Article

Not peer-reviewed version

Implementation of an Integrated System for Preventive Maintenance Management and Alerts in Light Vehicles

Joseph Barreiro-Zambrano , Juan Martinez-Parrales , Roberto López-Chila *

Posted Date: 25 March 2026

doi: 10.20944/preprints202603.1980.v1

Keywords: preventive maintenance; Arduino; GPS; GSM; vehicle telemetry; road safety

 doi:10.20944/preprints202603.1980.v1

Article

Implementation of an Integrated System for Preventive Maintenance Management and Alerts in Light Vehicles

Joseph Barreiro-Zambrano, Juan Martinez-Parrales and Roberto López-Chila *

GITICEA and GMOVINT Research Groups, Automotive Engineering Career, Politecnica Salesiana University, Guayaquil 090109, Ecuador

* Correspondence: rlopezch@ups.edu.ec

Abstract

Inadequate vehicle maintenance management is one of the main causes of road accidents and elevated operating costs in light vehicles. This paper addresses this problem through the development and implementation of a low-cost integrated system for preventive maintenance management and alerts. The device, based on an open-hardware architecture (Arduino Mega 2560), integrates Global Positioning System (GPS) and mobile communication (GSM/LTE) modules to monitor distance traveled in real time and notify the user via SMS about the proximity of critical services such as oil changes, brake inspections, and timing-belt replacements. Experimental validation was conducted in the city of Guayaquil using a 2012 Hyundai Accent. Field tests were carried out in three scenarios: a dense urban route, a peripheral road, and interurban routes. Results showed satisfactory accuracy with a global average percentage error of 3.98% compared to the vehicle's odometer, and 100% effectiveness in sending alerts. It is concluded that the proposed system is a viable and reliable technological solution to mitigate the "forgetfulness factor" among private drivers, improving road safety and vehicle lifespan.

Keywords: preventive maintenance; Arduino; GPS; GSM; vehicle telemetry; road safety

---

1. Introduction

In the technical management of light vehicles, preventive maintenance represents a strategy aimed at anticipating mechanical failures and keeping operational performance within safe parameters. At the global level, inadequate management of service intervals is a precursor to critical mechanical failures, which represent a significant cause of road accidents [1]. In the Ecuadorian context, this problem acquires a structural dimension due to the characteristics of the vehicle fleet. According to the Annual Report of the Automotive Sector by AEADE, the national vehicle fleet shows a marked aging trend: 75% of vehicles are more than six years old, and 60% exceed eleven years of use [2].

This statistical reality reveals a substantial technological gap. The vast majority of these vehicles, manufactured before the mass adoption of consumer telematics, lack native maintenance-alert systems, forcing owners to rely on manual methods that are prone to the "forgetfulness factor" [3]. Technological obsolescence not only increases operating costs due to corrective repairs [4], but also has direct environmental implications. National regulations such as RTE-002 of INEN [5] establish strict emission control limits; however, the technical literature warns that internal combustion engines are primary sources of harmful substances such as carbon monoxide (CO), carbon dioxide (CO_2), and nitrogen oxides (NO_x) [6]. Consequently, when these engines operate without adequate maintenance (oil changes, filters, or spark plugs), their operation deviates from optimal design parameters, compromising regulatory compliance and worsening air quality.

Although recent literature proposes OBD-II-based monitoring solutions and cloud-based platforms [7,8], these typically present cost and compatibility barriers for the average user of a conventional M1 category vehicle [9]. There is therefore a need to develop accessible aftermarket solutions.

Additionally, recent studies have shown that real urban traffic conditions in congested cities such as Guayaquil exhibit high dynamic variability and a complex urban morphology characterized by high building density and "urban canyon" effects that hinder the propagation of wireless signals [10]. This limits the representativeness of measurements obtained under controlled scenarios or standardized cycles, making experimental validation of vehicular systems—especially those relying on GPS/GSM telemetry—necessary directly on real traffic routes [6].

This work addresses this need through the implementation of an integrated preventive maintenance management and alert system. The device uses an open-hardware architecture (Arduino Mega) and GSM (Global System for Mobile Communications) communication via SMS (Short Message Service), technologies selected for their high availability and low operating cost [11]. The objective is to validate a prototype capable of calculating the distance traveled via GPS and autonomously notifying the user about the proximity of critical services, bridging the technological gap in the vehicle fleet and contributing to road safety.

The main objective of this study is to develop and experimentally validate a functional low-cost prototype that optimizes preventive vehicle maintenance management. As its primary contribution, the work integrates accessible technologies with a quantitative technical validation approach, evaluating the precision of distance measurement, the effectiveness of the generated alerts, and system latency under real operating conditions.

2. Materials and Methods

The materials used and the methodological procedure for developing and validating the proposed system are presented below.

2.1. Research Design

The research was structured under a pretest–posttest comparative scheme, corresponding to a quasi-experimental design without random assignment, in accordance with the methodological guidelines described for applied studies in real contexts where experimental manipulation of environmental variables is not possible [12].

The pre-intervention phase consisted of recording mileage conventionally via the vehicle odometer, used as an instrumental reference. The post-intervention phase comprised the implementation of the proposed electronic system, enabling automatic measurement of distance traveled via GPS and the scheduled generation of preventive maintenance alerts.

The vehicle, habitual routes, and general operating conditions were kept constant during both phases, allowing comparison of instrumental behavior before and after the technological intervention, focusing on measurement coherence and the functional performance of the implemented system.

The unit of analysis consisted of an M1 category light vehicle, and the experimental period lasted approximately five weeks, encompassing both phases of the experimental process.

2.2. Methodological Approach Oriented Toward Operational Optimization

Preventive maintenance optimization is approached from an operational perspective, understood as an improvement in the reliability and timeliness of the information used to schedule technical interventions. In this sense, digitizing monitoring constitutes a key element for reducing uncertainty in planning and strengthening data-driven decision-making. Recent studies on condition-based maintenance indicate that the incorporation of digital tracking systems helps improve operational efficiency by providing continuous and verifiable information [13]. It has also been reported that the integration of electronic monitoring tools can increase control precision and reduce unplanned events in maintenance processes [14].

Consistent with this conceptual framework, the developed system was designed to automate mileage control, reduce dependence on manual logging, and improve the timeliness of service alerts. The implemented operational sequence and the decision criteria adopted are represented schematically in Figure 1. The optimization considered therefore corresponds to a functional improvement in preventive maintenance control, supported by the availability of quantifiable real-time information.

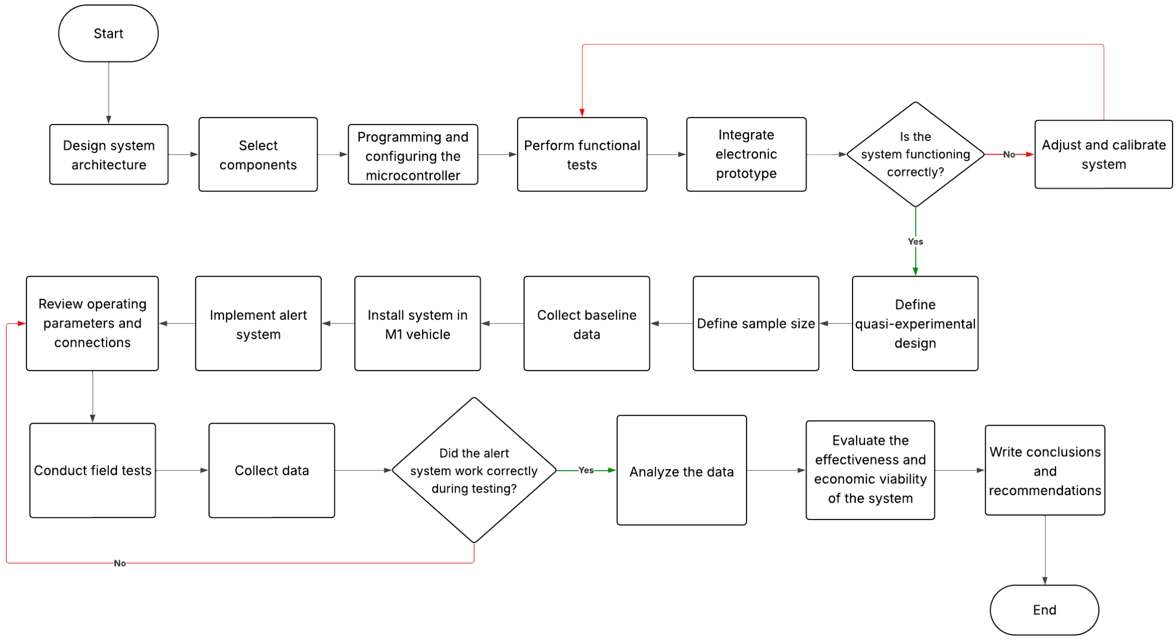

Figure 1. Methodological flowchart of the system.

2.3. Unit of Analysis and Testing Environment

Experimental tests were carried out in the city of Guayaquil, Ecuador, selected for its heterogeneous traffic conditions and dense urban topography. According to prior studies on driving cycles in this locality, the variability of stops and the built environment create a challenging scenario for satellite signal stability, known as the "urban canyon" effect [6,7]. These conditions allow evaluation of the robustness of the prototype's filtering algorithm against momentary precision losses.

The unit of analysis consisted of an M1 category light vehicle (sedan). A 2012 Hyundai Accent with an initial odometer reading of 145,230 km was selected, as it is representative of the majority segment of the national vehicle fleet according to AEADE [2]. For distance validation, the vehicle's digital odometer was established as the reference instrument. This decision is grounded in the functional coherence of the system: since maintenance intervals are defined and perceived by the user from the dashboard reading, the relevant comparison metric is the odometer and not an absolute geodetic measurement. To ensure that the odometer did not introduce a disproportionate error into the validation, the physical variable of greatest impact was controlled: the dynamic wheel radius. Prior to each test, tire pressure was standardized at 30 PSI, mitigating variations due to rubber deformation [15].

Furthermore, the legal acceptance tolerance for commercial odometers, established at 4% by the National Institute of Standards and Technology (NIST) [16], was considered as a control limit for evaluating prototype performance.

2.4. Materials and System Components

The system is based on a modular open-hardware architecture, validated in the literature for its cost-effectiveness [17]. An Arduino Mega 2560 microcontroller was used, chosen for its capacity to handle multiple simultaneous serial ports for GPS and GSM [18]. A Ublox Neo 6M GPS module with concurrent tracking capability was integrated, which improves precision in urban environments [19]. An A7670SA GSM module was employed for transmitting alerts via SMS. This architecture guarantees a higher delivery rate in areas with intermittent coverage and reduces monthly operating costs by 75–85% compared to mobile data solutions [20,21]. A voltage sensor for detecting vehicle ignition status, and a MicroSD card reader module for local data storage, were also included. The cost details and specifications of the components used are shown in Table A2.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 March 2026 doi:10.20944/preprints202603.1980.v1

Logical development and data analysis were supported by specialized software tools. Microcontroller firmware was programmed in the Arduino IDE version 2.3.7 integrated development environment, using C++ for managing positioning and communication libraries. For schematic design and pre-experimental circuit validation (see complete diagram in Figure A1 and connections in Table A1, the Fritzing simulator version 1.0.4 was used, allowing verification of connection integrity before physical implementation. Raw data processing, tabulation of field tests, and generation of descriptive charts were performed in Microsoft Excel 2021. Finally, for sample size calculation and inferential statistical analysis of percentage errors, Minitab® version 21.4 was used, configured for a 95% confidence level.

2.5. Data Collection Procedures

Data collection was structured in two main phases:

2.5.1. Pre-Intervention Phase

In this stage, reference values for the system were established. Maintenance intervals by mileage for the vehicle's main systems (engine oil change, transmission oil, brakes, spark plugs, and timing system) were collected from manufacturer manuals. These values were programmed into the microcontroller memory as alert thresholds.

2.5.2. Post-Intervention Phase

Once the prototype was installed in the vehicle (see physical assembly in Figures A2–A5), the system automatically began calculating the distance traveled from GPS coordinates. The vehicle odometer reading was manually recorded at the start and end of each weekly test period, enabling direct comparison with the GPS Ublox Neo 6M module measurement. Each alert event generated was automatically logged to the MicroSD card, including date, time, remaining mileage for each maintenance interval, and total mileage.

Laboratory tests included verification of GPS signal reception, reliability of SMS alert sending using the A7670SA GSM module, and measurement of system latency. Only after meeting the established technical criteria were field tests conducted under real operating conditions.

2.6. Maintenance Alert Algorithm

The implemented algorithm continuously compares the accumulated mileage calculated by the GPS module against the maintenance thresholds defined for each vehicle. When proximity to a maintenance interval is detected, the system generates three progressive SMS alerts: a first alert at 50 km before the scheduled maintenance, a second at 10 km, and a third upon reaching the exact established mileage. This strategy enables anticipatory planning of maintenance interventions.

2.7. Data Analysis Methods

Data analysis was developed using descriptive statistical techniques. GPS system accuracy was determined by calculating the Mean Absolute Percentage Error (MAPE) and the Root Mean Square Error (RMSE), using the vehicle odometer record as the reference standard. Additionally, the standard deviation of the error in each test scenario was analyzed to quantify system repeatability and to verify the consistency of the systematic bias (Positive Bias) inherent to GNSS technology. Detailed results of these indicators are presented in Table A3.

The performance of the alert system was evaluated using a confusion matrix, in which events were categorized as correct and incorrect detections, both positive and negative [22]. From this classification, the overall precision index of the system was calculated, considering values above 95% as valid.

Finally, system latency was analyzed through descriptive statistics, estimating the mean and standard deviation of the time elapsed between the activation of the maintenance threshold and the user's receipt of the SMS (see statistical summary in Table A4. The system met requirements when latency was below 10 seconds.

2.8. Statistical Determination of Sample Size

To ensure the representativeness of the experimental data, an initial pilot phase was conducted consisting of 3 repetitions for each test scenario. From these preliminary results, the standard deviation of the measurement error (S) was calculated for each route type.

To ensure the accuracy of the final results, the Minitab statistical software was used to determine the minimum sample size (n) required to estimate the mean error with a 95% confidence level ($Z = 1.96$). Maximum permissible error margins (E) specific to each distance scale were established, yielding the following experimental design parameters:

Urban Route (A): An initial standard deviation of $S = 0.065$ km was observed. For a target error margin of 0.1, the calculation determined a sample size of 5 repetitions.

Peripheral Route (B): With a pilot standard deviation of $S = 0.161$ km and an admissible error margin of 0.2, 5 repetitions were required.

Long-Distance Route (C): Due to greater variability in route conditions (error standard deviation $S = 0.709$ km), an error margin of 0.5 was set, yielding a requirement of 11 repetitions.

3. Results

The results of the validation are presented below.

3.1. Experimental Validation of the Proposed System

To verify system functionality, specific tests were carried out on each of its main subsystems that support the prototype's functionality: positioning (GPS), communication (LTE/GSM), storage (MicroSD), and energy/ignition state (voltage detection). Results are organized by subsystem, including measured parameters, observed behavior, and the corresponding technical interpretation.

3.1.1. Validation of the Global Positioning Module (GPS)

GPS module validation aimed to verify three essential aspects for prototype operation: (i) the ability to acquire a satellite signal in an adequate time (Fix), (ii) the accuracy of the calculated distance traveled via coordinate-based odometry, and (iii) the consistency of accumulated error when compared to the vehicle odometer, used as the operational field reference.

Signal Acquisition Time (Fix).

In the static cold-start test, the GPS module achieved valid satellite fix in an average time of 45 seconds. This result is considered adequate for an embedded system operating with consumer-grade GNSS signals, especially considering that in real scenarios, the initial Fix is often affected by environmental conditions (partial obstruction, satellite availability, and signal quality). The recorded time allows the system to be operational within a reasonable interval from vehicle start-up, enabling the initiation of distance counting and maintenance logic without prolonged delays.

Accuracy of Accumulated Distance Measurement.

The prototype's odometry is based on the accumulation of distances between successive points calculated from GNSS coordinates. To reduce the impact of "urban noise" and avoid accumulation of micro-displacements when the vehicle is stationary (e.g., at traffic lights), firmware logical filtering was applied. Distance calculation was optimized using the native function of the TinyGPS++ library (`distanceBetween`), replacing complex manual calculations. Additionally, firmware logic was applied to discard displacements at speeds below 3.0 km/h or distances less than 5 meters between readings. This approach reduces the risk of "phantom sums" in stop-and-go environments.

The following code as shown in Listing 1 excerpt illustrates how the algorithm calculates real distance and dynamically updates maintenance counters:

 doi:10.20944/preprints202603.1980.v1

Listing 1. GPS processing and odometer update logic executed in the main loop.

```
// GPS processing logic and odometer update (inside void loop)
while (Serial1.available() > 0) {
  if (gps.encode(Serial1.read())) {
    if (engineOn && isConfigured) {
      if (gps.location.isValid()) {
        if (firstFix) {
          lastLat = gps.location.lat();
          lastLon = gps.location.lng();
          firstFix = false;
        } else {
          double distMeters = gps.distanceBetween(
            gps.location.lat(), gps.location.lng(),
            lastLat, lastLon);

          // Urban noise filters (Stop-and-Go)
          if (gps.speed.kmph() > 3.0 && distMeters > 5.0) {
            totalOdometer += (unsigned long)distMeters;

            // Dynamic update of 6 maintenance counters
            for (int i = 0; i < NUM_MAINT; i++) {
              maintCounters[i] += (unsigned long)distMeters;
            }

            lastLat = gps.location.lat();
            lastLon = gps.location.lng();
          }
        }
      }
    }
  }
}
```

GPS vs. Vehicle Odometer Comparison.

The dynamic test consisted of a controlled long-distance trip, using the vehicle's digital odometer as a practical reference. After completing a standard route of 160.0 km (odometer), the prototype recorded 164.0 km as accumulated distance. In relative terms, this difference represents an error of approximately +2.5%, evidencing a positive deviation (overestimation). The relationship between both measurements is expressed by Equations (1) and (2):

$$\text{Error}_{\text{Abs}} = |\text{Value}_{\text{Ref}} - \text{Value}_{\text{Meas}}| = |160.0 - 164.0| = 4.0 \text{ km} \tag{1}$$

$$\text{Error}_{\text{Rel}} = \frac{\text{Error}_{\text{Abs}}}{\text{Value}_{\text{Ref}}} \times 100 = \frac{4.0}{160.0} \times 100 = 2.5\% \tag{2}$$

Error Margin Analysis and Systematic Tendency.

The 2.5% error is considered satisfactory for a GNSS system without differential correction (e.g., without RTK or advanced SBAS), particularly in urban conditions where signal can degrade due to obstructions, building reflections, and variations in horizontal precision. The observed positive tendency is interpreted as resulting from accumulated residual micro-variations in the signal during stops, which can introduce small phantom distances that add to the total, even with filtering applied.

In practical terms, this slight overestimation does not represent an operational risk. On the contrary, it introduces a conservative margin: by slightly overestimating distance, alerts will tend to be triggered ahead of the real mileage, reducing the risk of the user exceeding the critical oil change or component replacement interval. Consequently, the GPS module meets the prototype's primary requirement: delivering a consistent and sufficiently precise measurement to trigger mileage-based maintenance thresholds.

 doi:10.20944/preprints202603.1980.v1

3.1.2. Validation of the LTE/GSM Communication Module

The communication subsystem validation aimed to guarantee the prototype's ability to operate on the current telecommunications infrastructure, prioritizing fast connectivity, link stability, and low latency in alert delivery. Unlike traditional 2G solutions, the module used (A7670SA) allows LTE network operation, which improves performance in environments where 2G coverage is limited or obsolete.

SMS Alert Transmission Latency.

System latency was evaluated through 20 consecutive alert transmissions under standard coverage conditions. Statistical analysis reveals a mean of 4.13 seconds and a standard deviation of 0.63 seconds. As summarized in Table 1, the maximum recorded value was 5.00 seconds (see detailed log in Table A4. These results validate compliance with the design requirement (< 10 s), aligning with delivery times reported in the literature for low-cost SMS-based telemetry systems [23,24].

To quantify the operational effectiveness of the notification system, a confusion matrix was constructed based on 20 test events ($N = 20$). A "Positive Case" was defined as the condition in which the vehicle effectively reaches the programmed mileage threshold and the system must generate an SMS alert. Results were classified as:

- **True Positive (TP):** The system detected the threshold and correctly sent the SMS.
- **False Positive (FP):** The system sent an alert without the threshold having been reached (false alarm).
- **False Negative (FN):** The system reached the threshold but did not send the alert (detection failure).
- **True Negative (TN):** The maintenance time had not arrived, and the system correctly sent no alert.

Table 1. Confusion Matrix of the Alert System.

	Alert Sent (Predicted)	**No Alert (Predicted)**
Real Event (Threshold Reached)	TP = 20	FN = 0
Real Event (No Threshold)	FP = 0	TN = *

From these values, key performance indicators were calculated: Precision (Eq. 3) and Recall/Sensitivity (Eq. 4):

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{20}{20 + 0} = 1.0\ (100\%) \quad (3)$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{20}{20 + 0} = 1.0\ (100\%) \quad (4)$$

The 100% precision and recall results mathematically validate the reliability of the decision algorithm and the A7670SA communication module, surpassing the success criterion established for critical notification management ($> 95\%$).

Figure 2 shows the SMS alert message as received on the vehicle owner's mobile device, confirming the end-to-end communication chain from the prototype to the user.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 March 2026 doi:10.20944/preprints202603.1980.v1

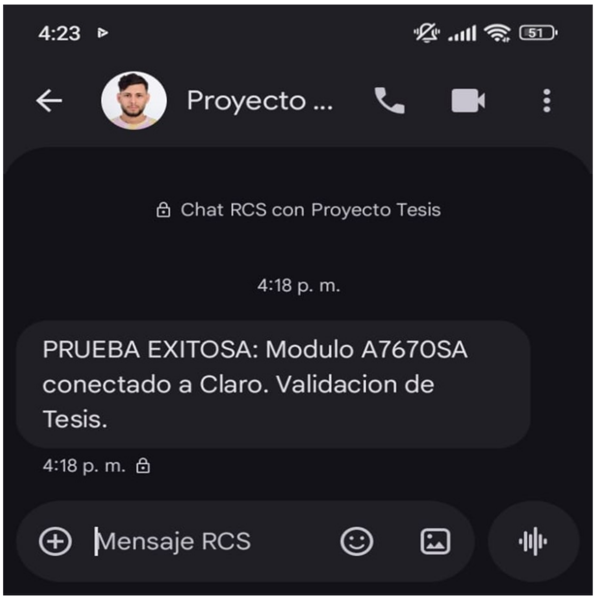

Figure 2. SMS maintenance alert received on the owner's mobile device.

3.1.3. Validation of the Data Storage System

The storage module (MicroSD) fulfills a critical function: ensuring information persistence upon power loss and enabling the historical logging of mileage, events, and alerts. Its validation focused on correct hardware initialization, write integrity, and the ability to recover state after abrupt power interruptions.

The storage system was redesigned to fulfill a dual critical function: hosting a static database (`DATA.CSV`) with the profiles and specific maintenance schedules for various vehicle models, and persisting the user's dynamic configuration in a local file (`CONFIG.TXT`). The latter stores the owner's contact number, selected vehicle, total odometer, and real-time counters for up to 6 independent maintenance intervals.

As shown in the code excerpts (see Listing 2), the system manages structured reading and writing of these parameters. This routine is fundamental for recovering exact information after power cuts or restarts, preventing loss of service history.

Listing 2. Functions for saving and loading user configuration and maintenance data.

```
// User configuration and maintenance persistence functions
void saveConfig() {
  SD.remove("CONFIG.TXT");
  File file = SD.open("CONFIG.TXT", FILE_WRITE);
  if (file) {
    file.println(ownerNumber);
    file.println(currentYear);
    file.println(currentModel);
    file.println(totalOdometer);
    for (int i = 0; i < NUM_MAINT; i++) {
      file.print(maintNames[i]); file.print(",");
      file.print(maintIntervals[i]); file.print(",");
      file.println(maintCounters[i]);
    }
    file.close();
  }
}

bool loadConfig() {
  if (!SD.exists("CONFIG.TXT")) return false;
  File file = SD.open("CONFIG.TXT");
  if (file) {
```

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 March 2026 doi:10.20944/preprints202603.1980.v1


```
    ownerNumber = file.readStringUntil('\n'); ownerNumber.trim();
    currentYear = file.readStringUntil('\n'); currentYear.trim();
    currentModel = file.readStringUntil('\n'); currentModel.trim();
    String odoStr = file.readStringUntil('\n'); odoStr.trim();
    totalOdometer = odoStr.toInt();
    for (int i = 0; i < NUM_MAINT; i++) {
      String line = file.readStringUntil('\n');
      int firstComma = line.indexOf(',');
      int secondComma = line.lastIndexOf(',');
      if (firstComma > 0 && secondComma > firstComma) {
        maintNames[i] = line.substring(0, firstComma);
        maintIntervals[i] =
          line.substring(firstComma+1, secondComma).toInt();
        maintCounters[i] =
          line.substring(secondComma+1).toInt();
        alert50[i]=false; alert10[i]=false; alert0[i]=false;
      }
    }
    file.close();
    return true;
  }
  return false;
}
```

System State Recovery.

After restoring power, the system executed the recovery routine from a state file, successfully restoring the last stored mileage. This behavior confirms that the prototype maintains operational continuity and avoids "zero restarts" that would affect monitoring validity. The storage subsystem thus demonstrates robustness for vehicular scenarios, where power-off and power-on events are a normal part of daily operation.

3.1.4. Validation of the Energy and Virtual Ignition System

The power system was validated with two objectives: (i) to ensure that the electronics operate under stable conditions against typical automotive electrical system variations, and (ii) to verify the effectiveness of the voltage-based "virtual ignition" algorithm, avoiding the need for physical connection to the ACC wire and enabling direct connection to the battery terminals (Figure A4).

Power Supply Stability.

During tests, the voltage regulator maintained a stable 5 V output for inputs up to 14.5 V, a range consistent with automotive operation when the alternator is under load. This result indicates that the regulation stage is adequate for protecting the microcontroller and associated modules against fluctuations in the vehicle's electrical system.

Engine State Detection by Voltage.

The virtual ignition algorithm correctly discriminated engine states by reading battery voltage using a conversion factor adjusted to 5.7. Clearly differentiated ranges were established to avoid erratic states:

- Engine off: (< 12.9 V) $\rightarrow$ system in standby mode (no distance accumulation).
- Engine on: (> 13.8 V) $\rightarrow$ system in active mode (km counting and threshold evaluation enabled).

This logic is implemented as a virtual ignition strategy, as shown in Listing 3.

Listing 3. Engine-state detection based on battery voltage thresholds (virtual ignition).

```
// Engine state detection function (Virtual ignition)
void readVoltage() {
  // Conversion factor 5.7 for the implemented voltage divider
  int adc = analogRead(PIN_VOLT);
  batteryVoltage = (adc * 5.0 / 1023.0) * 5.7;

  if (batteryVoltage > 13.8) {
    engineOn = true;
  } else if (batteryVoltage < 12.9) {
    engineOn = false;
  }
}
```

3.2. Field Test Results on Real Routes

To evaluate system performance under real operating conditions, field tests were conducted based on the repetition of representative trajectories (illustrated in Figure A3) for the most frequent driving scenarios: dense urban environment, mixed peripheral road, and long-distance routes. In each case, the distance traveled calculated by the prototype was directly compared against the vehicle's odometer reading, considered the operational reference. The obtained results are presented below.

3.2.1. Results on Dense Urban Route

Urban route tests were performed in an environment characterized by heavy traffic, multiple traffic lights, and frequent stops—conditions representing the most demanding scenario for satellite positioning systems due to the urban canyon effect and signal drift in stop-and-go situations.

Route A was defined within the urban center of Guayaquil, characterized by high vehicle flow and frequent traffic lights. The route begins in the south of Guayaquil, departing from the Cangrejo Fútbol sportsbar, and advances northward along Avenida Quito, crossing a large part of the city to the center, where an intermediate stop is made at Picantería "El Sargento Mayor." For the return, the route descends southward again, first taking Lorenzo de Garaycoa street, then turning onto Vicente Trujillo street, and finally onto Av. Domingo Comin to conclude near the Guayas riverbank, specifically at the Caraguay Market Metrovía stop, as illustrated in Figure 3.

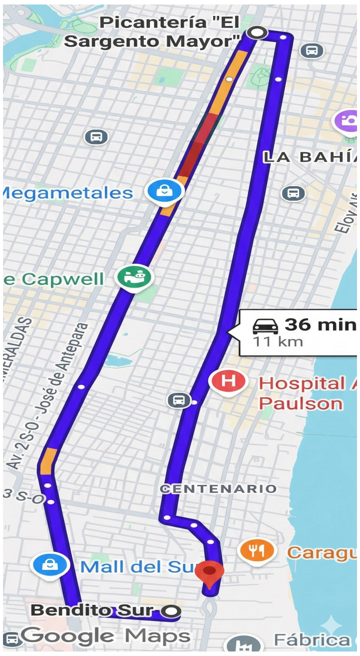

Figure 3. Route A: Dense urban route in Guayaquil city center.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 March 2026 doi:10.20944/preprints202603.1980.v1

Table 2. Results: Route A (Dense Urban — 11.0 km).

Repetition	Real Distance (Odometer)	Prototype Distance	Deviation (km)	Error (%)
Test A-1	11.00 km	11.67 km	+0.67 km	6.09%
Test A-2	11.00 km	11.76 km	+0.76 km	6.91%
Test A-3	11.00 km	11.69 km	+0.69 km	6.27%
Test A-4	11.00 km	11.73 km	+0.73 km	6.64%
Test A-5	11.00 km	11.70 km	+0.70 km	6.36%
AVERAGE	**11.00 km**	**11.71 km**	**+0.71 km**	**6.45%**

The average percentage error was 6.45% (overestimation). This deviation is the highest of the three scenarios and is attributed to the "multipath" effect or signal reflection off tall buildings in the city center. This phenomenon introduces "noise" into GPS coordinates, adding phantom extra meters while the vehicle moves slowly or is stopped at traffic lights, generating a count higher than the real distance.

3.2.2. Results on Mixed Peripheral Route

The mixed peripheral route corresponds to an intermediate scenario, characterized by moderate speeds (40–60 km/h), fewer stops, and continuous-flow sections. This environment enables evaluation of system behavior when the GPS operates with greater stability and less urban interference.

Route B begins in the southern part of the city, at Persianas Ecuador, and heads northward through Guayaquil primarily via the Vía Perimetral. The route continues to the reference point at Maximetales S.A. Norte, located in the northern sector of the city. From this point, the route returns following the same road axis back to the origin, completing a total round trip of 41 km south–north–south, as shown in Figure 4.

Figure 4. Route B: Mixed peripheral route via Vía Perimetral.

 doi:10.20944/preprints202603.1980.v1

Table 3. Results: Route B (Mixed Peripheral — 41.0 km).

Repetition	Real Distance (Odometer)	Prototype Distance	Deviation (km)	Error (%)
Test B-1	41.00 km	41.87 km	+0.87 km	2.12%
Test B-2	41.00 km	41.82 km	+0.82 km	2.00%
Test B-3	41.00 km	41.86 km	+0.86 km	2.10%
Test B-4	41.00 km	41.88 km	+0.88 km	2.15%
Test B-5	41.00 km	41.84 km	+0.84 km	2.05%
AVERAGE	**41.00 km**	**41.85 km**	**+0.85 km**	**2.08%**

Results show an average error of 2.08%. The significant reduction in error compared to the urban route (6.45%) confirms that, as building density decreases and average speed increases, the system improves its precision, while maintaining a slight positive tendency.

3.2.3. Results on Interurban Routes

Tests in the interurban scenario were carried out on a controlled 32 km route, designed to evaluate behavior at constant cruising speed. The obtained results (Table 4) show an average deviation of +1.09 km, representing a mean error of 3.40%.

Unlike the peripheral route (where the error was lower, 2.08%), in this scenario the sustained constant speed appears to have influenced the systematic accumulation of positive micro-positioning errors. This confirms a consistent tendency of the module to slightly overestimate distances on continuous routes without stops, aligning with the expected positive bias.

Route C begins on the Vía a la Costa highway, at the Delportal supermarket, continues along this main artery to the Chongón entrance (the return point), and finishes at the same starting location, forming a round trip of 32 km, as shown in Figure 5.

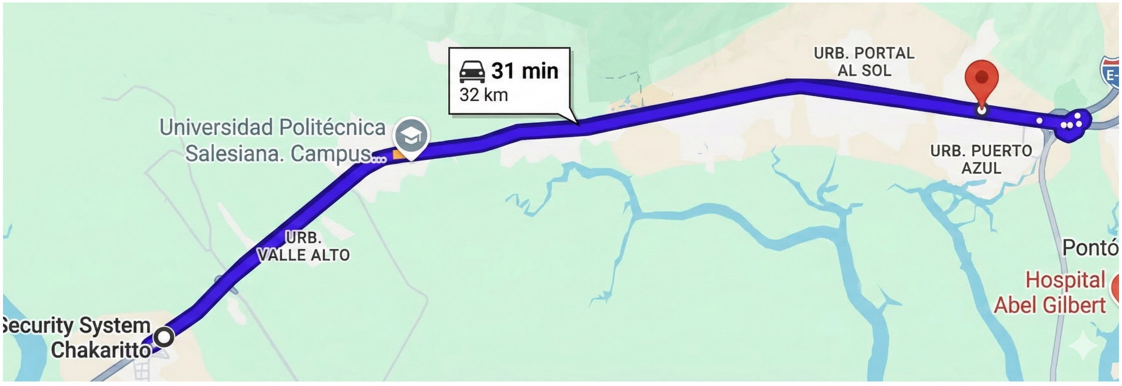

Figure 5. Route C: Interurban route on Vía a la Costa.

Table 4. Results: Route C (Interurban Road).

Repetition	Real Distance (Odometer)	Prototype Distance	Deviation (km)	Error (%)
Test C-1	32.00 km	33.06 km	+1.06 km	3.31%
Test C-2	32.00 km	33.10 km	+1.10 km	3.44%
Test C-3	32.00 km	33.09 km	+1.09 km	3.41%
Test C-4	32.00 km	33.08 km	+1.08 km	3.37%
Test C-5	32.00 km	33.09 km	+1.09 km	3.42%
Test C-6	32.00 km	33.11 km	+1.11 km	3.47%
Test C-7	32.00 km	33.08 km	+1.08 km	3.36%
Test C-8	32.00 km	33.08 km	+1.08 km	3.36%
Test C-9	32.00 km	33.11 km	+1.11 km	3.47%
Test C-10	32.00 km	33.10 km	+1.10 km	3.42%
Test C-11	32.00 km	33.07 km	+1.07 km	3.35%
AVERAGE	**32.00 km**	**33.09 km**	**+1.09 km**	**3.40%**

The low dispersion observed confirms that the error does not grow in an uncontrolled manner with increasing distance, but maintains stable proportionality relative to the total traveled. This behavior is indicative of a systematic and predictable drift, characteristic of commercial-grade GNSS systems without differential correction.

From an operational standpoint, this consistency is especially relevant as it allows anticipation of system behavior on extensive routes and validates its use for mileage-based preventive maintenance management.

3.3. Statistical Analysis of Results

3.3.1. Analysis of Accumulated Percentage Error

The consolidation of results reveals a clear pattern: the system overestimates distance in all scenarios. The mean error was 6.45% in urban, 2.08% in peripheral, and 3.40% in interurban routes. The global mean error of the system stands at **3.98%**. This consistency in overestimation (Positive Bias) is fundamental for reliability, as it indicates that the error is not random but systematic and predictable.

3.3.2. Comparison Between Satellite and Mechanical Odometry

The discrepancy observed between the odometry calculated by the prototype and the vehicle odometer reading responds to inherent differences between both measurement methods. While the satellite system calculates distance by summing geodetic segments between discrete positions, the mechanical odometer is based on tire rotation, being sensitive to physical factors such as inflation pressure and wear.

In this context, a systematic positive error close to 2.5% falls within acceptable margins for consumer-grade GNSS instrumentation. From a maintenance engineering perspective, this behavior is even favorable, as it introduces a quantifiable operational safety margin: projecting the mean error over a typical 5,000 km maintenance cycle, the system will generate the alert at approximately 4,875 real km, anticipating the intervention and reducing to zero the risk of exceeding the critical component operation limit.

3.3.3. Statistical Validation of Mean Differences

To determine whether the observed difference between the prototype measurement and the vehicle odometer is statistically significant, an inferential analysis was conducted using Minitab software, applied independently for the three test scenarios (Urban, Peripheral, and Interurban routes).

First, the normality assumption was verified using the Shapiro-Wilk test. In all three evaluated scenarios, p-values > 0.100 were obtained, indicating that the data follow a normal distribution. Confirming that the data distribution is parametric, the Student's t-test was applied. The corresponding normal probability charts are presented in (Figures A6–A8).

A paired t-test was applied to compare the means of the distance calculated by the prototype against the odometer reference distance. The analysis was performed under the following hypotheses, with a 95% confidence level ($\alpha = 0.05$):

- **Null Hypothesis (H_0):** There is no significant difference between the prototype and odometer measurements.
- **Alternative Hypothesis (H_1):** There is a significant difference between both measurements.

Analysis results yielded $p = 0.000$ for all three evaluated routes (see Figures A9–A11)). Since the p-value is less than the significance level ($p < 0.05$), the null hypothesis is rejected in all cases.

Rejection of the null hypothesis ($p < 0.05$) confirms that statistically significant differences exist between the measurements. However, this result should not be interpreted as random error, but as the confirmation of systematic behavior: the prototype consistently tends to measure a slightly greater distance than the odometer. This finding statistically validates the trend observed in the samples from all three scenarios (Tables 2–4), confirming that the 3.98% average error is attributable to the Positive

Bias inherent to GNSS technology and not to chance, which guarantees an operational safety margin for maintenance.

3.4. Discussion of Results and Experimental Conclusions

The comprehensive evaluation of the proposed system, based on unit and field tests conducted, allows analysis not only of the quantitative performance of each subsystem, but also of the overall reliability of the prototype as a support tool for preventive vehicle maintenance management. The results obtained evidence coherent, stable, and technically consistent behavior, adequate for application in light vehicles under real operating conditions.

3.4.1. Consistency and System Reliability

One of the most relevant findings of the study is the measurement system's consistency, observed across different route typologies and operating conditions. The filtering algorithm implemented in the GPS module, designed to mitigate urban noise and avoid accumulation of micro-displacements when the vehicle is stationary, demonstrated stable performance in demanding scenarios such as dense urban traffic. Despite the presence of frequent stops and signal obstructions, the system maintained a bounded and predictable error, confirming the stability of the filtering algorithm against external perturbations.

Repetition of trajectories under similar conditions allowed verification of measurement repeatability, evidenced by the low dispersion of percentage error between consecutive tests. In particular, in long-distance routes, error variation remained within a narrow range, indicating that the observed deviation is not the product of random fluctuations but of systematic behavior inherent to the satellite measurement method. This characteristic is essential for preventive maintenance applications, where system predictability is a key factor for reliable decision-making.

Taken together, these results validate that the system is not only functional but also reliable over time, preserving its performance under different traffic conditions, speeds, and environments, meeting the technical criteria defined in the methodological phase.

3.4.2. System Operational Safety Margin

Analysis of the impact of the measurement error on maintenance cycles allows evaluation of the system from an operational and reliability engineering perspective. The observed systematic positive error, with a mean value close to 2.5%, implies that the system tends to slightly overestimate the distance traveled. Projecting this behavior over a typical 5,000 km maintenance cycle, the alert would be generated at approximately 4,875 real km, introducing an anticipation margin of approximately 125 km.

This behavior adheres to the "fail-safe" design principle, widely applied in critical systems, where safety is prioritized over absolute precision. In the context of vehicular maintenance, it is preferable for the system to anticipate the need for intervention before the component reaches its critical operating limit, rather than delaying the alert and increasing the risk of excessive wear or mechanical failure.

From this perspective, the observed operational safety margin does not represent a limitation of the system, but a functional advantage, as it protects engine and component integrity regardless of driving style or road conditions. Experimental results consequently confirm that the proposed system fulfills its fundamental objective: optimizing preventive maintenance management through timely, reliable, and safety-oriented alerts.

4. Discussion

4.1. Analysis of Technical Precision Against Industry Standards

Experimental results yielded a global systematic mean error of 3.98% (overestimation) in distance measurement. Contrasting this value with the specialized literature, consistent behavior is evident across all three evaluated scenarios, where deviation was invariably positive, validating the "Positive Bias" theory [25]. This phenomenon, inherent to the discrete sampling of trajectories via GPS in

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 March 2026 doi:10.20944/preprints202603.1980.v1

environments with signal noise, mathematically explains the tendency observed in our tests. In this context, recent literature agrees that experimental evaluation of systems applied to the vehicular environment should be conducted under real operating conditions, as these more accurately reflect the technical challenges associated with urban traffic and vehicle-infrastructure interaction [26].

Despite variations by environment, the overall prototype performance remains within accepted safety ranges. As a regulatory reference, the National Institute of Standards and Technology (NIST) establishes a 4% acceptance tolerance for vehicle odometers [16]. Given that our system presented a global mean deviation of 3.98%, the device strictly complies with the precision standards allowed for vehicular mechanical instrumentation.

Finally, in terms of road safety, the automation of alerts reduces dependence on driver memory, mitigating the risk of accidents caused by mechanical failures, which are consistently reported by ANT [27]. Similarly, compliance with maintenance intervals favors control of pollutant emissions (CO, CO_2, NO_x), aligning with the objectives of current environmental regulations [5,6].

4.2. Economic Impact and Viability in Maintenance Management

The viability of an aftermarket telemetry system depends directly on its economic accessibility compared to existing market alternatives. To evaluate this aspect, a comparative analysis was performed between the proposed prototype and two widely distributed commercial solutions (Vyncs and Fleetio), detailed in Table 5.

Table 5. Comparison of Similar Solutions.

Solution	Technology	Initial Cost	Monthly Cost	Communication	Limitations
Vyncs	OBD-II tracker with DTC reader	$146	$22.00	Proprietary app (internet)	Depends on data coverage; manual scheduling
Fleetio	Fleet management cloud software	N/A	$10.00	Proprietary app (internet)	Limited automated integration; requires daily manual km update
Proposed Prototype	Satellite telemetry-based preventive alert system	$78.84	$3.00 (SMS)	SMS messaging	Does not extract DTC fault codes from ECU

4.2.1. Cost Structure and Scalability

The initial cost of $78.84 USD for the functional prototype represents the version built with standard commercial components (Arduino Mega, independent GPS/GSM modules) acquired at retail prices. A formal scalability analysis reveals significant optimization opportunities if the system advances to a manufacturing phase:

- **Series production (batches of 100 units):** Bulk component acquisition enables an estimated 25–30% cost reduction, placing the base cost between $55.00 and $59.00 USD. Migration toward a custom Printed Circuit Board (PCB) design would add approximately $15.00 USD per unit at this volume, resulting in a final production cost of ∼$70.00–$74.00 USD per device.
- **Industrial production (batches of 1,000+ units):** Scaling production and integrating Surface Mount Device (SMD) components directly into a System-on-Chip (SoC) design reduces manufacturing costs by 40–45%. The projected cost would drop to ∼$43.00–$47.00 USD per unit, also achieving greater physical robustness and a significantly smaller installation size.

4.2.2. Return on Investment Analysis (ROI)

To evaluate the real impact on the user, an implementation scenario was considered with a projected final price of $120.00 USD and a monthly operating cost of $3.00 USD (basic SMS plan). The Total Cost of Ownership (TCO) during the first year would be $156.00 USD. In contrast, the commercial Vyncs solution represents an annual TCO of $410.00 USD ($146.00 + $264.00), demonstrating that the proposed prototype generates a **62% savings** compared to international alternatives.

 doi:10.20944/preprints202603.1980.v1

Furthermore, studies show that lack of management and corrective maintenance exceeds the costs of preventive maintenance by 60% [4]. For a light vehicle with average annual preventive expenditure of $800.00 USD, omitting service intervals would raise the cost to ∼$1,280.00 USD annually due to mechanical failures. Implementation of this system ensures preventive compliance, generating a net annual savings of ∼$480.00 USD. Under this metric, the payback period for the end user is reached in just 3 to 4 months of operation.

Considering that 75% of the Ecuadorian vehicle fleet exceeds 6 years of age and lacks native telematics systems [2], this solution demonstrates not only economic viability, but structural relevance for mitigating the high operating costs of the sector.

4.3. Impact on Maintenance Management

Beyond economic viability, the experimental results allow direct quantification of the extent to which the system optimizes preventive maintenance management. First, the pre/post intervention design declared in Section 2.1 establishes the reference comparison: before system implementation, the vehicle lacked any automated alert mechanism and depended exclusively on manual methods subject to the "forgetfulness factor" [3]; after implementation, the system notifies with 100% effectiveness ($N = 20$, $FN = 0$), technically eliminating this factor as a cause of non-compliance with maintenance intervals. Second, the systematic 2.5% positive bias introduces a quantifiable anticipation margin: projected over a typical 5,000 km cycle, the system generates the alert at approximately 4,875 real km, anticipating the critical threshold by 125 km and reducing to zero the risk of interval overshoot if the driver responds to the notification. This behavior is consistent with what has been documented in automatic alert systems for machinery, where alert automation has been shown to improve adherence to maintenance schedules and reduce unplanned downtime [11]. Together, these two indicators—100% notification effectiveness and deterministic 125 km anticipation per cycle—constitute quantitative evidence of compliance with the preventive maintenance optimization objective set out in this study.

4.4. Technical Comparison Against On-Board Diagnostic Systems (OBD-II)

While recent literature proposes OBD-II-based monitoring solutions and cloud platforms for fleet management [7], this project opted for an independent hardware architecture. To objectively evaluate the differences between both solutions, a technical comparative analysis was developed (Table 6), contrasting the capabilities of the OBD-II standard against the proposed satellite system.

Table 6. OBD-II Comparative Analysis.

Technical Parameter	Commercial OBD-II Standard	Proposed Prototype (GPS/GSM)
Odometry source	Direct (ECU and wheel sensor readings)	Indirect (satellite odometry via Haversine formula)
Engine diagnostics	Yes (DTC fault code extraction)	No (ignition state detection by voltage only)
Intrusion level	High (physical connection to CAN/K-Line data bus)	None (total isolation from data bus)
Universality	Limited (depends on manufacturer protocol and year)	High (works in any vehicle with 12 V electrical system)
Energy consumption	Risk of parasitic drain when scanner left connected	Controlled (LM2596S regulator and automatic standby)

In the Ecuadorian context, where 75% of the vehicle fleet exceeds 6 years of age and 60% has more than 11 years of use [2], relying on the OBD-II port drastically limits compatibility. Vehicles of these generations often present worn diagnostic ports, closed manufacturer communication protocols, or incompatibilities with generic scanners (ELM327). Furthermore, keeping a device permanently connected to the OBD-II port can generate parasitic consumption that drains the battery.

 doi:10.20944/preprints202603.1980.v1

By not depending on the on-board computer (ECU) or the OBD-II standard, the proposed system offers installation universality that commercial solutions dependent on the CAN network do not possess [9]. The decision to use an isolated system guarantees that there will be no electronic interference with the vehicle's critical systems, prioritizing accessibility and universal compatibility to mitigate the "forgetfulness factor" in maintenance management.

4.5. Study Limitations

While overall accuracy was satisfactory, greater error dispersion was observed on dense urban routes (2.88%) compared to peripheral routes (1.48%). This is consistent with findings reported by Hussain et al. [7] regarding the challenges of the "urban canyon" effect in IoT systems. Future iterations could benefit from more aggressive Kalman filtering algorithms to smooth readings in stop-and-go traffic situations.

Finally, it is pertinent to note that direct quantification of the behavioral impact of the system—such as the percentage reduction in mileage overshoot measured over a sample of real drivers over several months—would require a longitudinal study with a control group beyond the scope of this work. The quasi-experimental design adopted is appropriate for technical prototype validation, which constitutes the necessary precondition prior to any large-scale deployment. Evaluation of the impact on user maintenance behavior constitutes a natural line of future research.

5. Conclusions

An integrated preventive maintenance management and alert system based on an open architecture was successfully developed and implemented, incorporating positioning and mobile communication modules, which enabled improvement of vehicular maintenance management through quantifiable indicators. Experimental validation evidenced distance measurement accuracy with a mean error of 3.98%, sufficient to guarantee the reliability of established alert thresholds, as well as 100% effectiveness in SMS notification sending during tests conducted, technically eliminating the omission of maintenance intervals due to user forgetfulness. Additionally, an anticipation margin of approximately 2.5% relative to the 5,000 km reference maintenance interval was obtained, which reduces the risk of exceeding critical service limits and favors timely planning of interventions. The average latency of 4.13 seconds confirmed the system's real-time operation capability. Together, these results constitute objective evidence of more timely, controlled, and predictable preventive maintenance management, fulfilling the optimization objective set out in this research.

The study concludes that the proposed technology is a viable solution to bridge the technological gap in the Ecuadorian vehicle fleet, whose average age exceeds 16 years. By not depending on the on-board computer (ECU) or the OBD-II standard, the system offers installation universality that commercial solutions do not possess. This enables the democratization of access to vehicular safety, transforming old analog vehicles into connected assets capable of preventing catastrophic failures.

Validation of SMS messaging as the exclusive alert channel confirmed its economic sustainability. With reduced operating costs (75–85% savings), the system is accessible to the average user and small transportation fleets, aligning with the need to reduce the high costs associated with unplanned corrective maintenance.

Future lines of work identified include the integration of OBD-II reading modules for compatible vehicles, allowing crossing of mileage data with engine diagnostic variables (temperature, RPM). Additionally, development of a complementary mobile application capable of reading the history stored on the MicroSD card via Bluetooth is suggested, offering the user a graphical visualization of their driving habits and vehicle health over time.

Appendix A. Complete Circuit Diagram

The circuit design was performed using Fritzing software, enabling pre-simulation of the system before physical implementation. Figure A1 presents the complete system schematic, showing connec-

doi:10.20944/preprints202603.1980.v1

tions between the Arduino Mega 2560 microcontroller and peripheral modules (GPS, GSM, MicroSD, and voltage sensor).

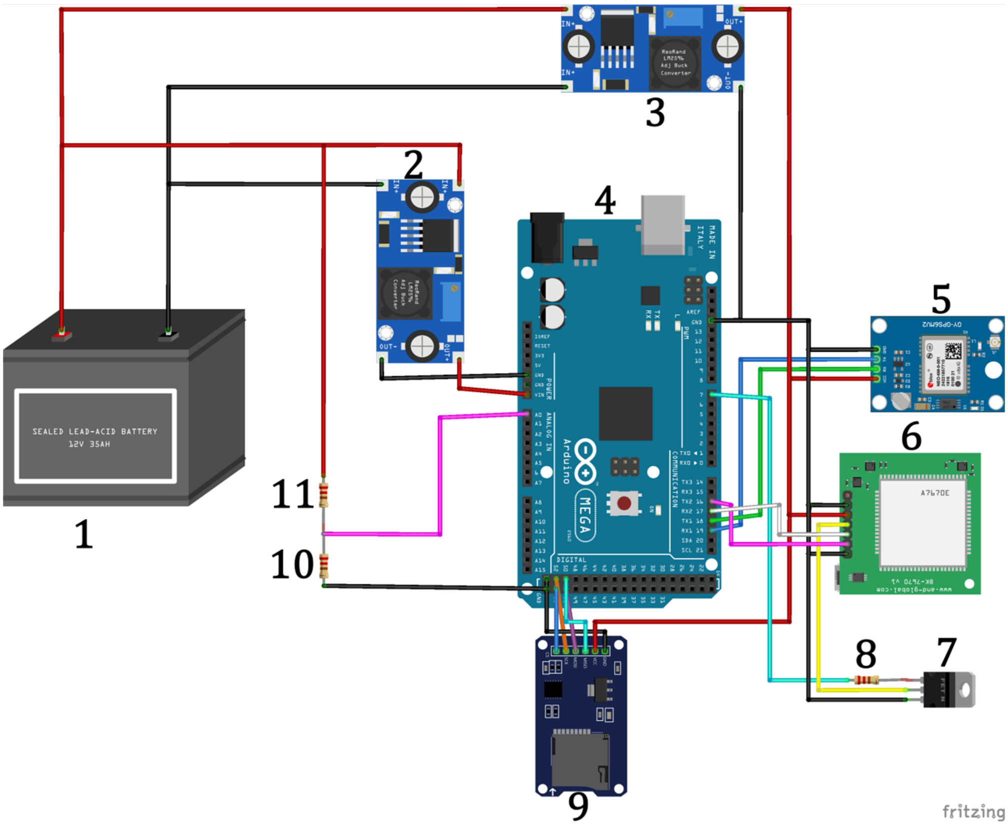

Figure A1. Complete schematic diagram of the preventive maintenance management system simulated in Fritzing. Parts: 1 (12V vehicle battery), 2 and 3 (voltage converter), 4 (Arduino Mega), 5 (GPS module), 6 (GSM module), 7 (MOSFET), 8 (resistor), 9 (MicroSD module), 10 and 11 (resistors).

Table A1. System Connections.

Component	Component Pin	Destination Device	Destination Pin
GPS Module	GND	Arduino Mega	GND
GPS Module	VCC	Voltage conv. #3	OUT+
GPS Module	TX	Arduino Mega	RX(19)
GPS Module	RX	Arduino Mega	TX(18)
GSM Module	GND	Arduino Mega	GND
GSM Module	VCC	Voltage conv. #3	OUT+
GSM Module	RX	Arduino Mega	TX2(16)
GSM Module	TX	Arduino Mega	RX2(17)
GSM Module	PWRKEY	MOSFET	Drain
MicroSD Module	CS	Arduino Mega	53
MicroSD Module	MOSI	Arduino Mega	51
MicroSD Module	MISO	Arduino Mega	50
MicroSD Module	SCK	Arduino Mega	52
MicroSD Module	VCC	Voltage conv.	OUT+
MicroSD Module	GND	Arduino Mega	GND
MOSFET	Gate	Arduino Mega	7
MOSFET	Source	Arduino Mega	GND

Appendix B. Component List and Technical Specifications

Component selection was made considering criteria of commercial availability, voltage compatibility, energy consumption, and operational reliability. Table A2 details all electronic components used in the prototype, including model, relevant technical specifications, and quantity used.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 March 2026 doi:10.20944/preprints202603.1980.v1

Table A2. Electronic Components of the System.

N°	Component	Qty.	Unit Cost (USD)	Subtotal (USD)
1	Arduino Mega 2560	1	$25.00	$25.00
2	Jumper cables	40	$0.05	$2.00
3	GPS antenna	1	$10.25	$10.25
4	Protoboard	1	$1.91	$1.91
5	Ring terminals	2	$0.15	$0.30
6	DC connectors	2	$0.45	$0.90
7	AWG cable	1	$4.00	$4.00
8	Resistors	3	$0.05	$0.15
9	LM2596S voltage converter module	2	$2.50	$5.00
10	GSM Module A7670SA	1	$10.17	$10.17
11	GPS Module Ublox Neo 6M	1	$5.95	$5.95
12	MicroSD module	1	$1.81	$1.81
13	MicroSD 64 GB	1	$6.90	$6.90
14	Enclosure	1	$3.00	$3.00
15	MOSFET irf540N	1	$1.50	$1.50
			TOTAL	**$78.84**

Appendix C. Physical Prototype Assembly

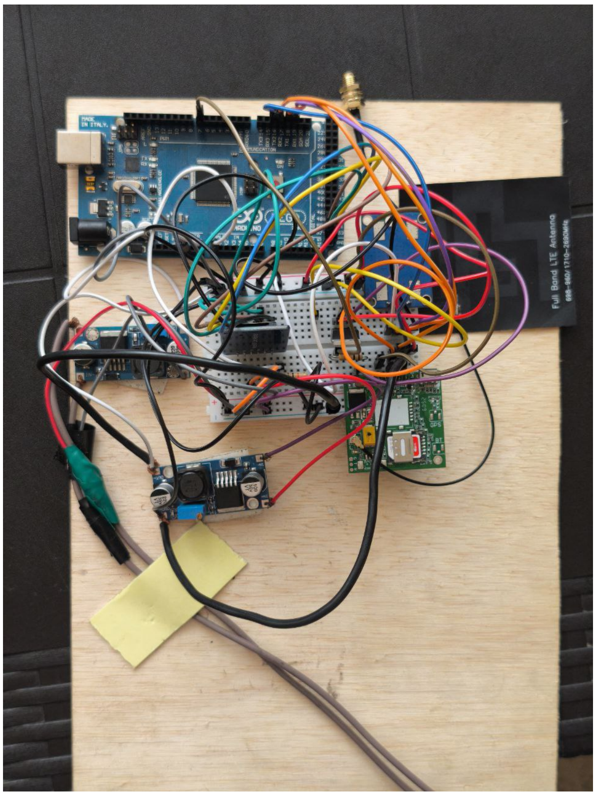

Figure A2. Physical assembly of the implemented prototype.

doi:10.20944/preprints202603.1980.v1

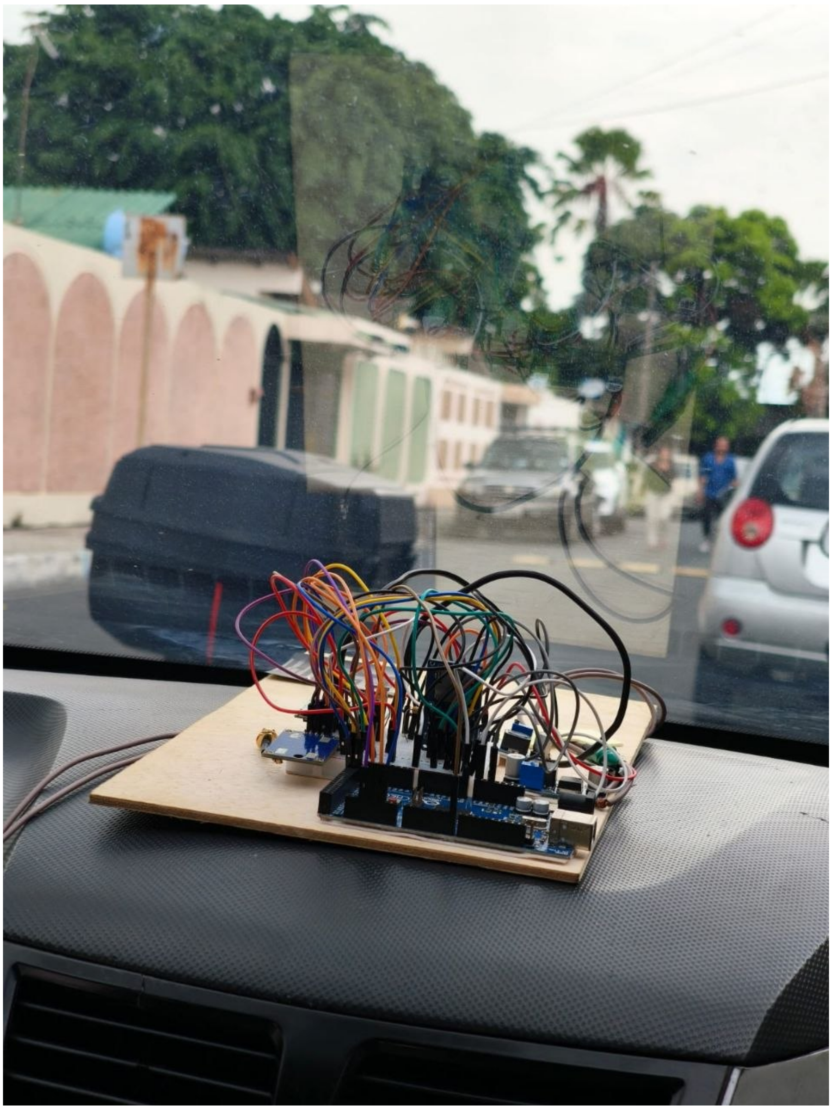

Figure A3. Field route test of the prototype.

Figure A4. Battery connection.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 March 2026 doi:10.20944/preprints202603.1980.v1

Figure A5. Connector output to the vehicle cabin.

Appendix D. General Code Structure

This appendix presents the complete and final source code of the firmware implemented in the Arduino Mega 2560 microcontroller. The system has been programmed with a dynamic architecture that includes a state machine for interactive configuration via SMS messages. This allows the user to select their vehicle from a pre-loaded database (DATA.CSV) and automatically manage multiple maintenance intervals without requiring physical hardware reprogramming.

```
// 1. GLOBAL VARIABLES AND DYNAMIC STRUCTURES
float batteryVoltage = 0.0;
bool engineOn = false, isConfigured = false;
unsigned long totalOdometer = 0;
const int NUM_MAINT = 6;
String maintNames[NUM_MAINT];
unsigned long maintIntervals[NUM_MAINT], maintCounters[NUM_MAINT];
// State machine for interactive configuration
enum SetupState {
  ST_IDLE, ST_WAIT_MODEL_INDEX, ST_WAIT_ODO, ST_WAIT_HISTORY
};
SetupState setupState = ST_IDLE;

// 2. VIRTUAL IGNITION AND SATELLITE ODOMETRY
void readVoltage() {
  batteryVoltage = (analogRead(PIN_VOLT) * 5.0 / 1023.0) * 5.7;
  engineOn = (batteryVoltage > 13.8);
}
/* Main loop: urban noise filter and accumulation
if (gps.speed.kmph() > 3.0 && distMeters > 5.0) {
  totalOdometer += distMeters;
  for (int i = 0; i < NUM_MAINT; i++)
    maintCounters[i] += distMeters;
} */

// 3. STATE MACHINE FOR SMS CONFIGURATION
void processIncomingSMS(String sender, String text) {
  if (!isConfigured) {
    switch(setupState) {
      case ST_IDLE:
        sendModelListSMS();
```

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 March 2026 doi:10.20944/preprints202603.1980.v1


```
        setupState = ST_WAIT_MODEL_INDEX; break;
      case ST_WAIT_MODEL_INDEX:
        /* Extracted model selection logic... */
        setupState = ST_WAIT_ODO; break;
      case ST_WAIT_ODO:
        totalOdometer = text.toInt() * 1000UL;
        setupState = ST_WAIT_HISTORY; break;
      case ST_WAIT_HISTORY:
        maintCounters[setupMaintIndex++] = text.toInt() * 1000UL;
        if (setupMaintIndex >= NUM_MAINT) {
          isConfigured = true;
          saveConfig();
        }
        break;
    }
  }
}

// 4. ALERT ENGINE AND PERSISTENCE (MicroSD)
/* Periodic check routine (executed in loop)
for (int i = 0; i < NUM_MAINT; i++) {
  long remKm = (maintIntervals[i]-maintCounters[i])/1000;
  if (remKm<=50 && remKm>10)
    sendSMS("ALERT: "+maintNames[i]+" in 50 km.");
  else if (remKm<=10 && remKm>0)
    sendSMS("URGENT: "+maintNames[i]+" in 10 km.");
  else if (remKm<=0)
    sendSMS("OVERDUE: "+maintNames[i]+".");
} */
void saveConfig() {
  File file = SD.open("CONFIG.TXT", FILE_WRITE);
  // Persists: phone, vehicle, odometer, maintenance counters
  file.close();
}
```

Appendix E. Experimental Data Obtained

Table A3. Summary of Statistical Performance Indicators and Error by Test Scenario.

Test Scenario	N	Mean Error (Bias) [km]	Std. Dev. (σ) [km]	RMSE [km]	MAPE [%]
Route A (Urban)	5	+0.710	0.035	0.711	6.45%
Route B (Peripheral)	5	+0.854	0.024	0.854	2.08%
Route C (Interurban)	11	+1.088	0.016	1.088	3.40%
Global Average	**21**	**+0.942**	**0.166**	**0.956**	**3.98%**

Table A4. Detailed Log of Alert Transmission Latency Times ($N = 20$).

Test #	Latency (s)	Test #	Latency (s)
1	4.67	11	3.94
2	4.19	12	3.94
3	4.78	13	4.47
4	5.00	14	3.00
5	4.12	15	3.00
6	4.12	16	3.87
7	5.00	17	3.53
8	4.87	18	4.53
9	3.94	19	3.61
10	4.70	20	3.23

Mean = 4.13 s; Std. Dev. = 0.63 s; Max = 5.00 s

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 March 2026 doi:10.20944/preprints202603.1980.v1

Shapiro-Wilk Normality Tests and Paired t-Tests:

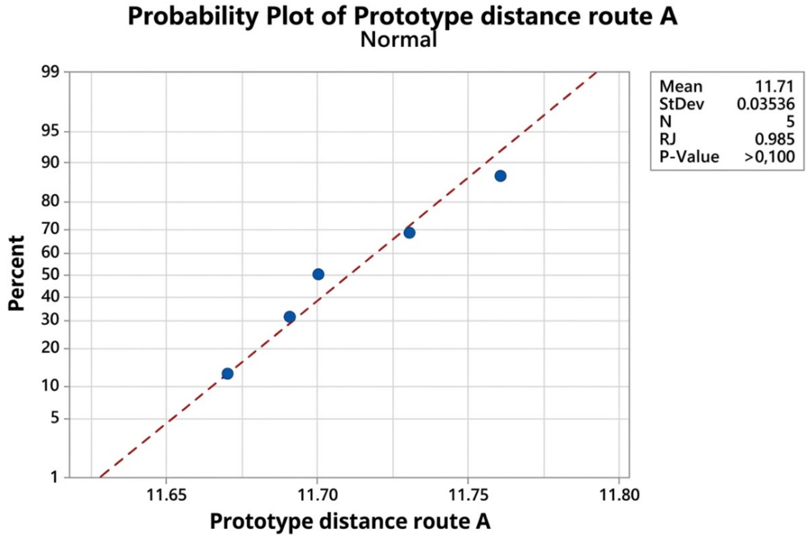

Figure A6. Shapiro-Wilk normality test — Route A ($p > 0.100$).

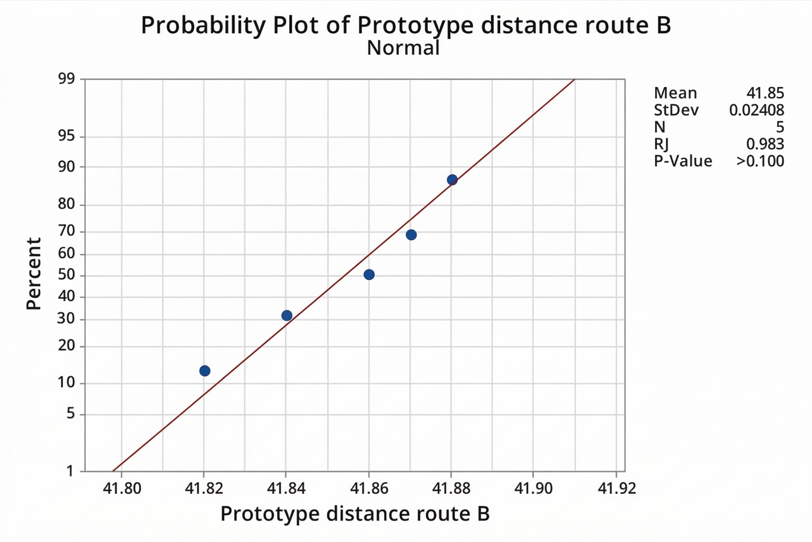

Figure A7. Shapiro-Wilk normality test — Route B ($p > 0.100$).

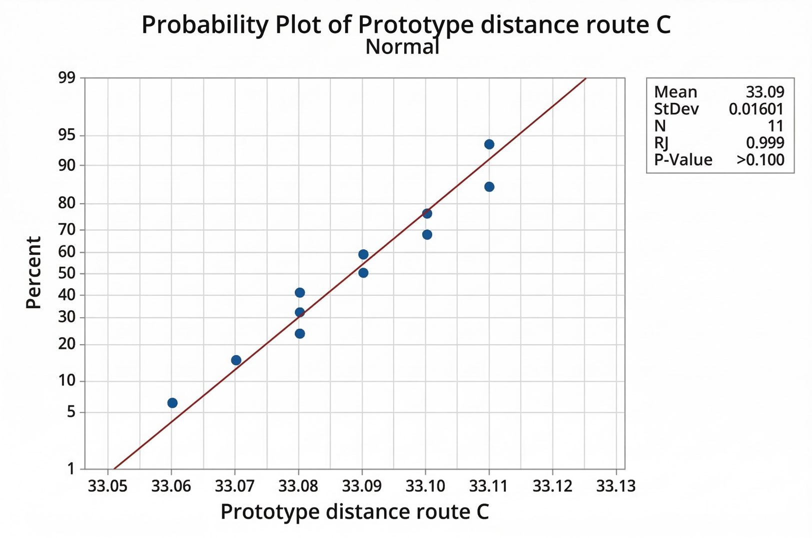

Figure A8. Shapiro-Wilk normality test — Route C ($p > 0.100$).

Paired Student's t-Test

H_0:	mean diff. $= 0$
H_1:	mean diff. $\neq 0$
T value	p value
44.90	0.000

Figure A9. Paired t-test – Route A.

H_0:	mean diff. $= 0$
H_1:	mean diff. $\neq 0$
T value	p value
79.29	0.000

Figure A10. Paired t-test – Route B.

H_0:	mean diff. $= 0$
H_1:	mean diff. $\neq 0$
T value	p value
225.41	0.000

Figure A11. Paired t-test – Route C.

References

1. World Health Organization. *Global Status Report on Road Safety 2023*; WHO: Geneva, Switzerland, 2023. Available online: https://iris.who.int/handle/10665/375016 (accessed on 5 December 2025).
2. Asociación de Empresas Automotrices del Ecuador (AEADE). *Anuario del Sector Automotriz 2023*; AEADE: Quito, Ecuador, 2023. Available online: https://www.aeade.net/ (accessed on 5 December 2025).
3. Soon, T.W.; Osman, Z.; Nafis, M. Investigation of accuracy and response time in GPS-based car mileage monitoring device. *Mekatronika* **2022**, *4*, 1–7. https://doi.org/10.15282/mekatronika.v4i1.7535.
4. Lino, G.V.; Arriaga, J. Análisis de eficacia de un modelo de gestión para mantenimiento preventivo de vehículos en la Policía de Santa Elena. *Polo del Conocimiento* **2025**. Available online: https://polodelconocimiento.com/ojs/index.php/es/article/view/8671 (accessed on 16 December 2025).
5. Instituto Ecuatoriano de Normalización (INEN). *RTE-002: Reglamento Técnico Ecuatoriano de Emisiones de Vehículos Automotores*; INEN: Quito, Ecuador, 1996. Available online: https://www.normalizacion.gob.ec/buzon/reglamentos/RTE-002.pdf (accessed on 16 December 2025).
6. López-Chila, R.; Abad-Reyna, H.; Morocho-Cajas, J.; Fierro-Jiménez, P. Development of representative urban driving cycles for congested traffic conditions in Guayaquil using real-time OBD-II data and weighted statistical methods. *Vehicles* **2025**, *7*, 95. https://doi.org/10.3390/vehicles7030095.
7. Hussain, S.M.; Yang, U.; Shunkun. Car e-Talk: An IoT-enabled cloud-assisted smart fleet maintenance system. *IEEE Internet of Things Journal* **2021**, *8*, 1234–1245. https://doi.org/10.1109/JIOT.2020.2986342.
8. Giraldo, M.; Huertas, J.I. Real emissions, driving patterns and fuel consumption of in-use diesel buses operating at high altitude. *Transportation Research Part D* **2019**, *77*, 21–36. https://doi.org/10.1016/j.trd.2019.10.003.
9. Sanasie, G.; Radzak, Z.I.A.; Zakaria, M.A. Preventive maintenance data logger monitoring system. *Mekatronika* **2021**, *3*, 1–6. https://doi.org/10.15282/mekatronika.v3i1.7154.
10. Farré, V.; Vega-Sánchez, J.; Garzón, V.; Garzón, N.O.; Mora, H.C.; Benitez-Olivo, E.E. Design and performance analysis of sub-THz/THz mini-cluster architectures for dense urban 5G/6G networks. *Sensors* **2025**, *25*, 6717. https://doi.org/10.3390/s25216717.
11. Addo, P.A.; Dwomoh, L.; Ofori, C. Automatic maintenance alert system for heavy duty haulage machines. *Jurnal Nasional Teknik Elektro* **2022**, *11*, 1–8. https://doi.org/10.25077/jnte.v11n2.1002.2022.
12. Hernández-Sampieri, R.; Mendoza, C. *Metodología de la Investigación: Las Rutas Cuantitativa, Cualitativa y Mixta*; McGraw-Hill: Mexico City, Mexico, 2018.
13. Nguyen, K.T.P.; Medjaher, K. A new dynamic predictive maintenance framework using deep learning for failure prognostics. *Reliability Engineering & System Safety* **2019**, *188*, 251–262. https://doi.org/10.1016/j.ress.2019.03.018.

14. Zhang, P.; Gao, Z.; Cao, L.; Dong, F.; Zou, Y.; Wang, K.; Zhang, Y.; Sun, P. Marine systems and equipment prognostics and health management: A systematic review. *Machines* **2022**, *10*, 72. https://doi.org/10.3390/machines10020072.
15. Einabadi, B.; Mahmoodjanloo, M.; Baboli, A.; Rother, E. Dynamic predictive and preventive maintenance planning in the automotive industry. *Journal of Manufacturing Systems* **2023**, *69*, 292–310. https://doi.org/10.1016/j.jmsy.2023.05.012.
16. National Institute of Standards and Technology (NIST). *Handbook 44: Specifications, Tolerances, and Other Technical Requirements for Weighing and Measuring Devices*; NIST: Washington, DC, USA, 2023. https://doi.org/10.6028/NIST.HB.44-2023.
17. Kondaveeti, H.K.; Kumaravelu, N.K.; Vanambathina, S.D.; Mathe, S.E.; Vappangi, S. A systematic literature review on prototyping with Arduino. *Computer Science Review* **2021**, *40*, 100364. https://doi.org/10.1016/j.cosrev.2021.100364.
18. Hernández-Rodríguez, E.; González-Rivero, R.A.; Schalm, O.; Martínez, A.; Hernández, L.; Alejo-Sánchez, D.; Janssens, T.; Jacobs, W. Reliability testing of a low-cost multi-purpose Arduino-based data logger. *Sensors* **2023**, *23*, 7412. https://doi.org/10.3390/s23177412.
19. U-blox AG. *NEO-M8 Series GNSS Modules Data Sheet*; U-blox: Thalwil, Switzerland, 2021. Available online: https://www.u-blox.com/en/product/neo-m8-series (accessed on 16 December 2025).
20. Menaka, M.; Tharani, R.; Pemila, M.; Bhavana, H. GSM-based vehicle anti-theft and maintenance alert device using vibration sensor and Arduino. In *Proceedings of the International Conference on Data Science and Business Systems (ICDSBS)*, Chennai, India, 17 April 2025. https://doi.org/10.1109/ICDSBS63635.2025.11031878.
21. Navixy. SMS in GPS tracking and transport telematics as vital as GPRS. 2020. Available online: https://www.navixy.com (accessed on 16 December 2025).
22. Fawcett, T. An introduction to ROC analysis. *Pattern Recognition Letters* **2006**, *27*, 861–874. https://doi.org/10.1016/j.patrec.2005.10.010.
23. Zhang, H.; Zhang, R.; Sun, J. Developing real-time IoT-based public safety alert and emergency response systems. *Scientific Reports* **2025**, *15*, 13465. https://doi.org/10.1038/s41598-025-13465-7.
24. Althoubi, A.; Alshahrani, R.; Peyravi, H. Delay analysis in IoT sensor networks. *Sensors* **2021**, *21*, 3876. https://doi.org/10.3390/s21113876.
25. Ranacher, P.; Brunauer, R.; Mackaness, W.A.; Van der Spek, S. Why GPS makes distances bigger than they are. *International Journal of Geographical Information Science* **2016**, *30*, 316–333. https://doi.org/10.1080/13658816.2015.1086924.
26. Lopez-Chila, R.; Abad-Correa, A.; Cruz-Zavala, H. Analysis of the influence of driving modes in a hybrid vehicle on fuel consumption optimization. *International Journal of Innovative Research in Social Sciences* **2025**, *8*, 6017. https://doi.org/10.53894/ijirss.v8i2.6017.
27. Agencia Nacional de Tránsito (ANT). *Estadísticas de Siniestralidad de Tránsito: Anuario 2024*; ANT: Quito, Ecuador, 2024. Available online: https://www.ant.gob.ec (accessed on 16 December 2025).