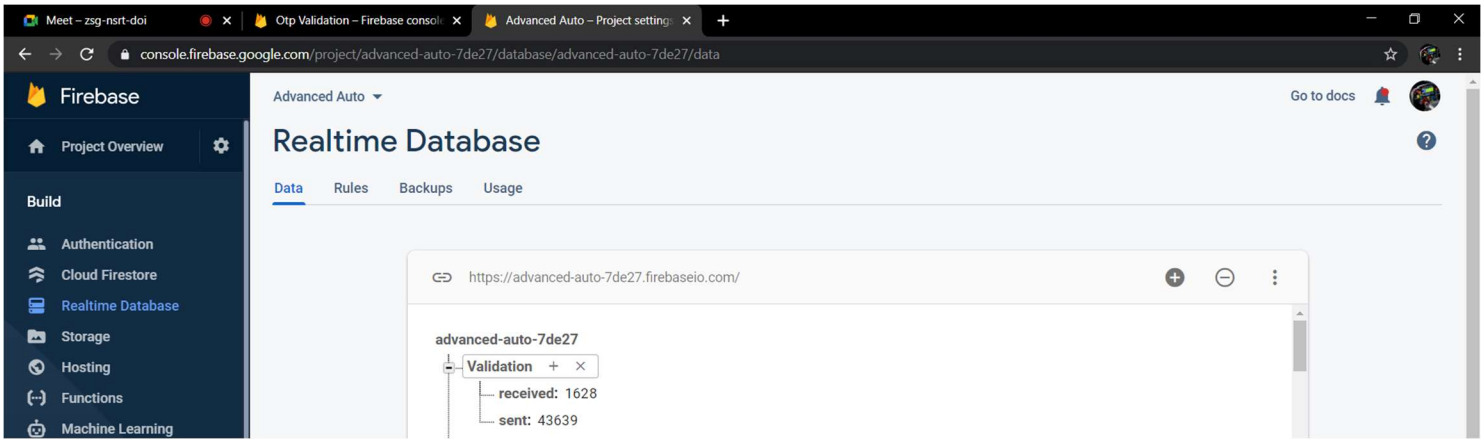


2. Validation and approval to access Car

On successful authentication of an OTP entered by the customer, the motion sensors will be deactivated thus allowing access to car. However unsuccessful authentication triggers an annunciation system.

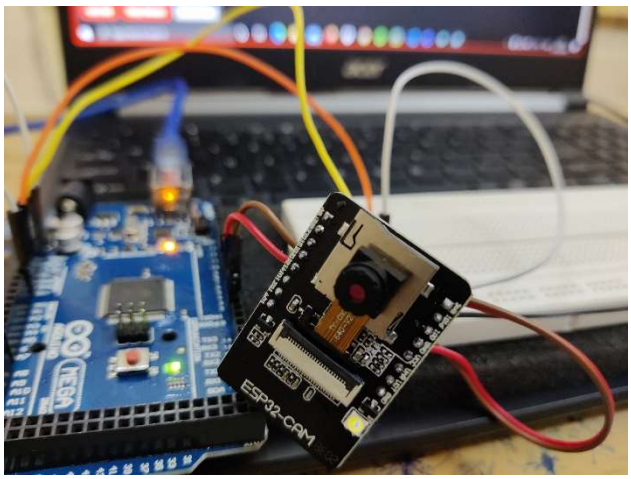






3. Surveillance System Using ESP32 CAM module, Servos and PROCESSING 3 IDE

An ESP32 CAM module can provide real-time surveillance of the car in the garage to an owner sitting in the comfort of his/her home. This is implemented by an ESP32 CAM module mounted on a PAN and TILT servo assembly which can be controlled by Movement of mouse around a GUI. Processing 3 IDE is used to establish a Serial Communication with the Arduino Board. Opencv in Processing IDE generates a GUI and maps pointer position to angle on servos which is sent as a tuple to the Arduino through a serial connection.



processingcode | Processing 3.5.4

File Edit Sketch Debug Tools Help

processingcode

```
import processing.video.*;

import processing.serial.*;

import gab.opencv.*;
import java.awt.*;


Capture video;
OpenCV opencv;

Serial Com5;
float fpan,ftilt;
int pan, tilt, x, y;
int[] inBytes = new int[3];

void setup(){
size(500,500);
delay(500);
String portName = Serial.list()[2];
delay(500);
  Com5 = new Serial(this, portName, 9600);
   video = new Capture(this, 640/2, 480/2);
     opencv = new OpenCV(this, 640/2, 480/2);

     opencv.loadCascade(OpenCV.CASCADE_FRONTALFACE);

  video.start();

//Com5 = new Serial(this, Serial.list()[2], 9600); background(0,0,0);
ellipse(width/2,width/2,10,10);
}

void draw()
```

Console Errors

Type here to search

processingcode | Processing 3.5.4

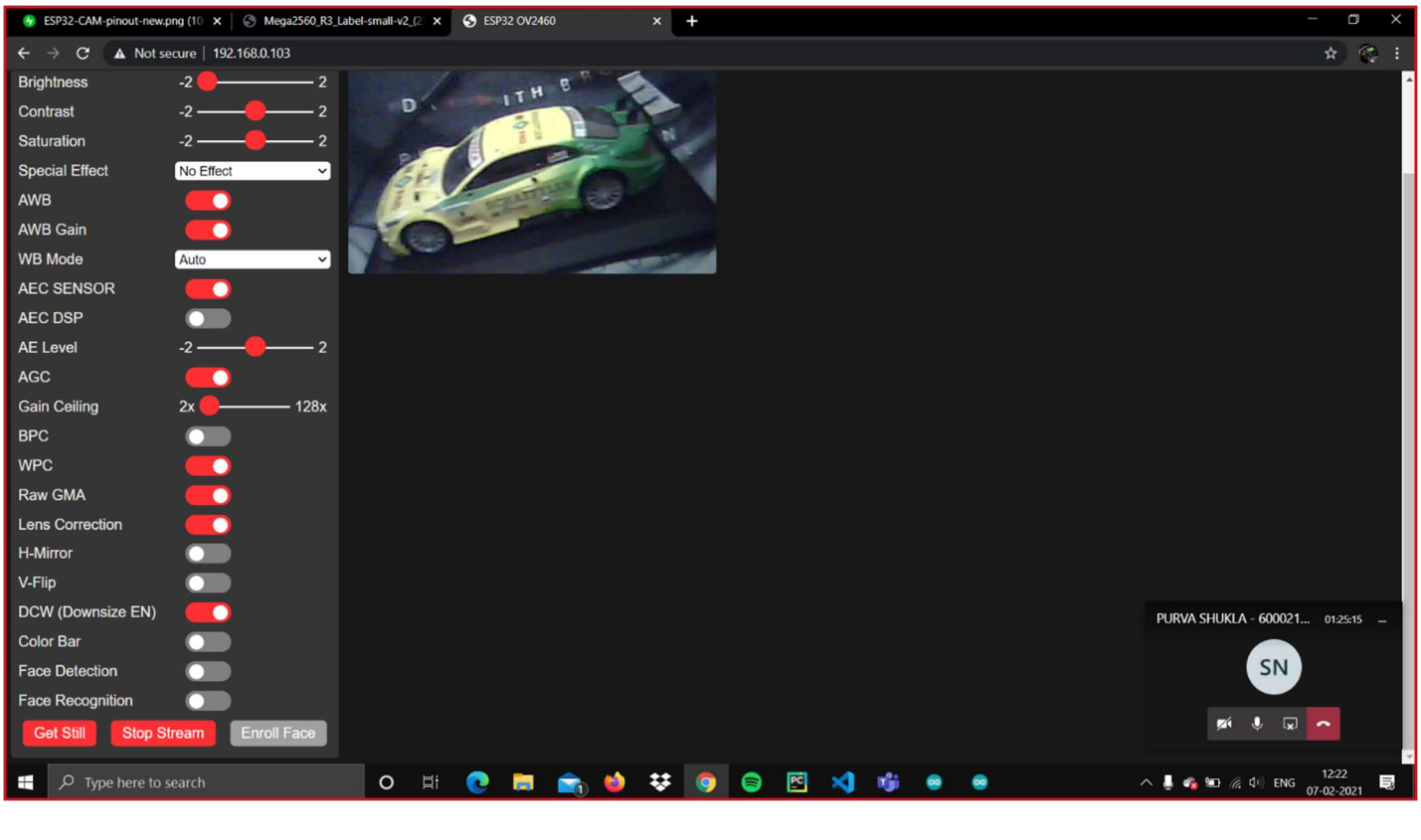
File Edit Sketch Debug Tools Help

processingcode

```
  strokeWeight(3);
    image(video, 0, 0 );

  Rectangle[] faces = opencv.detect();
  println(faces.length);

  for (int i = 0; i < faces.length; i++) {
    println(faces[i].x + "," + faces[i].y);
    rect(faces[i].x, faces[i].y, faces[i].width, faces[i].height);

 pan = int(faces[i].x*180/width);
tilt = int(faces[i].y*180/height);}
pan = constrain(180 - pan,0,180);
tilt = constrain(tilt,0,180);
Com5.write(pan);
Com5.write(tilt);
Com5.write(255);


while (Com5.available()>0)
{



  inBytes[0] = inBytes[1];
  inBytes[1] = inBytes[2];
inBytes[2] = Com5.read();
if(inBytes[2] == 255)
{
  println(inBytes[0] +" , " + inBytes[1]);
//background(0,0,0);
  //ellipse(width - inBytes[0]*width/180,inBytes[1]*width/180,10,10);
}
}
```

Console Errors

Type here to search



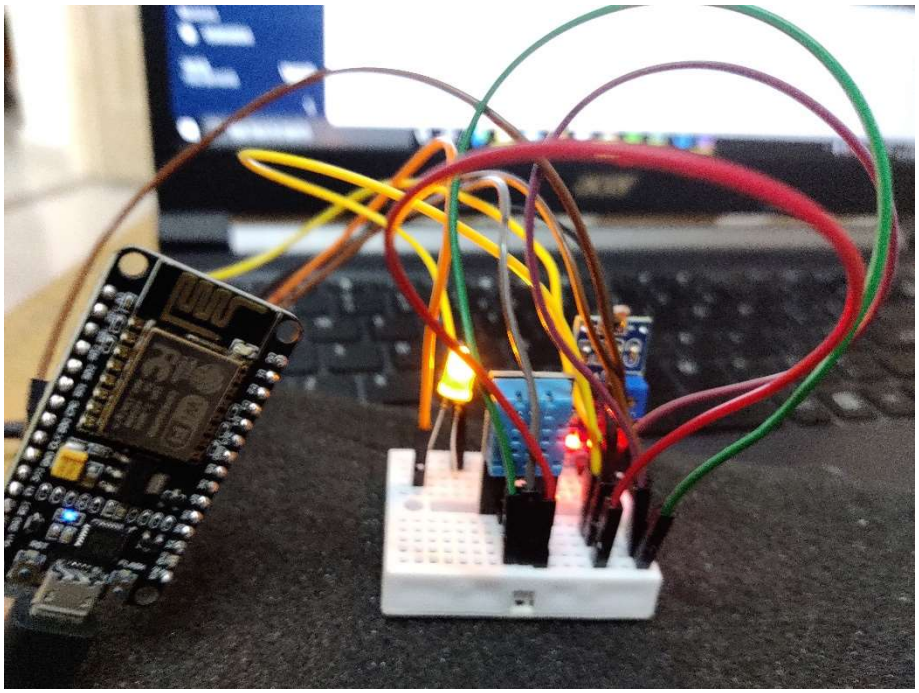
4. Sensor interfacing for Data Telemetry

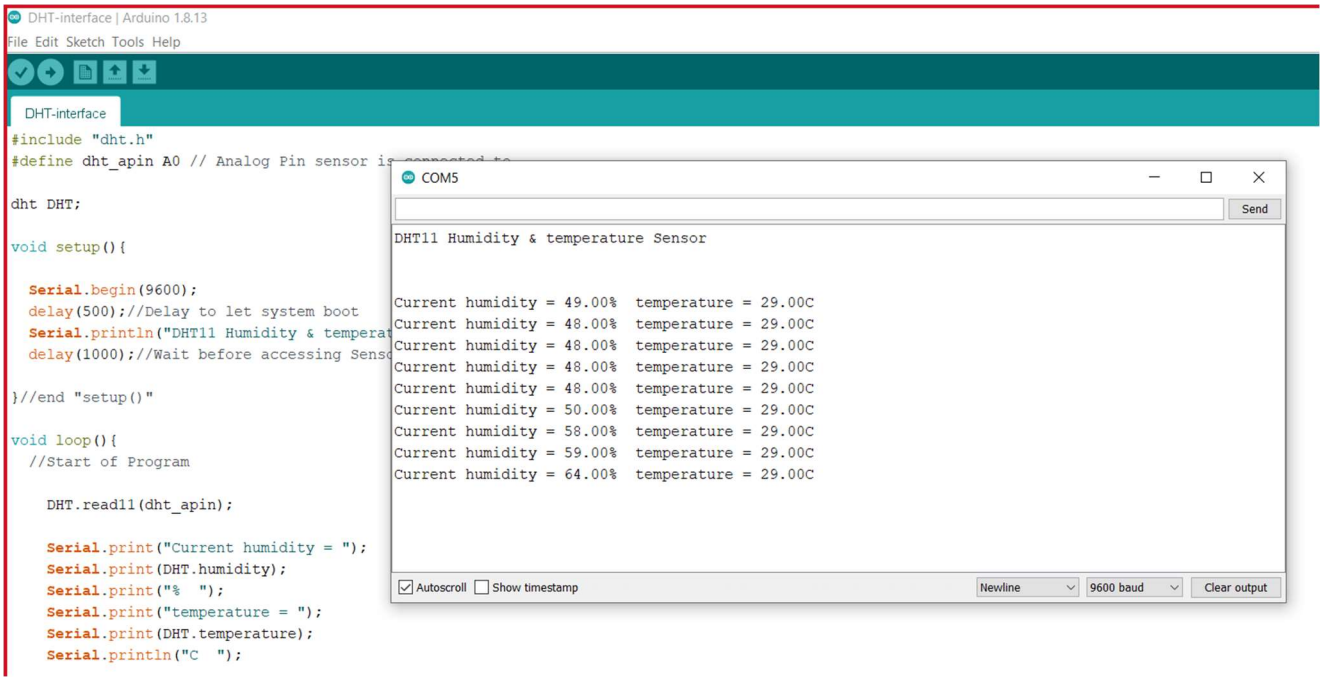
The following sensors are connected to a NodeMCU ESP8266 module for data telemetry through firebase:

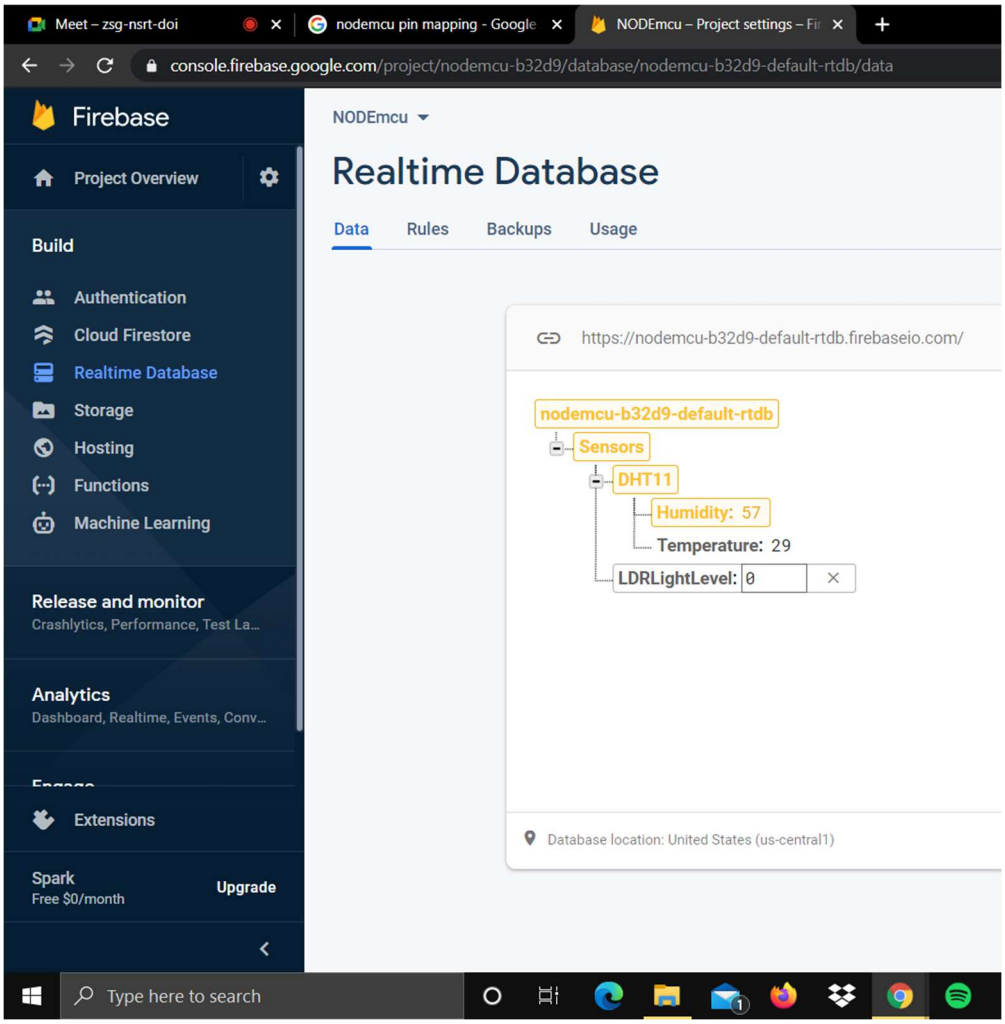
- DHT11 Temperature and Humidity Sensor
- LDR Sensor
- LM393 Speed RPM sensor (Grove Coupler)

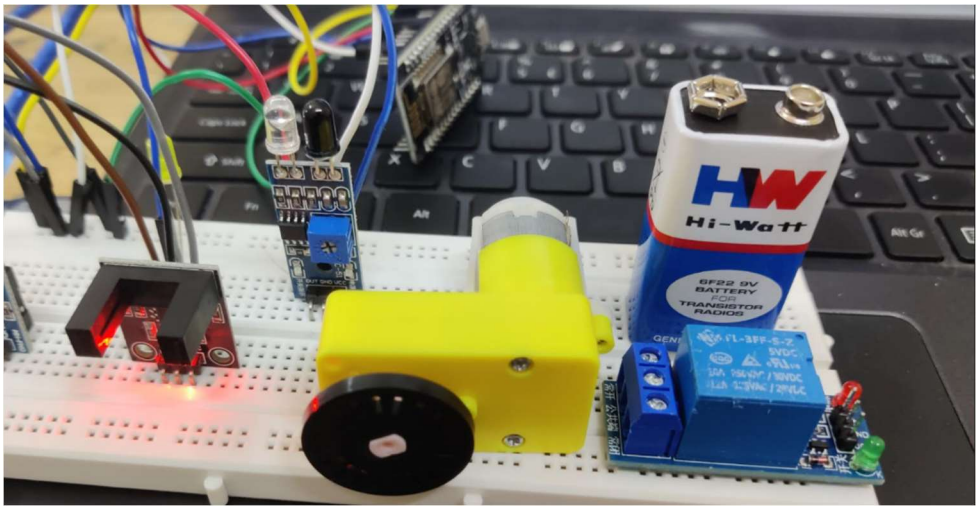
These sensors provide data to the Digital Dashboard GUI and triggers various functions on the steering wheel and screen like auto-lights and Throttle levels based on the speed. Buttons on the GUI can simulate acceleration and control a motor connected to the L293D motor driver and single channel relay. The Grove Coupler returns the RPM of the motor and the value of which is extracted from firebase to set the rev lights on the dashboard.

The speed, RPM and gear values are plotted in a live Plot using library plotly.js and the temperature is plotted on a heatmap relative to time. This helps understand vehicle thermodynamics and dynamics of the engine.




DHT-interface | Arduino 1.8.13
File Edit Sketch Tools Help
DHT-interface
#include "dht.h"
#define dht_apin A0 // Analog Pin sensor is
dht DHT;
void setup(){
Serial.begin(9600);
delay(500);//Delay to let system boot
Serial.println("DHT11 Humidity & temperat
delay(1000);//Wait before accessing Sens
}//end "setup()"
void loop(){
//Start of Program
DHT.read11(dht_apin);
Serial.print("Current humidity = ");
Serial.print(DHT.humidity);
Serial.print("% ");
Serial.print("temperature = ");
Serial.print(DHT.temperature);
Serial.println("C ");
COM5
Send
DHT11 Humidity & temperature Sensor
Current humidity = 49.00% temperature = 29.00C
Current humidity = 48.00% temperature = 29.00C
Current humidity = 48.00% temperature = 29.00C
Current humidity = 48.00% temperature = 29.00C
Current humidity = 48.00% temperature = 29.00C
Current humidity = 50.00% temperature = 29.00C
Current humidity = 58.00% temperature = 29.00C
Current humidity = 59.00% temperature = 29.00C
Current humidity = 64.00% temperature = 29.00C
Autoscroll
Show timestamp
Newline
9600 baud
Clear output


Meet – zsg-nsrt-doi
nodemcu pin mapping - Google
NODEmcu – Project settings – Fi
console.firebase.google.com/project/nodemcu-b32d9/database/nodemcu-b32d9-default-rtdb/data
Firebase
Project Overview
Build
Authentication
Cloud Firestore
Realtime Database
Storage
Hosting
Functions
Machine Learning
Release and monitor
Crashlytics, Performance, Test La...
Analytics
Dashboard, Realtime, Events, Conv...
Extensions
Spark
Free $0/month
Upgrade
NODEmcu
Realtime Database
Data
Rules
Backups
Usage
https://nodemcu-b32d9-default-rtdb.firebaseio.com/
nodemcu-b32d9-default-rtdb
Sensors
DHT11
Humidity: 57
Temperature: 29
LDRLightLevel: 0
Database location: United States (us-central1)
Type here to search



Speed Sensor LM393 with single channel relay