Arduino-Based Early Warning Prototype for Left Vehicle Key Indication Using SW-420

Dinda Melisa Juliani[a1], Muhammad Gald Teary[a2], Fery Antony[a3]
[a]Program Studi Sistem Komputer, Fakultas Ilmu Komputer dan Sains, Universitas Indo Global Mandiri, Palembang, Indonesia.
e-mail: [1]2023310032@students.uigm.ac.id, [2]m.gald@uigm.ac.id, [3]feryantony@uigm.ac.id

Abstrak

Penelitian ini telah merancang purwarupa sistem peringatan dini untuk mencegah kelalaian pengguna yang meninggalkan kunci pada kontak kendaraan. Sistem ini telah menggunakan mikrokontroler Arduino Uno dan sensor getaran (SW-420) untuk memantau aktivitas fisik di sekitar kendaraan. Logika peringatan telah diatur agar indikator cahaya (LED) dan peringatan suara (buzzer) aktif jika getaran tidak terdeteksi selama lima detik. Hasil pengujian menunjukkan bahwa sensor telah merespons getaran tinggi secara sempurna (100%) dan getaran sedang dengan baik (80%), namun kurang sensitif pada getaran rendah (20%). Kesimpulannya, purwarupa peringatan dini ini telah berhasil beroperasi sesuai batas waktu yang dirancang, meskipun memiliki batasan utama yakni belum mampu mendeteksi posisi fisik kunci secara langsung.

Kata kunci: *Sistem keamanan, peringatan dini, Arduino Uno, sensor getaran, kunci tertinggal.*

Abstract

This study designed an early warning prototype system to prevent user negligence of leaving keys in the vehicle ignition. The system utilized an Arduino Uno microcontroller and a vibration sensor (SW-420) to monitor physical activity around the parked vehicle. The warning logic was configured to activate visual (light-emitting diode) and audio (buzzer) indicators if no vibration was detected for five seconds. Testing results showed that the sensor responded perfectly to high vibration (100%) and well to medium vibration (80%), but lacked sensitivity to low vibration (20%). In conclusion, the vibration-based early warning prototype successfully operated according to the designed delay timing, although its main limitation was the inability to physically detect the direct key position.

Keywords: *Vehicle security, early warning, Arduino Uno, vibration sensor, left key.*

1. Introduction

Motor vehicles are essential assets in daily life, but their security is often compromised by user negligence, such as leaving the key in the ignition after parking [1]. This common oversight significantly increases the risk of unauthorized use and theft [2]. As an initial solution, microcontrollers like the Arduino Uno offer a low-cost, easily programmable platform to build a simple security prototype utilizing an SW-420 vibration sensor, an LED, and a buzzer [3].

Previous studies have developed vehicle security systems using various approaches [4]. For instance, Gunawan et al. and Maesa & Hartono utilized vibration sensors combined with GPS for tracking and remote monitoring [5]. However, these existing systems generally treat vibration as a dangerous disturbance or focus solely on user authentication, leaving a gap for systems that monitor post-parking negligence[6].

Unlike traditional alarms, the system in this study reverses the logic: vibration is treated as an indication of the user's ongoing physical activity around the vehicle (e.g., touching the vehicle, closing the seat, or shifting items). When the sensor detects a complete absence of vibration for five seconds, the system assumes the user has walked away and activates the alarm. The main technical contribution of this study is the implementation of a continuous, non-blocking timer using the millis() function, allowing the microcontroller to calculate this duration without halting sensor readings.

This study aims to design and implement a vibration-based early warning prototype using Arduino Uno. However, it must be explicitly emphasized that this system acts strictly as an early warning reminder, not as a direct key-position detector. Because the prototype relies on the absence of movement rather than reading the ignition line directly, future developments are strongly recommended to integrate contact sensors or remote communication modules to achieve definitive key-left detection.

2. Research Method

This research utilized a design and prototype testing method focusing on building an Arduino Uno-based warning system. Using an Input-Process-Output approach, the SW-420 sensor served as the input to read vibration, Arduino Uno as the central processor, and an LED and buzzer as the warning outputs.

A five-second threshold was established as a practical delay so the alarm would not trigger during brief pauses in vibration. This specific duration was empirically estimated based on preliminary observations of typical user disembarkation behavior; it accounts for the average brief pause required to turn off the engine, gather personal belongings, and step away from the vehicle. Crucially, time calculation was executed using the millis() function, allowing sensor reading to continue without the blocking effect of delay()

2.1. Research Stages

The research stages were arranged from problem identification to result analysis so that the design process could be traced clearly and replicated easily.

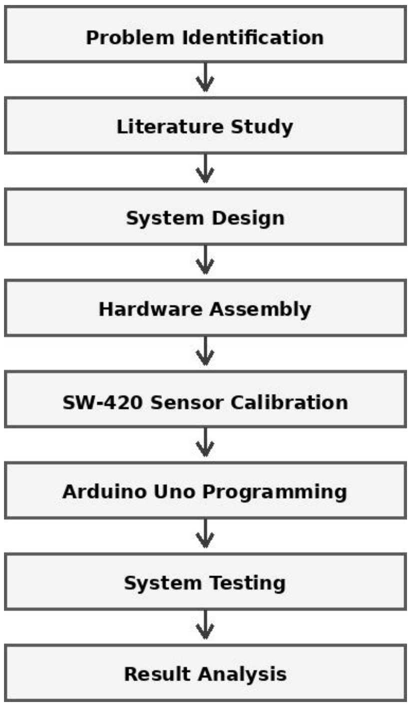

Figure 1. Research stages

Problem identification was based on the possibility that users leave the key in the ignition position after parking. The literature study was used to understand Arduino Uno, the SW-420 vibration sensor, LED, buzzer, and related microcontroller-based vehicle security systems. The design stage included preparing the system workflow, selecting components, and mapping the pin connections. The hardware was then assembled, the sensor was calibrated, the program was written using Arduino IDE and C/C++, and the prototype was tested based on sensor response, timer logic, actuator output, and disturbance conditions.

2.2. System Requirements

Table 1. Hardware Requirements

No	Component	Function
1	Arduino Uno	Main microcontroller for reading the SW-420 sensor, calculating the no-vibration duration, and controlling outputs.
2	SW-420 vibration sensor	Reads vibration activity through the digital output pin.
3	Active 5V buzzer	Provides an audio warning when the alarm status is active.
4	LED	Provides a visual warning indicator when the alarm is active.
5	220 ohm resistor	Limits LED current to keep it within a safe operating range.
6	Breadboard	Temporary assembly medium without soldering.
7	Jumper wires	Connect Arduino Uno to the sensor, LED, buzzer, VCC, and GND.
8	Two 18650 batteries	Main power source for Arduino Uno and the circuit.
9	2 x 18650 battery holder	Holds the batteries and improves power connection stability.
10	DC jack / barrel jack connector	Connects the battery output to the Arduino Uno DC jack.

A simple LED current calculation was performed using Ohm's Law. With a 5V Arduino output, an estimated red LED forward voltage of 2V, and a 220 ohm resistor, the LED current was approximately $(5V - 2V) / 220\ ohm = 13.6\ mA$. This value was suitable for a simple visual indicator. If the buzzer requires a higher current, a transistor driver should be added so the Arduino pin does not directly carry an excessive load.

Table 2. Software and Programming Requirements

No	Tool / Language	Function
1	Arduino IDE	Writing, verifying, and uploading the program to Arduino Uno.
2	Serial Monitor	Observing sensor readings, system time, and alarm status during testing.
3	C/C++ for Arduino	Building the sensor-reading logic, millis()-based timing, and LED/buzzer control.

2.3. Circuit Scheme and Pin Mapping

The circuit used Arduino Uno as the control center. The SW-420 sensor was connected through the digital output pin, while the analog output pin was not used because the system only required vibration readings in digital logic form. The prototype used two 18650 batteries connected to the Arduino Uno through the DC jack rather than through the 5V pin. This connection allowed the battery voltage to pass through the Arduino regulator first.

Two 18650 batteries connected in series produced a nominal voltage of about 7.4V and could reach about 8.4V when fully charged. After the Arduino was active, the SW-420 sensor received 5V from the Arduino, while the LED and buzzer were controlled through digital pins. All components shared the same GND line to keep sensor readings and actuator operation stable.

Table 3. System Pin Mapping

Component	Pin	Arduino Uno Connection	Description
Battery holder 2 x 18650	Positive (+)	Arduino DC jack center	Voltage input to Arduino Uno.
Battery holder 2 x 18650	Negative (-)	Arduino DC jack outer side	Power ground line.
SW-420 sensor	VCC	Arduino 5V	Sensor voltage from Arduino regulator.
SW-420 sensor	GND	Arduino GND	Sensor ground connected to Arduino ground.
SW-420 sensor	DO	D2	Digital input for vibration reading.
LED	Anode (+)	D13 through 220 ohm resistor	Visual warning indicator.
LED	Cathode (-)	GND	LED ground path.
Active 5V buzzer	Positive (+)	D8	Audio warning output.
Active 5V buzzer	Negative (-)	GND	Buzzer ground path.

2.4. SW-420 Sensor Calibration

SW-420 sensor calibration was essential prior to the main tests because its sensitivity heavily relies on the module's potentiometer setting. To prevent false triggers from minor environmental noise , the potentiometer was adjusted while observing the Serial Monitor until the baseline output remained stable at a LOW logic state during no-vibration conditions. The calibration was deemed successful and established as the baseline for all subsequent testing once the sensor ignored a stationary holder but consistently produced a HIGH logic signal in response to medium taps and strong shakes.

2.5. System Block Diagram

The system followed an Input-Process-Output concept. Input came from the SW-420 vibration sensor. Processing was performed by Arduino Uno by reading the sensor, saving the last vibration time, calculating the no-vibration duration, and determining the alarm status. The outputs were an LED as a visual warning and a buzzer as an audio warning.

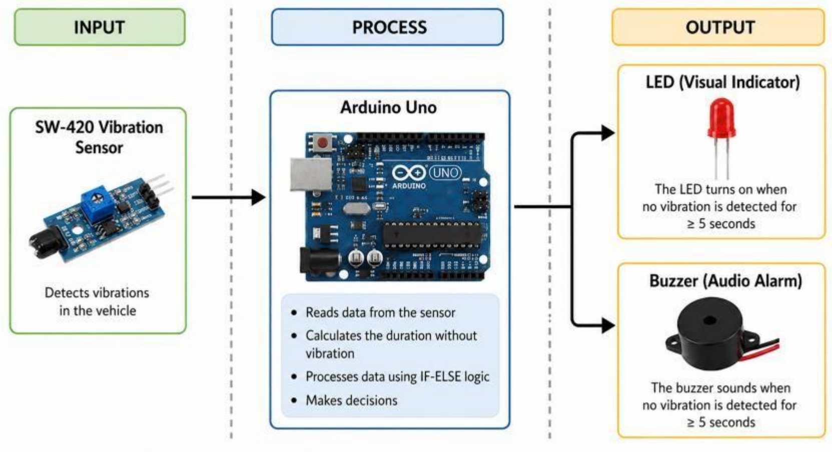

Figure 2. System block diagram

The LED and buzzer turned on when the system did not detect vibration for five seconds or more. When vibration was detected, the system updated the timer and kept the LED and buzzer off. This logic had to remain consistent in the block diagram, flowchart, equations, and program.

2.6. System Flowchart

The system flowchart describes the device workflow from initialization to decision making. The system starts by initializing the sensor, LED, and buzzer pins. Arduino Uno then reads the digital signal from the SW-420 sensor. If vibration is detected, the last vibration time is updated, the LED and buzzer are turned off, and the system returns to monitoring. If no vibration is detected, the system calculates the duration since the last detected vibration. When the duration is less than five seconds, the alarm remains off. When the duration reaches five seconds or more, the LED and buzzer are activated as a warning.

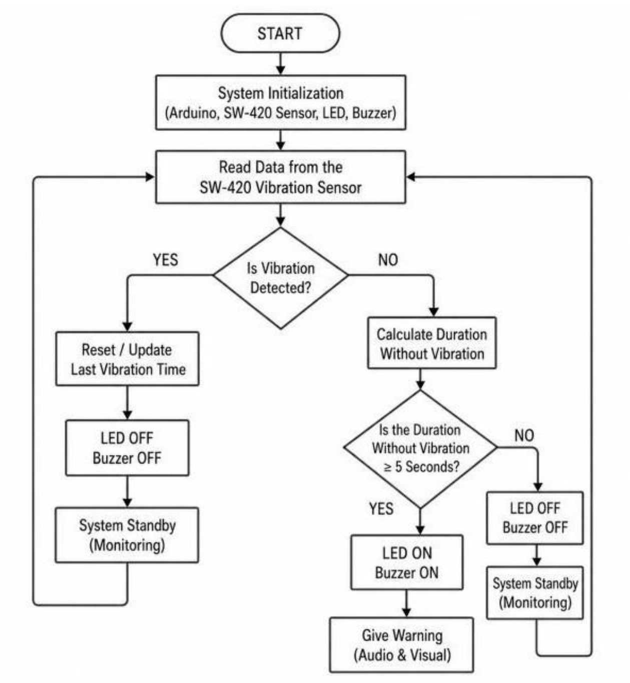

Figure 3. System flowchart

2.7. Algorithm and System Logic

The core algorithm utilized the SW-420 digital output and the non-blocking millis() function to continuously monitor vibration without halting the system. Sensor status (S) was defined as $S = 1$ for detected vibration and $S = 0$ for no vibration. Consequently, the alarm status (A) activated ($A = 1$) only when $S = 0$ and the time elapsed since the last vibration (Δt) was $\geq 5000\ ms$ otherwise, it remained off ($A = 0$). Operationally, the system transitions between three primary states: active monitoring (vibration present, timer updated, alarm off), waiting (no vibration, $\Delta t < 5000ms$, alarm off), and active alarm (no vibration, $\Delta t \geq 5000ms$, LED and buzzer on). Any new vibration detection instantly resets the system back to the active monitoring state.

2.8. Sensor Filtering and Program Implementation

To prevent false triggers from minor environmental noise, a simple digital filter was applied[7]. The sensor was read every 100 ms, and vibration was deemed valid and the timer updated only if detected three consecutive times. Programmed in the Arduino IDE using pins D2 for sensor input, D13 for LED output, and D8 for buzzer output, the core logic strictly utilized the non-blocking millis() function instead of delay() to ensure continuous vibration monitoring. Additionally, to prevent data overload, the Serial Monitor was configured to print logs exclusively during system state changes.

2.9. Testing Design and Procedure

Testing was conducted on a stable bench setup with the sensor securely fixed to a circuit holder. To evaluate reading consistency, three levels of manual vibration (simulating low, medium, and high physical impacts) were applied 10 times each to establish a preliminary detection percentage. Subsequent evaluations verified the correct ON/OFF functionality of the actuators, validated the five-second timer logic by comparing Serial Monitor logs against a stopwatch, and assessed the system's overall resilience to minor environmental disturbances.

Table 4. System Testing Design

No	Test Type	Objective	Number of Tests	Observed Indicator
1	SW-420 sensor test	Measure sensor response to different vibration levels.	10 times per condition	Detected or not detected status
2	Actuator test	Verify LED and buzzer operation when alarm is ON/OFF.	10 times	LED on/off and buzzer sound/off
3	Timer test	Verify alarm timing against the five-second threshold.	Several timing scenarios	Alarm OFF before five seconds and ON after five seconds
4	Disturbance and sensitivity test	Measure the effect of small and strong vibration on system behavior.	10 times per condition	Number of active alarms and logic suitability

It should be noted that conducting 10 trials per condition serves strictly as preliminary testing for this prototype stage, future evaluations should employ a larger sample size to achieve broader statistical validity.

3. Literature study

Previous research has explored various microcontroller-based vehicle security systems, frequently combining vibration sensors, GPS, and IoT modules for remote tracking, or utilizing RFID and reed switches for access authentication [8]. While these systems effectively improve security, they generally treat vibration as a direct trigger for danger or unauthorized access [9]. This study introduces a contrasting approach by utilizing vibration strictly as an indicator of ongoing user activity around the parked vehicle. The warning is uniquely activated only when vibration is completely absent for five seconds or more. This foundational logic addresses post-parking negligence and can be integrated in future developments with ESP8266 modules, RFID, or IoT event logging to build a more comprehensive security solution.

4. Result and discussion

4.1. System Implementation

The prototype was controlled by an Arduino Uno, which processed digital signals from the SW-420 vibration sensor to calculate the no-vibration duration and trigger the warnings. Powered independently by two 18650 batteries via the DC jack, the hardware linked the sensor to pin D2, the LED (with a 220-ohm resistor) to pin D13, and the active 5V buzzer to pin D8. All components shared a common GND line to guarantee stable readings and reliable actuator performance.

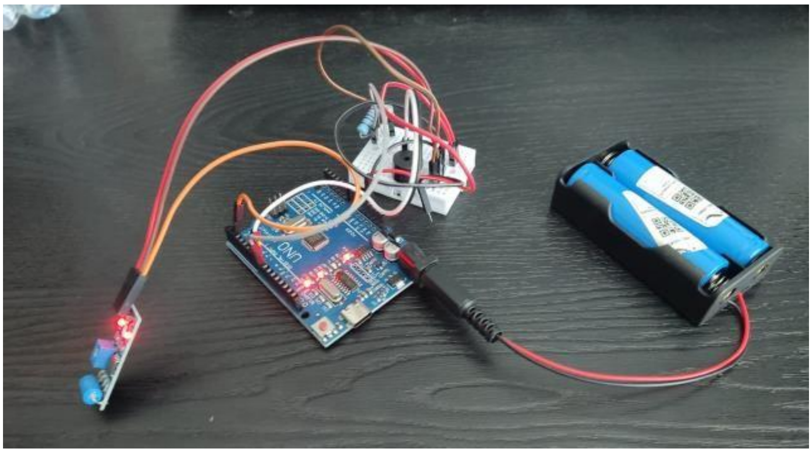

Figure 4. System circuit implementation documentation

Table 5. System Component Implementation Results

No	Component	Observed Parameter	Observation Result	Status
1	Arduino Uno	Power input through DC jack	Arduino turned on and the program ran properly.	Successful
2	SW-420 sensor	Digital output on DO pin	Sensor produced logic changes when vibration was received.	Successful
3	LED	LED condition during active alarm	LED turned on when alarm was ON and turned off when alarm was OFF.	Successful
4	220 ohm resistor	LED current limiter	LED turned on steadily and the current was within the safe calculated range.	Successful
5	Active 5V buzzer	Sound during active alarm	Buzzer sounded when alarm was ON and stopped when alarm was OFF.	Successful
6	Breadboard and jumper wires	Connection stability	The circuit connection remained usable during testing.	Successful
7	Battery holder and barrel jack	System power source	The system could be powered without an adapter through the Arduino DC jack.	Successful

As detailed in Table 5, all major components functioned successfully. This operational success was evaluated beyond mere power-on states, focusing strictly on behavioral adherence to the system design: the LED and buzzer activated exclusively during the ON alarm status, while the SW-420 sensor consistently generated accurate digital logic changes in response to applied vibrations.

4.2. SW-420 Sensor Testing Results

The SW-420 sensor was evaluated across three simulated real-world vibration levels: 'low' (a singular, low-force contact), 'medium' (consecutive, moderate impacts at ~2 Hz, e.g., closing a motorcycle seat), and 'high' (continuous, high-force oscillation >3 Hz, e.g., chassis shifting). Detections were considered valid only if the DO pin output consistently appeared on the Serial Monitor. Because these vibrations were applied manually, the findings establish a preliminary baseline requiring future validation with quantified acceleration metrics. For brevity, data from all 30 trials is synthesized in the summary table below.

Table 6. Summary of SW-420 Sensor Detection Results (10 Trials per Condition)

Vibration Condition	Valid Detections (Out of 10)	Missed Detections	Detection Percentage	Interpretation
Low	2	8	20%	The sensor was not consistent in reading low vibration.
Medium	8	2	80%	The sensor was responsive enough to medium taps.
High	10	0	100%	The sensor responded most stably to strong vibration.

Detection percentages (valid detections divided by total trials) demonstrate that the SW-420 sensor responds optimally to medium and high vibrations. Conversely, with only a 20% success rate for low vibrations, it proved unreliable for minor movements. Consequently, the system depends on medium to strong physical activity for consistent operation.

4.3. LED and Buzzer Actuator Testing Results

Actuator testing was conducted to ensure that the LED and buzzer worked according to the alarm status. The LED acted as a visual warning indicator, while the active 5V buzzer acted as an audio warning. The test was performed by activating and deactivating the alarm status 10 times.

Table 7. LED and Buzzer Actuator Testing Results

Trial	Alarm Status	LED	Buzzer	Measured Response Time (ms)
1	ON	On	Sounding	~2 ms
2	ON	On	Sounding	~1 ms
3	ON	On	Sounding	~3 ms
4	ON	On	Sounding	~2 ms
5	ON	On	Sounding	~2 ms
6	OFF	Off	Off	~1 ms
7	OFF	Off	Off	~2 ms
8	OFF	Off	Off	~2 ms
9	OFF	Off	Off	~1 ms
10	OFF	Off	Off	~2 ms

Based on Table 7, the LED and buzzer worked perfectly according to the alarm status. Instead of relying on qualitative direct observation, the response time was quantitatively estimated using Serial Monitor timestamps—tracking the exact millisecond difference between the logic condition being met and the digitalWrite() execution. Across all trials, the execution delay was consistently between 1 to 3 milliseconds. This rapid response firmly demonstrates the processing efficiency of the non-blocking millis() logic.

4.4. Five-Second Timer Logic Testing Results

Timer logic testing was performed to ensure that the alarm did not turn on before the no-vibration duration reached five seconds. This test was important because the system was not only designed to read vibration, but also to calculate the time since the last detected vibration.

Table 8. Five-Second Timer Logic Testing Results

No	Testing Scenario	Expected Condition	Observation Result	Status
1	Vibration is detected	Alarm OFF	Alarm OFF	Suitable
2	No vibration for 1-4 seconds	Alarm OFF	Alarm OFF	Suitable

3	No vibration for >= 5 seconds	Alarm ON	Alarm ON	Suitable
4	Vibration appears again before 5 seconds	Timer is reset and alarm OFF	Alarm OFF	Suitable
5	Vibration appears at about 4.9 seconds	Alarm does not activate and timer is reset	Alarm OFF	Suitable
6	No vibration after timer reset	Alarm activates again after >= 5 seconds	Alarm ON	Suitable

The results showed that the alarm stayed off while the no-vibration duration had not reached five seconds. When vibration appeared again before the time threshold, the last vibration time was updated so the alarm remained off. The scenario near 4.9 seconds was used to ensure that the alarm did not turn on prematurely. Thus, the millis() function could run the timing logic without stopping sensor reading.

4.5. Disturbance and Sensitivity Testing Results

Disturbance and sensitivity testing was conducted to observe the effect of surrounding conditions on SW-420 readings. The table is not referred to only as false trigger testing because an active alarm during the no-vibration condition was actually consistent with the system logic. An error would occur when the alarm turned on in a condition that should still be considered active movement, or when the alarm did not turn on in a condition that should produce a warning.

Table 9. Disturbance and Sensitivity Testing Results

No	Testing Condition	Number of Tests	Active Alarm	Interpretation
1	Sensor still without vibration	10 times	10 times	Consistent with system logic because no vibration was detected until the time threshold was reached.
2	Light touch on sensor holder	10 times	8 times	Light touch was not consistently enough to reset the timer.
3	Medium vibration on sensor holder	10 times	2 times	Most medium vibrations successfully prevented the alarm from activating.
4	Strong vibration on sensor holder	10 times	0 times	Strong vibration was read stably, so the alarm did not activate.

As intended by the system logic, the alarm correctly activated after five seconds of inactivity-representing a true positive rather than a false trigger. However, without a key-position sensor, this logic inevitably warns safely parked vehicles. Regarding sensitivity, the SW-420 reliably handled medium and strong vibrations but struggled with low vibrations, failing to reset the timer in 80% of light-touch trials.

4.6. System Response Analysis

Overall, the system successfully executed its designed workflow by keeping the LED and buzzer deactivated during detected vibrations, and triggering the warnings only after five seconds of inactivity. However, this core logic reveals a fundamental limitation. Because a safely parked vehicle with the key properly removed also produces no vibration, the system will inevitably activate the alarm under normal, secure conditions. Consequently, this device cannot be classified as a definitive key-left detector. Instead, it serves strictly as an early warning prototype designed to remind users to recheck their ignition once physical activity around the vehicle ceases. To evolve into a robust detection system, future iterations must integrate a direct key-position sensor, contact sensor, or ignition line reading.

4.7. Reading Error Analysis

Based on the sensor and disturbance test results, reading errors primarily occurred during low vibration scenarios, which yielded only a 20% detection rate compared to the highly reliable medium (80%) and high (100%) vibrations. Within this system context, errors manifest in two distinct ways. First, the alarm might falsely trigger if ongoing user activity produces vibrations too minor to surpass the sensor's threshold. Second, the alarm will activate even when the vehicle is safely parked and the key is removed. This second scenario is not a programming flaw but a fundamental design limitation, as the prototype cannot directly verify the physical key position. Consequently, the system's true accuracy as a definitive "key -left" detector cannot be fully quantified; the measurable accuracy in this study remains strictly limited to vibration detection and timer logic execution.

4.8. Comparison with Previous Studies

Table 10. Comparison with Previous Studies

No	Study	Main Sensor / Component	System Output	Strength	Limitation
1	Gunawan	Vibration sensor and GPS based on Arduino	Vehicle security and tracking	Supports vehicle location monitoring	Does not focus on key-left indication.
2	Maesa and Hartono	SW-420 vibration sensor and Neo-6M GPS based on IoT	Vehicle control and monitoring	Uses SW-420 and supports remote monitoring	More complex because it needs communication and GPS modules.
3	Martias	RFID and Arduino Uno	Vehicle access authentication	Stronger for user access limitation	Does not read vibration or post-parking vehicle condition.
4	Anwar and Latifah	Reed switch and ATMEGA16	Vehicle security with immobilizer	Closer to reading contact or position condition	Requires more specific sensor placement on the vehicle.
5	This study	SW-420, Arduino Uno, LED, and buzzer	Local audio and visual warning	Simple circuit and non-blocking timer logic	Does not directly detect key position.

This study distinguishes itself from previous research through its reversed alarm logic approach. While existing systems typically use vibration sensors as direct danger triggers or combine them with GPS and IoT for continuous monitoring , this prototype uniquely utilizes vibration to indicate ongoing human activity around the vehicle , activating the alarm only after a five-second absence of movement. The primary technical contribution here is the integration of non-blocking millis() timer logic within an SW-420-based warning framework. Nevertheless, compared to systems utilizing reed switches, this prototype remains limited as it cannot directly verify the physical state of the key or the ignition line.

4.9. Power Supply and System Limitation Analysis

The system operates independently without an adapter by utilizing two 18650 batteries connected through the Arduino's DC jack. Because current consumption was not measured directly with a multimeter during this study, the exact battery duration remains phy sically unquantified. However, a theoretical baseline can be established using component datasheet estimations alongside the formula $t = C / I$ (where t is duration, C is battery capacity in mAh, and I is the average system current in mA). Assuming the Arduino Uno draws approximately $50\ mA$, the LED $13.6\ mA$, and the active buzzer $30\ mA$, the total estimated current during an active alarm state is roughly $93.6\ mA$. Utilizing two standard $2000\ mAh$ 18650 batteries, the theoretical continuous alarm duration would be $t = 2000\ mAh/93.6mA\ \approx 21.3\ hours$. This calculation provides an essential functional estimate, highlighting the necessity for future testing to physically measure current draws during both standby and active alarm states to accurately predict real - world battery life.

Crucially, the system's primary limitation is its exclusive reliance on the SW-420 sensor; because it only reads vibration, the system cannot distinguish between a vehicle with an abandoned key and one that is safely parked with the key removed. Consequently, it functions strictly as a preliminary early warning prototype. To evolve into a definitive security system, future development must prioritize integrating direct key-position sensors or ignition line readings. Once reliable key detection is established, the system could be significantly enhanced with ESP8266 or ESP32 modules for mobile alerts, GPS for location tracking[10], RFID for access control, and IoT integration for comprehensive remote monitoring [11] ignition after parking.

5. Conclusion

An Arduino Uno and SW-420-based early warning bench prototype was successfully designed using non-blocking millis() logic to continuously monitor vehicle vibration without halting the program. Testing validated the system's functionality: the sensor successfully detected high (100%) and medium (80%) vibrations, though it lacked consistency for low vibrations (20%). Furthermore, the timer and actuators performed correctly by activating the visual and audio warnings strictly when no vibration was detected for five seconds or more. However, because the system assumes a key is left behind based solely on the absence of vibration, it fails to distinguish this from a safely parked vehicle and cannot be classified as a final key-left detector. Future development must explicitly integrate a key-position sensor or ignition line reading to provide valid detection beyond just vibration, and transition from this laboratory bench testing scenario to real - world vehicle installations for field validation.

References

[1] J. Jurnal, I. Teknik, E. Komputer, A. Rahman, A. Stefanie, and B. Santoso, “Perancangan Sistem Instrumentasi Berbasis Internet Of Things Pada Sistem Keamanan Pencurian Kendaraan Bermotor (SIKESMI)”.

[2] Gunawan, Achmad Yani, Junaidi, Zumhari, E Hutajulu, and R Sirait, “Design And Development of a Vehicle Security System Using Vibration Sensors and GPS Based on Arduino,” *J. Inf. Technol. Comput. Sci. Electr. Eng.*, vol. 1, no. 3, pp. 527–535, 2024, doi: 10.61306/jitcse.v1i3.152.

[3] Fernando Maesa and H. Hartono, “Rancang Bangun Sistem Kontrol dan Monitoring Keamanan Sepeda Motor Berbasis IoT dengan Modul GPS Neo-6M dan Sensor Getar SW-420,” *J. Surya Tek.*, vol. 12, no. 1, pp. 37–43, 2025, doi: 10.37859/jst.v12i1.8532.

[4] M. Martias, M. Abby Ghaly, and H. Adianto, “Alat Pengaman Kendaraan Berbasis RFID Menggunakan Mikrokontroller Arduino Uno,” *IMTechno J. Ind. Manag. Technol.*, vol. 5, no. 2, pp. 24–30, 2024, doi: 10.31294/imtechno.v5i2.3214.

[5] R. Septian Anwar and F. Latifah, “Perancangan Alat Keamanan Kendaraan dengan Immobilizer Menggunakan Sensor Reed Switch Berbasis ATMEGA16,” *J. Tek. Inform. STMIK Antar Bangsa*, vol. 4, no. 2, p. 125, 2019, doi: 10.51998/jti.v4i2.254.

[6] A. B. S. Sugiakto, Desiyanna Lasut, and E. Dwiyanthi Kusuma, “Rancang Bangun Sistem Keamanan Kendaraan Bermotor Berbasis Iot Dengan Menggunakan Arduino Dan Kartu Rfid,” *Algor*, vol. 5, no. 2, pp. 8–19, 2024, doi: 10.31253/algor.v5i2.2363.

[7] N. C. Putri, M. G. Teary, G. Mandiri, J. L. Jend, and S. No, “Analysis of the Sensitivity of

the SW-420 Vibration Sensor to Minor Mechanical Disturbances in the Safe Security System," vol. 6, no. 3, 2025.

[8] C. M. G. Butar-butar and Y. T. Samuel, "PERANCANGAN SISTEM KENDALI KENDARAAN BERMOTOR JARAK JAUH MENGGUNAKAN NodeMCU ESP8266," *TelKa*, vol. 9, no. 01, pp. 87–97, 2019, doi: 10.36342/teika.v9i01.793.

[9] S. Atin, D. Abdullah, Y. Darmi, and M. H. Rifqo, "Sistem Keamanan Kendaraan Roda Dua Berbasis Internet Of Things (IOT)," *J. Media Infotama*, vol. 19, no. 2, pp. 272–277, 2023, doi: 10.37676/jmi.v19i2.3948.

[10] Abdur Rohman Wakhid, Ulul Ilmi, Affan Bachri, Rifky Aisyatul Faroh, Eko Wahyu Santoso, and Gilbran Bintang Erlangga, "Sistem Keamanan Sepeda Motor menggunakan Modul GPS Neo-7M, Selenoid Key Berbasis ESP8266," *Jural Ris. Rumpun Ilmu Tek.*, vol. 5, no. 1, pp. 240–252, 2025, doi: 10.55606/jurritek.v5i1.7432.

[11] L. Velasquez-Jimenez, C. Marrujo-Ingunza, S. Rubiños-Jimenez, J. Grados-Gamarra, and H. Grados-Espinoza, "IoT-Based Vehicle Security System: Real-Time Monitoring and Event Logging," *Int. J. Saf. Secur. Eng.*, vol. 15, no. 5, pp. 987–996, 2025, doi: 10.18280/ijsse.150512.