Journal on Internet of Things
DOI: 10.32604/jiot.2022.022200
Article

Dynamic Automation of a Moving Vehicle at Real Time Along with Position and Condition Tracking

Shruti[1], **Soumen Santra**[1] **and Arpan Deyasi**[2,*]

[1]Department of Computer Application, Techno International New Town, Kolkata, 700156,India
[2]Department of Electronics and Communication Engineering, RCC Institute of Information Technology, Kolkata, 700015, India
*Corresponding Author: Arpan Deyasi. Email: deyasi_arpan@yahoo.co.in
Received: 15 November 2021; Accepted: 15 February 2022

Abstract: The present work deals with real-time database updation of moving vehicle through dynamic automation system, where position and internal conditions of the car are auto-saved in the cloud server. The total embedded system is a simulated prototype, already implemented virtually for smaller distance movement on suitable road condition, where the auto-checking of the fuel remained shows the novelty of the system compared to the existing data available in literature. The beneficiary will not only be confined to the owner of the vehicle, but the insurance companies also check the status of it if the car is suffered from accidents or unnecessary external coercive influences. Data display facility is available at real time to avoid any unwanted proximity with nearby moving/standing objects, and later can be retrieved from the secured database. The work may be extended as a complete autonomous tracker with condition monitor in future automobile industry.

Keywords: Dynamic condition monitoring; secured database; moving vehicle; position tracking; auto fuel checking

1 Introduction

Authors Tracking and closet determination of moving vehicle is a subject of investigation in the long past, and research are progressed day-by-day for inclusion of the robust system in the automobile industry, to make it more secure. However, published proposals have different angles, as per the varying requirement of the industry, time-to-time, where priority is only given to vehicle-owner only. This is one of the features of automobile industry, where insurance company is severely neglected. However, recent published articles in several newspapers, as well as published dataset from Govt records in different states of India suggest that rash driving of the drivers is one of the major cause of accidents, but justification of that behaviour at later stage from the side of insurance companies are really difficult. Therefore, win-to-win situation from both the sides are extremely desirable, and that forces the researchers to think differently from practical-point-of-view, with the fundamental motive to reduce accidents, and also a dynamic checking of real-time hazards. Analogous to the hysteresis effect, previous conditions of the car (in the form of dynamic log) is extremely beneficial for both the automobile engineers as well as the other non-technical industries associated with it. Here lies the novelty of the present work.

We all are familiar with the basic principle of car. However, to get a brand-new level of sophistication, understanding of the present working area of the prototype becomes extremely important. For example, knowledge of the previous status of the car is critically important based on its present condition in certain cases. The speed, acceleration, fuel consumption of the past is some important data that is always desirable in case-to-case basis. To make the concept a little clearer, we will take a simple example of insurance company. We know there are a good number of insurance companies that provide security to your vehicle. But the sad part is whether it is the mistake of car user; insurance company has to compensate the loss. So the financial agreement part, we have to take care of that, from the hand of misuse. In the next section, we are briefly summarizing the previous noted works, and their drawbacks.

2 Literature Review

In Mohanpriya et al. [1] the algorithm has been designed to generate a local obstacle map, instead of a global map, because of difficulties in using accurate vehicle position information. The proposed algorithms have been successfully implemented and tested using a test vehicle. Desai et al. [2] has shown the recent trends of research on development of driving assist systems was reviewed. The focus was on collision warning and collision avoidance systems and their impact on driver's comfort, safety and traffic flow. This review of the research on driver assist systems, collision warning and avoidance systems, provides a convenient way of evaluation of the recent research advances in the field. Shetgaonkar et al. [3] described the prototype design of a smart car that can detect humps, potholes and theft, help deal with traffic chaos and provide parking assistance. It is very simple, durable and low in cost. It consumes less power and can be applied in the real world. It can work in bad weather condition too. Gupta et al. [4] has given an extent by taking help of Google map to indicate the nearby fuel station. In this article a prototype was successfully implemented by which parameters of a vehicle were monitored and controlled & also provides the remote control for the user by using smartphones. Gallardo et al. [5] introduced deep learning methods were discussed. Using deep learning techniques such as CNN, AlexNet, and supervised learning, the research shows that autonomous driving can successfully be simulated in a game environment using the deep learning architecture presented in this paper. The results of this experiment show that these principles are ready to be applied to a real world platform in order to experimentally verify its ability to become an everyday part of our lives. Aher et al. [6] shows many strong socio-economic motivators for adopting the smart automobiles such as human safety, infrastructure efficiency, quality of life, physically challenged people as they can use self-driving car for their commuting. Özgüner et al. [7] discussed that despite the need to enhance sensor technology and algorithms beyond the state-of-the-art, the practical performance of today's autonomous vehicles is governed by robustness and improvements due to significant on-terrain testing of hardware and algorithms. The successful teams have mainly employed well-tested hardware and software achieved by sound engineering.

Kaur [8] explains the results were recorded by testing in a simulated environment that had potholes and speed breakers along with all the challenges that the system may have to overcome to ensure a smooth, and safe ride. Kodali [9] shows how Internet of Things (IoT) conceptualizes the idea of remotely connecting and monitoring real world objects (things) through the Internet. When it comes to our house, this concept can be aptly incorporated to make it smarter, safer and automated.

Mandula et al. [10] focuses on building a smart wireless home security system which sends alerts to the owner by using Internet in case of any trespass and raises an alarm optionally. Pavithra et al. [11] proposes an efficient implementation for IoT (Internet of Things) used for monitoring and controlling the home appliances via World Wide Web. Home automation system uses the portable devices as a user interface. They can communicate with home automation network through an Internet gateway, by means of low power communication protocols like Zigbee, Wi-Fi etc. This project aims at controlling home appliances via Smartphone using Wi-Fi as communication protocol and raspberry pi as server system. Soumya et al. [12] has been successfully implemented. The software produced for the project is functionally correct, reasonably robust, and usable. The project has met the entire General and Non-Functional Requirements and in addition, has been implemented in a modular fashion, which can be easily modified or rewritten at a later stage. Valli et al. [13] introduced 2 different technologies. The paper adopted two different technologies namely embedded and android. Embedded technology is used to detect the accident using accelerometer sensor and android technology is used to determine the name of that location instead of latitude and longitude values so that even a layman can understand these values and can know about the vehicle location. An android app that specifies the location name when the mobile receives GPS data plays a major role in the paper. Jesudoss et al. [14] shows how to prevent the drunk & drive accident & it is also prevented from the rash driving accident. This method will prevent from the lots of accident because it will check the driver is drunk or not by using alcohol sensors it will also check whether the driver is getting sleepy or not by using eye blink sensor but this can be applied or proceed only when the car windows are closed. A Kaplan et al. [15] developed a driver's drowsiness detection, alarm and a novel semi-automatic parking system to prevent road crashes. In addition, it has extended the work to report the nearby police station of the occurrence. Automatic prevention is accident is recently reported with working prototype model [16] which shows the accuracy as well as efficiency to the satisfactory level.

3 Proposed Design

The proposed design for autonomous car security and condition monitoring system is represented in Fig. 1 where six (6) sensors (ultrasonic sensor, humidity sensor, temperature sensor, infrared (IR) LED, ultrasonic range finder, soil moisture sensor) are connected to Arduino mega board along with NRF24L01 - 2.4G Wireless Transceiver Module.

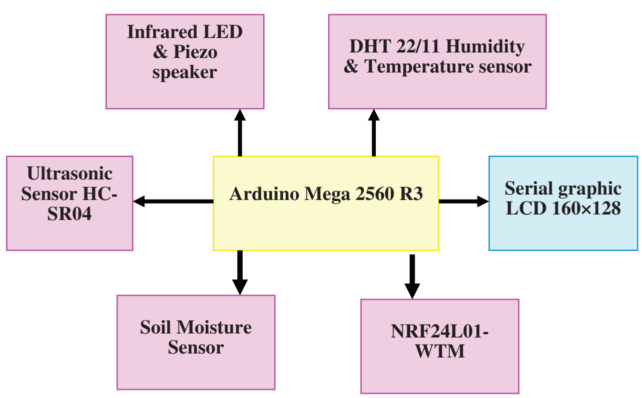

Figure 1: Schematic diagram of the set-up

Ultrasonic Sensor here is used to measure the quantity of fuel in fuel tank. Humidity and Temperature Sensor is used to serve dual purpose, one to measure the temperature of the engine, and 2nd one is used to measure the temperature of the inside of the car. IR LED detects the distance between the car and obstacle. It detects the distance and activates the break beforehand. It can also help in parking, but it doesn't work in water, so during rain, ultrasonic range finder is utilized. It measures distance by emitting a pulse of ultrasonic sound that travels through the air until it hits an object. It helps to reduce accident. Soil moisture sensor detects the rain. One relay is also attached with the relay so that as soon as soil moisture detects rain, the wipers are automatically switched on. NRF24L01 - 2.4G is used to detect the tyre pressure that is calculated by tyre pressure meter. Corresponding circuit diagram is represented in Fig. 2.

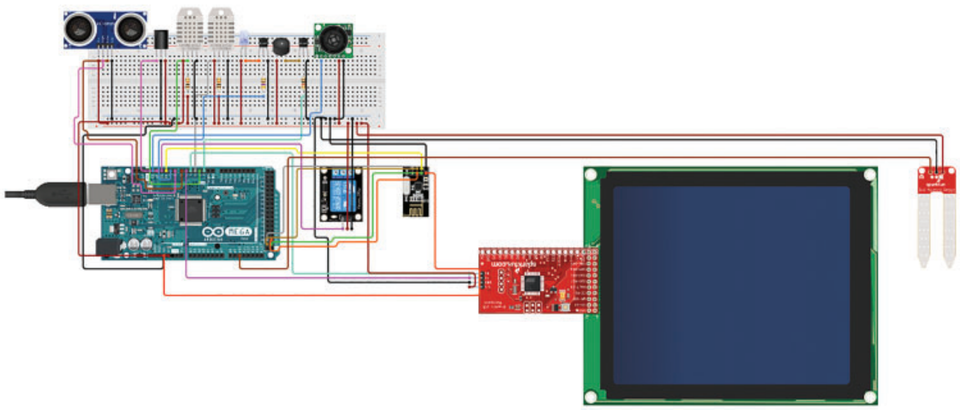

Figure 2: Circuit diagram for the proposed design

4 Workflow

In this section, detailed algorithm of each sensor is discussed with functionalities.

4.1 Algorithm for Ultrasonic Sensor

At first, algorithm for the ultrasonic sensor is mentioned below which not only calculates distance with the other objects, but also checks the status of fuel tank at running condition:

Step 1: Start

Step 2: Initialize object

New Ping csr04(HCSR04_PIN_TRIG,HCSR04_PIN_ECHO)

Step 3: Call setup1()

Step 4: Initialize int hcsr04Dist

Step 5: Open Serial port

Step 6: Set hcsr04Dist = hcsr04.ping_cm () for each iteration of 1 s. Step 6: Print ("fuel tank level is" + hcsr04Dist)

Step 7: Close serial port

Step 8: End

In this algorithm, we will first initialise the object New Ping hcsr04. This object is used to detect ultrasonic sensor. Here we have initialised our sensor. Next in step 4, we have initialised an internal variable int hcsr04Dist to store the distance value, where the function hcsr04.ping_cm () is used to calculate the fuel level in the fuel tank. In step 5, the distance calculated by the function hcsr04.ping_cm (), will be stored in the variable hcsr04Dist. It will calculate the distance again in next 1 s. This iteration will continue till we stop the simulation. In step 6, we will display the fuel tank level.

4.2 Algorithm for Humidity and Temperature Sensor

Next the algorithm for humidity and temperature sensor is given:

Step 1: Start

Step 2: Initialise object DHT dht_1(DHT_1_PIN_DATA)

Step 3: Call setup2()

Step 4: Initialise variable float dht_1Humidity, dht_1TempC

Step 5: Open serial port

Step 6: Set dht_1Humidity = dht_1.readHumidity(), dht_1TempC = dht_1.readTempC()

// to read the data of inside the car

Step 7: Print (“Humidity: ” + dht_1.readHumidity + “%”) Print (“Temp :”+ dht_1.readTempC+ “°C”)

// Display data on LCD

Step 8: Close Serial port

Step 9: End

In the above given algorithm, we have first initialised the object, DHT dht_1(DHT_1_PIN_DATA). It is used to read the pin taken by sensor in the Arduino board. Then we will initialise two float variables, dht_1Humidity, dht_1TempC, to store the humidity and temperature data. In the next step, we open the serial port, so that each value calculated can be displayed. In step 6, we will set dht_1Humidity = dht_1.readHumidity(). dht_1 Humidity will store the humidity data calculated by the sensor. Similarly, we will set dht_1TempC = dht_1.readTempC(), to store the temperature calculated by the sensor in Celsius. Then in step 7, we will print both the humidity and temperature data. After printing the data, we will close the serial port and end the program.

4.3 Algorithm for IR LED

Step 1: Start

Step 2: Initialise objects IRsend ir_led and IR ir(IR_PIN_OUT) //for sender and receiver

Step 3: Call setup3() //For IR Sender

Step 4: Transmit IR code as ir_led.sendSony(0xa, 12) //transmit custom IR using Sony convention

Step 5: Set delay (1000) //waits 1 s until next send for IR Receiver

Step 6: Open serial port

Step 7: initialise long irCode

Step 8: Set irCode = ir.detect() // ir.detect() takes data from IR sender

Step 9: if (irCode is found)

Print (“0x”), Print (irCode, HEX)

End if

Step 10: Close serial port

Step 11: End

In this algorithm, we have first initialised two objects, IRsend ir_led and IR ir(IR_PIN_OUT), for sending and receiving wavelengths respectively. Then we have done different steps for sender and receiver. For sender, IR data is sent. Here we have used ir_led.sendSony(0xa, 12), for custom IR code as per sony convention per 1 s. For more routines we can refer to IRremote.h such as sendRaw, sendRC5. For receiver, in step 5, serial port for giving the output is opened. In step 7, we have initialised long irCode. In step 8, if code is detected, it will be stored in irCode. If code is found, it will print 0x + Hexcode. Once output is printed, serial port will be closed.

4.4 Algorithm for Ultrasonic Range Finder

Step 1: Start

Step 2: Initialise object UltraSonic ultraSonic_5v(ULTRASONIC_5 V_PIN_PW),

PiezoSpeaker piezoSpeaker_5v(PIEZOSPEAKER_5 V_PIN_SIG)

Step 3: Initialise unsigned int piezoSpeaker_5vHoorayLength = 6;

// amount of notes in melody

initialise unsigned int piezoSpeaker_5vHoorayMelody[] =

{NOTE_C4, NOTE_E4, NOTE_G4, NOTE_C5, NOTE_G4, NOTE_C5};

// list of notes. List length must match HoorayLength!

Initialise unsigned int piezoSpeaker_5vHoorayNoteDurations[] = {8, 8, 8, 4, 8, 4};

Step 4: Initialise variable float sensor, mm, cm, inches, feet

Step 5: Call setup4 (), loop ()

Step 6: End

In this algorithm, we have first initialised two objects, UltraSonic ultraSonic_5v, PiezoSpeaker piezoSpeaker_5v, for the pins of LV- Maxsonar, and piezo speaker. In step 3, we will initialise the variables for piezo speaker. We have first initialised unsigned int piezoSpeaker_5vHoorayLength as 6, to set the amount of notes in a melody. Then, we have initialised unsigned int piezoSpeaker_5vHoorayMelody [] as {NOTE_C4, NOTE_E4, NOTE_G4, NOTE_C5, NOTE_G4, NOTE_C5}. We have to take care that note list should match the number of nodes. Then we have initialised the durations as unsigned int piezoSpeaker_5vHoorayNote Durations [] = {8, 8, 8, 4, 8, 4}. In step 4, we have initialised the LV-maxsonar variables as float sensor, mm, cm, inches, feet.

4.5 Algorithm for Soil Moisture Sensor

Step 1: Start

Step 2: Initialise object SoilMoisture soilMoisture_5v(SOILMOISTURE_5 V_PIN_SIG), Relay relayModule(RELAYMODULE_PIN_SIGNAL)

Step 3: Call setup5(), check()

Step 4: Set delay(1000) //1000 = 1 s

Step 5: End

In the above algorithm, we have initialised two objects, SoilMoisture soilMoisture_5v and Relay relay Module for the pin of soil moisture sensor and relay module. Then in step 3, we will call check () function that we have explained above to display the soil and moisture level and activate the car wipers. This function will be called at the delay of 1 s till power is given to the Arduino board for this case.

4.6 Algorithm for Wireless Transceiver Module

Step 1: Start

Step 2: Initialise two objects, OBJECT1 (NRF24L01_PIN_CE 10) and OBJECT2 (NRF24L01_PIN_CS)

Step 3: Set const byte address [6] = "00001"

Step 4: Call setup6() and loop2()

Step 5: End

In the above given algorithm, we have first initialised two objects for two pins (CE and CS) of the NRF module. Then we have set the common address through which NRF can communicate. In step 4, we will call function setup6(), loop2() to complete the program.

4.7 Arduino Main Board Algorithm

Step 1: Start

Step 2: define pins DHT_1_PIN_DATA, DHT_2_PIN_DATA, GRAPHICLCD_PIN_TX, GRAPHICLCD_PIN_RX, HCSR04_PIN_TRIG, HCSR04_PIN_ECHO, NRF24L01_PIN_CE, NRF24L01_PIN_CS PIEZOSPEAKER_5 V_PIN_SIG, RELAYMODULE_PIN_SIGNAL, SOIL-MOISTURE_5 V_PIN_SIG A10, IR_PIN_OUT ULTRASONIC_5 V_PIN_PW

Step 3: Initialise int x

Step 4: Print (1. For ultrasonic sensor, 2. For humidity and temperature sensor (inside the car), 3. For humidity and temperature sensor (inside the car engine), 4. For IR LED, 5. For LV-Maxsonar, distance range calculator, 6. For soil and moisture sensor, 7. For NRF24L01 - 2.4G Wireless Transceiver Module)

Step 5: Take input x

Step 6: Pass switch (x)

Step 7: Use the above given algorithms for each case.

Step 8: End

We have mentioned the algorithm for all the sensors and for all the functions related to sensors. This is the algorithm of arduino mega board where all the functions will be used. In step 2, we will define all the pins related to each sensor. In step 3, we have initialised a variable x that will be used for switch case. In step 4, we have given the user to choose from the sensors. In step 5, the user will input the value. In step 6, we will pass the switch case. The algorithms that we had mentioned before, will be placed in each switch case from 1 to 7. The default will be "wrong input". In this way, each function related to each sensor can be used here.

5 Results

Based on the algorithms described in the previous section, the present proposal, as described in Fig. 1, is successfully implemented as per the circuit depicted in Fig. 2. In this section, first responses of the individual sensors are simulated. Results of individual responses are mentioned below followed by overall system responses.

Fig. 3 shows the data variation of ultrasonic sensor as measured in real time. The plot calculates amount of fuel left in the fuel tank of a car. The plot suggests that reading are in approximately descending order, which speaks about correct working of the sensor, as the fuel level has been decreased due to the consumption by car's engine.

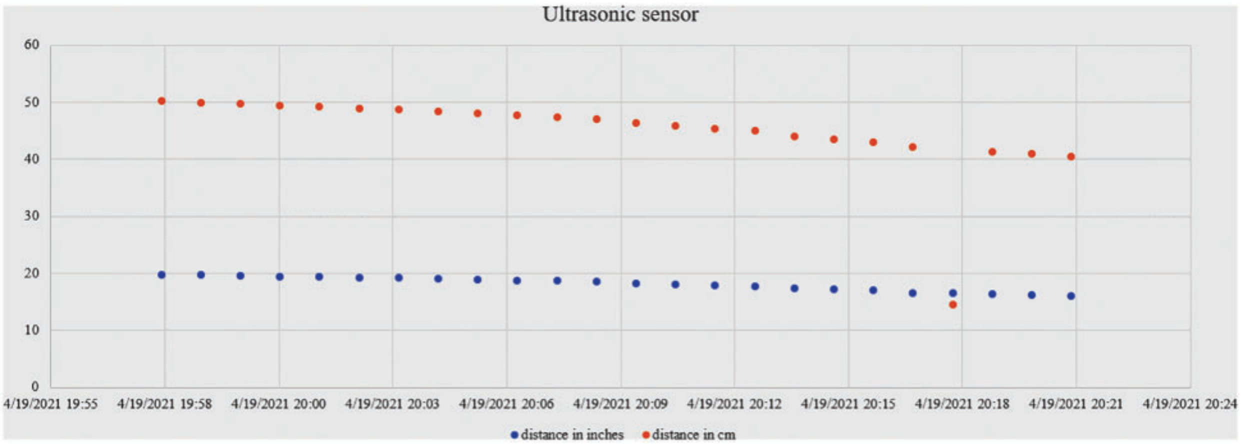

Figure 3: Results of ultrasonic sensor for fuel consumption

Fig. 4 shows the data obtained from humidity and temperature sensor. It is an electronic device that measures and reports both moisture and air temperature of the surrounding environment where they are deployed. The measurement indicates the concentration of water vapor presented in the air.

A novelty of the proposed circuit is simultaneous measurement of the data both inside and outside of the car. In contrast to the scattered usual graph of Fig. 4a, a steady rise is observed in Fig. 4b, which speaks about the internal environment of the vehicle. The advantage is that the driver will get the information about engine overheating.

The IR LED does distance or proximity sensing through emitting IR beam and calculating angle of reflection. So the reading depends on the objects/obstacles present on the road while driving, and results are therefore, definitely random numbers.

Ultrasonic range finder offers very short to long range detection and ranging, in an incredibly small package with ultra-low power consumption. The results are illustrated in Fig. 5. The interface output formats included width output, analog voltage output, and a synchronous serial digital output. It balances the detection of people with a narrow beam width.

Since the direct gravimetric measurement of free-soil moisture requires removing, drying, and weighing of a sample, soil moisture sensors measure the volumetric water content indirectly by using some other property of the soil, such as electrical resistance, dielectric constant, or interaction with neutrons, as a proxy for the moisture content. The relation between the measured property and soil moisture must be calibrated and may vary depending on environmental factors such as soil type, temperature, or electric conductivity. The corresponding results can be seen from Figs. 6–7.

Wireless Transceiver Module is designed to operate in 2.4 GHz worldwide ISM frequency band and uses GFSK modulation for data transmission. The data transfer rate can be one of 250 kbps, 1 and 2 Mbps. Here, this module is also used to calculatethe tyre pressure of the car.

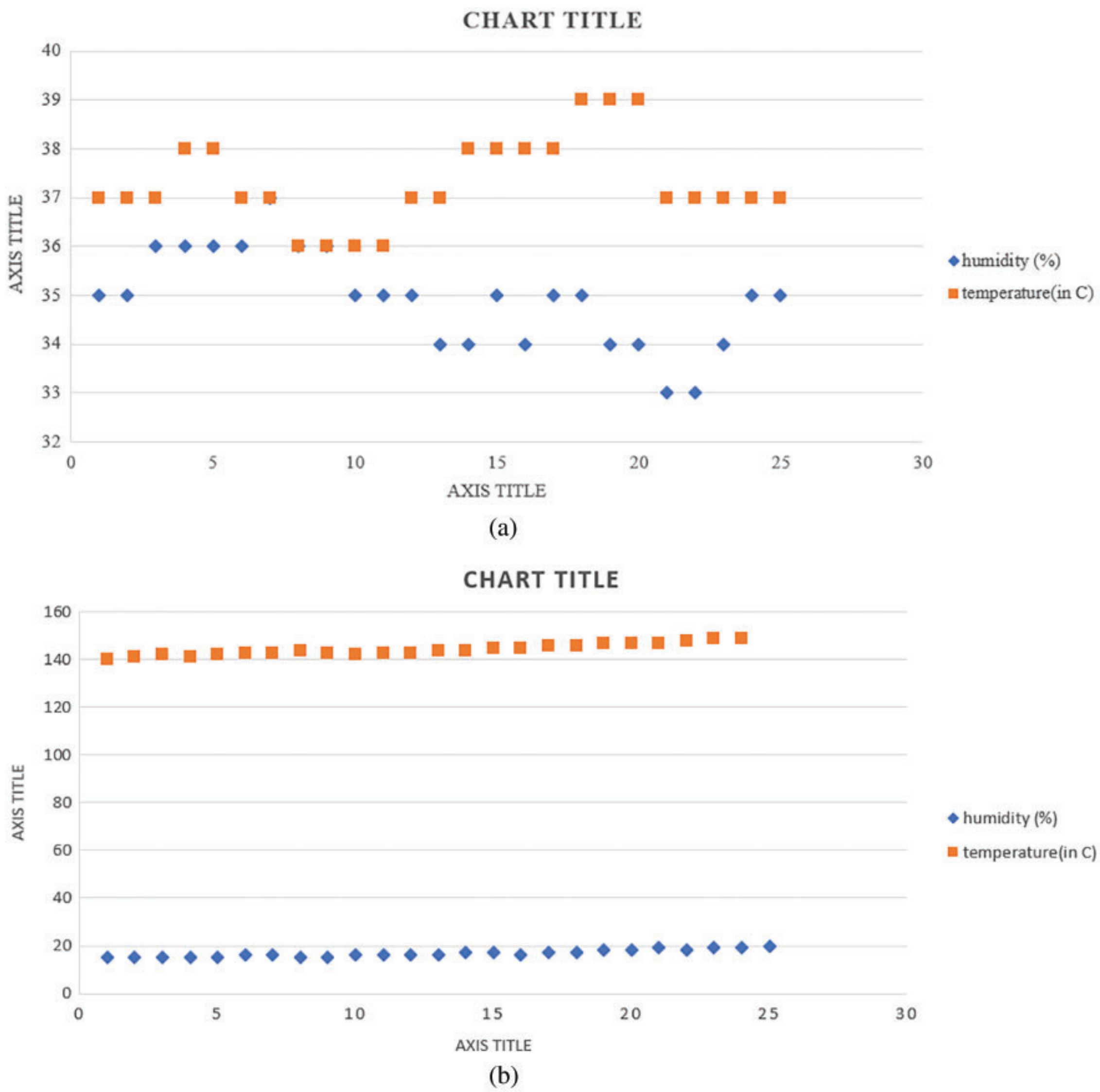

Figure 4: a: Results of humidity and temperature sensor outside the car b: Results of humidity and temperature sensor inside the car

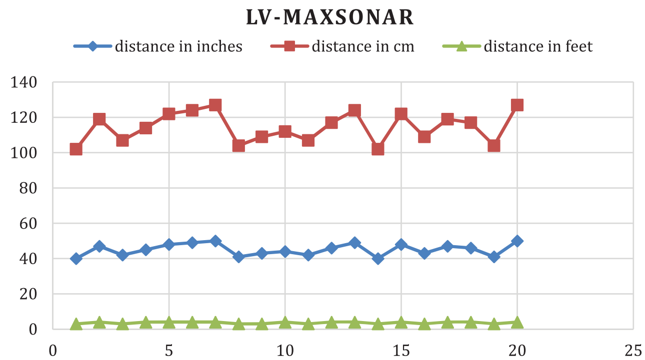

Figure 5: Results of ultrasonic range finder

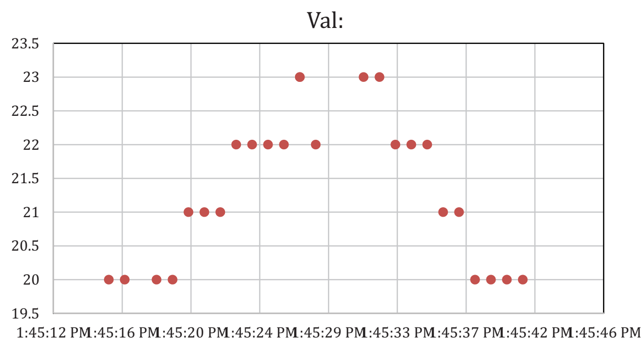

Figure 6: Results of soil moisture sensor

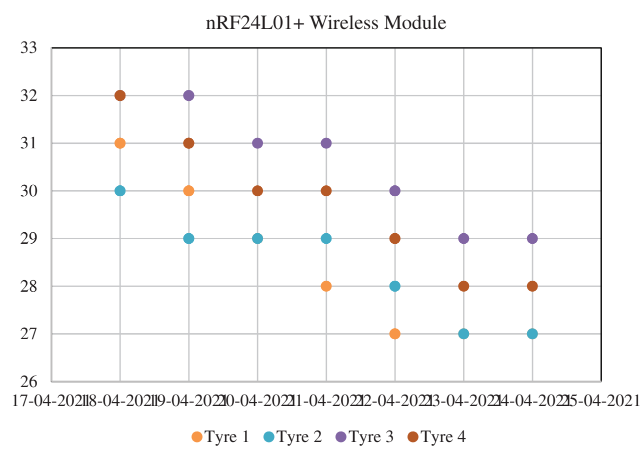

Figure 7: Results of wireless transceiver module

6 Conclusion

The present autonomous system works in favor of both owners for dynamic upgradation of real-time data which are important for monitoring, as well as the insurance company whenever the vehicle undergoes any undesired event. The simulated results for all the sensors clearly indicate the fact that the system is working fine as per the requirement with mutual coupling between them. All the algorithms are developed by the present group of authors, and the sub-functions actually makes interdependency between all the circuit elements. The idea is novel, and may be improved in future with inclusion of machine learning algorithm for classification of real-time data and corresponding automatic decision making.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] R. Mohanapriya, L. K. Hema, D. K. Yadav and V. K. Verma, “Driverless intelligent vehicle for future public transport based on GPS,” *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 3, no. 3, pp. 378–384, 2014.
[2] S. Desai and S. Desai, “Smart vehicle automation,” *International Journal of Computer Science and Mobile Computing*, vol. 6, no. 9, pp. 46–50, 2017.
[3] A. Shetgaonkar, T. Shirodkar, R. Yadavade and S. Raykar, “Smart Car,” *International Research Journal of Engineering and Technology*, vol. 4, no. 6, p. 591, 2017.
[4] V. Gupta, V. Mane, M. R. Pradhan and K. B. Kotangale, “IOT based Car automation using raspberry Pi,” *Imperial Journal of Interdisciplinary Research*, vol. 3, no. 4, pp. 1445–1448, 2017.
[5] N. Gallardo, N. Gamez and M. Jamshidi, “Autonomous decision making for a driver-less Car,” in *IEEE 12th System of Systems Engineering Conf.*, Waikoloa, HI, USA, pp. 18–21, June 2017.
[6] S. Aher, P. Bakaya and R. Mulla, “A prototype of autonomous vehicle using stereoscopic vision,” *International Journal of Computer Science and Mobile Computing*, vol. 6, no. 4, pp. 228–233, 2017.
[7] Ü. Özgüner, K. Redmill and C. Stiller, “Systems for safety and autonomous behaviour in cars: The DARPA grand challenge experience,” *Proceedings of the IEEE*, vol. 95, no. 2, pp. 397–412, 2007.
[8] G. I. Kaur, S. Joshi, A. Kaur and S. Kaur, “Going driverless with sensors,” *International Journal of Engineering Research & Technology*, vol. 3, no. 10, pp. 1–3, 2015.
[9] R. K. Kodali, V. Jain, S. Bose and L. Boppana, “IoT based smart security and home automation,” in *IEEE Int. Conf. on Computing, Communication and Automation*, Greater Noida, India, pp. 29–30, April 2016.
[10] K. Mandula, R. Parupalli, C. H. A. S. Murty, E. Magesh and R. Lunagariya, “Mobile based horne automation using internet of things(IoT),” in *IEEE Int. Conf. on Control, Instrumentation, Communication and Computational Technologies*, Kumaracoil, India, pp. 18–19, Dec 2015.
[11] D. Pavithra and R. Balakrishnan, “IoT based monitoring and control system for home automation,” in *IEEE Proc. of 2015 Global Conf. on Communication Technologies*, Thuckalay, India, pp. 23–24, April 2015.
[12] S. Soumya, M. Chavali, S. Gupta and N. Rao, “Internet of things based home automation system,” in *IEEE 1st Int. Conf. on Recent Trends in Electronics Information Communication Technology*, Bangalore, India, pp. 20–21, May 2016.
[13] B. A. Valli and P. Jonnala, “Vehicle positioning system with accident detection using accelerometer sensor and andriod technology,” in *IEEE Int. Conf. on Technological Innovations in ICT for Agriculture and Rural Development*, Chennai, India, pp. 7–8, April 2017.
[14] A. Jesudoss, B. O. Muthuram and A. L. Emmanuel, “Safe driving using IoT sensor,” *International Journal of Pure and Applied Mathematics*, vol. 118, no. 20, pp. 3745–3750, 2018.
[15] S. Kaplan, M. A. Guvensan, A. G. Yavuz and Y. Karalurt, “Driver behaviour analysis for safe driving: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3017–3032, 2015.
[16] S. Sil, S. Daw and A. Deyasi, “Smart intelligent system design for accident prevention and theft protection of vehicle,” in *SPRINGER: Lecture Notes in Electrical Engineering: 5th Int. Conf. on Nanoelectronics, Circuits & Communication Systems, Chapter 47*, Singapore, pp. 523–530, 2021.