WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Smart Home Security System with Automatic Phone Calling Using Arduino and IoT

[1]Satyam Jha, [2]Sonu Dhakar, [3]Anurag Sharma, [4]Yash Sharma, [5]Dr. Akhil Pandey, [6]Er. Mohit Mishra, [7]Dr. Vishal Shrivastav

[1234]B.Tech Scholar, [5]Professor, [6]Professor,[7]Professor
Computer Science Department, Arya college of engineering and I.T., Kukas, Jaipur, Rajasthan
[1]jhasatyam305@gmail.com, [2]sonudhakar87@gmail.com, [3]anuragbtp561@gmail.com, [4]yasharmavivekvihar@gmail.com, [5]akhil@aryacollege.in, [6]Er.Mohit Mishras, [7]vishal500371@yahoo.co.in

ABSTRACT –

Residential and commercial security remains a global issue, and this has resulted in continuous innovation in surveillance and alarm technology. Older systems such as closed-circuit television (CCTV) cameras and conventional alarm systems primarily rely on human monitoring or manual attention. Consequently, legacy systems are usually plagued by slower response times, lack of coverage, and higher operational expenses. On the contrary, the advent of the Internet of Things (IoT) has made it possible for more advanced and affordable security systems to be developed, which can actively warn users of impending threats. This research paper provides an elaborate design and analysis of a Smart Home Security System based on Arduino or Node MCU microcontrollers, Passive Infrared (PIR) sensors, and wireless communication modules (Bluetooth HC-05 or Wi-Fi) for detecting intrusions and automatically placing calls immediately.

Key Words - Smart Home Security, Internet of Things (IoT), Arduino, Node MCU, PIR Motion Sensor, Bluetooth HC-05, Wi-Fi, Intrusion Detection, Automatic Phone Calling, Home Automation.

1. Introduction

Old security systems tend not to have the instant alert facility in case of any emergency. In this project, a Smart Home Security System with Automatic Phone Calling based on Arduino and IoT has been developed with the ability to detect intrusion and alert the user instantly. Incorporating motion sensor and door sensors with Arduino microcontroller and IoT service, a home can be monitored real-time. As soon as it detects an intruder, it sends an alert and automatically calls or messages the user, delivering instant response and enhanced security. The cost-effective, scalable option offers a safe way of home monitoring using state-of-the-art technology.

1.1. Background

Security and surveillance have become increasingly significant concerns for homeowners and business operators worldwide, particularly in an era of rising urbanization and technological accessibility. Traditional security systems, including door and window alarms, have been instrumental in deterring potential intruders, yet they often lack the capability to actively alert property owners or law enforcement in real time. In the same vein, closed-circuit television (CCTV) systems, although able to record images, need constant human surveillance in order to make sense of occurrences, detect threats, and launch corresponding reactions. Such dependence on human control has the potential to trigger delayed actions, thus undermining the entire efficiency of the system.

The proposed Smart Home Security System in this paper overcomes these challenges through the use of open-source hardware (Arduino or Node MCU), commonly available sensors (Passive Infrared or PIR), and wireless communication modules (Bluetooth HC-05 or Wi-Fi). Integrating the phone calling functionality into the system, the solution provides instantaneous and attention-capturing alerts, as opposed to the delayed or easily missed push notifications most current solutions use. By this project, shop owners and homeowners are able to enjoy a cost-efficient, scalable, and easy-to-use security system that greatly boosts their peace of mind.

1.2. Problem Statement

Even with a multitude of security solutions on offer in the marketplace, such as advanced surveillance cameras and advanced alarm systems, most of these products are either unaffordable or insufficient in basic features like instant phone alerts. It further adds the overhead of additional recurring costs

and dependencies with certain solutions relying on cloud-based services via subscription models. For homeowners or small business owners who might not be able to afford ongoing professional monitoring, this situation presents a huge risk to their properties.

2. Literature Review

2.1. Evolution of Traditional Security Systems

In the past, property owners have used basic mechanical or electrical intrusion detection systems like magnetic contacts on doors and windows or infrared beams that trip alarms when disrupted. While these systems are relatively low-cost and easy to install, they usually only produce local alarms, which can or cannot be heard by the owner of the property or adjacent neighbors. As a development, CCTV systems became popular late in the 20th century, making it possible for events to be monitored and recorded in real time. These systems, however, had to be monitored continuously by humans, either physically present or through security firms working remotely, which added costs and the risk of delays in reaction times.

2.2. Rise of IoT and Security Impacts:

The advent of the Internet of Things (IoT) at the beginning of the 21st century brought about a new paradigm in security system conceptualization and implementation. Through its capability to make various devices, like sensors, microcontrollers, and actuators, communicate via the internet, IoT provided new opportunities for sharing data in real-time, automated decision-making, and remote control. This innovation gave rise to a boom in research on smart home technology, which integrated IoT devices for use in temperature control, lighting, and security.

2.3. Arduino, Node MCU, and Other Microcontrollers in Security:

Open-source microcontrollers like Arduino and Node MCU have been central in democratizing the development of IoT. Arduino with the ATmega328P microcontroller offers an accessible platform to program and prototype and is supported by a very large community of developers. Node MCU, with the ESP8266 Wi-Fi module, enables easy internet connectivity and is particularly well-suited for cloud-based applications.

2.4. Wireless Communication Modules and Protocols

Wireless communication is the core feature of contemporary IoT security systems. Bluetooth modules like HC-05 or HM-10 are used for short-range communication and offer a simple way of linking a microcontroller to a smartphone or other devices locally. Bluetooth communication is generally practiced for small home environments where the user's smartphone remains at a short distance from the microcontroller. This method may have the benefit of relatively low power usage and simple pairing mechanism. Nevertheless, Bluetooth's relatively low range (typically up to 10 meters for Class 2 devices) can be restrictive in larger rooms.

3. Proposed System Architecture

3.1 Overview of the System

The system to be proposed will be designed to detect intruder entry or movement within a residential or small business environment and then make a phone call to notify the property owner. The heart of this system is a microcontroller—either an Arduino UNO or a Node MCU—connected to a PIR motion sensor. The PIR sensor continuously monitors changes in infrared radiation, which serves as an indicator of human presence. When an intrusion is detected, the microcontroller processes the signal and, upon verification, triggers an alert mechanism that comprises both a mobile application notification and an automatic phone call.

• **Communication Module (Bluetooth HC-05 or Wi-Fi on Node MCU):** Facilitates data transfer between the microcontroller and the user's smartphone or a remote server.

• **Mobile App:** Provides immediate notifications and enables users to set system parameters.

• **Auto Calling Interface (VoIP API or GSM Module):** Makes an outgoing call to a preconfigured number when it detects an intrusion.

3.2 Hardware Components and Their Roles

The Arduino UNO, which uses the ATmega328P microcontroller, is well known for its simplicity to program and prototype. It offers digital input/output pins that easily interface with the PIR sensor, Bluetooth module, and other peripherals. On the other hand, the Node MCU (which is ESP8266-based) supports Wi-Fi connectivity, and hence direct access to cloud services and remote servers is possible without any extra hardware modules. Selection between Arduino UNO and Node MCU depends heavily on the user's network requirement.

The communication is managed either using the Bluetooth HC-05 module or by utilizing the Node MCU's onboard Wi-Fi capabilities. Bluetooth is suitable for near communication when the smartphone of the user is close at hand, and Wi-Fi provides the benefit of being internet connected so that messages can be sent even when the user is away from the location.

4. Methodology

4.1. Research Approach

The project research methodology applied here is a design-build-test cycle, beginning with the determination of the functional requirements of an intelligent home security system. The system was progressively refined through laboratory testing, user feedback, and performance data collected during actual-life trials via iterative prototyping. This cyclical process made sure that the final design not only achieved.

4.2. Algorithmic Flow and Motion Verification

One of the critical methodological aspects of this project is the motion verification process. Although one pulse from the PIR sensor can signify movement, it can also be caused by short-term changes in the environment, like ambient temperature changes or small vibrations. In order to deal with this issue, a quick algorithmic loop was introduced to check the sensor output a number of times in a given period of time (typically a few seconds). If the PIR sensor remains high for a large number of these tests, the system concludes that the motion is likely due to an actual intruder and therefore triggers the alert process.

This approach is encapsulated in the pseudo-code shown below:

arduino

CopyEdit

```
readSensorValue = digitalRead(PIR_PIN);
if (readSensorValue == HIGH) {
   int confirmationCount = 0;
   for (int i = 0; i < VERIFICATION_CYCLES; i++) {
      delay(VERIFICATION_DELAY);
      if (digitalRead(PIR_PIN) == HIGH) {
         confirmationCount++;
      }
   }
   if (confirmationCount >= THRESHOLD) {
      triggerAlert();
   }
}
```

4.3. Communication Mechanisms

Within the context of this work, two major modes of communication are employed: Bluetooth and Wi-Fi. Bluetooth uses the technique of pairing the HC-05 module with the user's smartphone, which uses the custom mobile application. Once an intrusion has been detected, data is passed from the microcontroller to the smartphone through serial communication using Bluetooth. The phone can then have a notification message pop up or even utilize its own dialer to place an outgoing call.

For comparison, the Wi-Fi approach makes use of the ESP8266 chipset on the Node MCU. The microcontroller, which is linked to a local router, can send data to the cloud service or mobile app of the user directly via the internet. This strategy is beneficial in situations where the user would be physically away from the property because it removes the range limitation of Bluetooth. But it brings in network dependency, i.e., the effectiveness of the system depends on the availability and reliability of the local internet and Wi-Fi service.

4.4. Automatic Phone Calling Mechanism

The automatic phone calling feature can be implemented in multiple ways, each with distinct pros and cons. When a GSM module (e.g., SIM900A) is used, the microcontroller sends AT commands to the module, instructing it to dial a specific phone number. This one is rather straightforward but requires a functioning SIM card with voice, and it could be expensive depending on the charges of the local telecommunications. An alternative can also be deployed where there is an integration of a VoIP API such as Twilio into the system if the Node MCU is networked with the internet. The microcontroller sends the request to the API endpoint, and the service places the call on the user's phone. This strategy relies on an reliable internet connection but potentially allows for increased call routing and other web-based integration flexibility.

4.5. Testing and Validation Protocols

For ensuring operation of the system as safely as possible under different conditions, a test and validation process has been established. It includes: 1. **Controlled Lab Tests**: Verifying the PIR sensor's response to controlled motion, the microcontroller response time, and testing the calling feature accuracy in an interference-free environment.

2**. Field Tests:** In-place installation in typical commercial or residential buildings to test performance under real, normal conditions, including realistic intrusions, normal occupant traffic patterns, and environmental changes (e.g., temperature fluctuation, drafts, pets).

3. **Stress Tests**: In a bid to expose the system to rapid sequential stimulation, power undulations, and network failures in attempting to test its stability.

4. **User Feedback**: Gathering qualitative and quantitative information from users who interact with the system, in terms of ease of installation, easy to use, and perceived reliability.

Through this holistic testing strategy, the study intends to provide an exhaustive evaluation of the system's weak and strong points to direct future improvement and adoption initiatives.

5. Implementation

5.1 Hardware Setup and Circuit Design

The hardware setup generally starts with placing the microcontroller (Arduino UNO or Node MCU) on a breadboard or custom-printed circuit board (PCB) and then attaching the PIR sensor. The output pin of the sensor is attached to a digital input pin on the microcontroller, whereas the VCC and GND pins of the sensor are attached to the corresponding 5V (or 3.3V) and ground rails. A pull-down resistor may be used to stabilize the sensor output if required, although the feature is often built into most PIR sensor modules.

In case of Bluetooth-based systems, the HC-05 module is interfaced to the TX and RX pins of the microcontroller in a way that the baud rate defined in the code is identical to the default or programmed baud rate of the module (typically 9600 bps). For Node MCU-based systems using Wi-Fi, the ESP8266 chip is integrated onto the board and no external module is needed. When using a GSM module, the microcontroller interfaces with it through a serial interface and, for functions such as calling, sending SMS, or checking signal strength, using AT commands are issued.

5.2 Firmware Development

The firmware for the microcontroller controls all the internal operations, ranging from the reading of PIR sensor to the triggering of the calling mechanism. Within the setup() function, the code sets up serial communication (for module speech and debug), input/output pin setups, and debug printouts for confirming proper setup. The loop() function is the one that repeatedly runs, polling the output of the PIR sensor and, when motion is detected, invoking the verification process.

After verification of a valid intrusion by the verification, the code then proceeds to the alert routine, which typically involves ringing a buzzer or LED, and alerting the corresponding mobile app (via Bluetooth or Wi-Fi) and instructing the GSM module or VoIP API to make a call. Detailed logging can be employed to record each event's timestamp and outcome, such that analysis or debugging in the future would be facilitated. In high-end implementations, interrupts can be used to find power optimization by allowing the microcontroller to sleep in a low-power state and wake up only when the PIR sensor triggers an interrupt pin. The loop() function is the one that repeatedly runs, polling the output of the PIR sensor and, when motion is detected, invoking the verification process.

5.3 Mobile Application Development

The mobile application is one of the most important user interface components of the system, allowing the owners to view real-time activities, configure settings, and manage telephone call preferences. Developed using Kotlin (native Android development) or Python (Kivy) (for cross-platform compatibility), the application typically consists of the following modules:

1.Connection Manager: Manages Bluetooth pairing or Wi-Fi connections, depending on selected communication.

2. Dashboard: Indicates sensor state (disarmed or armed), recent notifications, and system history.

3. Settings Page: Enables the user to set up phone numbers to be used with emergency calls, set sensitivity levels of PIR sensors, and enable/disable additional features like a buzzer or LED.

4. Notification Handler: Processes data packets from the microcontroller, updating the user interface and optionally displaying local notifications or system-level alerts.

In using a Wi-Fi-based solution, the mobile app can also incorporate cloud services for storing and reading event logs, thereby allowing users to see past data even when their phone is not physically connected to the microcontroller during the event.

5.4 Automatic Calling Interface

The system's auto-call interface can be programmed to adapt to various infrastructure and user preferences. For GSM-based calls, the microcontroller merely sends AT commands to dial the user's phone number. An example command is Serial. Print ln("ATD+1234567890;"), which causes the GSM module to make a call to number 1234567890.After a delay interval programmed (e.g., 30 seconds), the microcontroller can issue another command, Serial. Print ln("ATH");, to drop the call.

In order to integrate with the VoIP API, Node MCU or Arduino board with the Wi-Fi module would make an HTTP POST call to the service endpoint. User's phone number, authentication token, and additional parameters that may be needed by the API could be included in the request body. The service will then initiate an outgoing call by bridging the user's phone and the system. It is highly customizable with configurable call flows but depends upon a good network connection and might have usage bills based on the billing scheme of the VoIP provider.

6. Experimental Results

6.1 Experimental Setup and Procedure

To critically test the performance of the presented security system, experiments were performed both in controlled lab conditions and actual residential environments. In experiments conducted in the lab, the system was installed in a 5m x 6m room under minimal extraneous disturbance, enabling researchers to accurately record the time from detection of motion to alarm generation. Motion occurrences were replicated by getting a volunteer walk into the range of the sensor at different speeds and angles.

6.2 Detection Accuracy

Detection accuracy was defined as the ratio of true intrusion detections to the total number of actual intrusions. In the lab setting, of 100 simulated intrusion events, the system accurately detected 98, a 98% detection rate. Two were missed due to the volunteer moving through the sensor's outer edge, suggesting the potential for sensor relocation or multiple sensors in large areas.

6.3 Response Time

Response time is an important measure for security applications since any lag in alert generation would jeopardize the user's capability to act quickly. In the controlled lab environment, the average time from motion detection to phone call initiation was found to be around four seconds. This delay was split as follows:

- Sensor detection and microcontroller processing: <1 second
- Verification loop: ~1 second
- Call initiation over GSM or VoIP: 2 seconds (can be different depending on the network conditions)

In home testing, the response time sometimes rose to 5–6 seconds because of uneven cellular network latency and small processing delays on the smartphone or cloud server. In spite of these small variations, the system always proved to have the ability to notify the user quickly, well beyond the capability of the conventional systems based on manual checks.

7. Discussion

7.1 Advantages and Practical Implications

The experiments point out a number of important strengths of the advanced Smart Home Security System proposed. Most important among these is the real-time phone call facility, which effectively commands the user's instant attention. This functionality plugs an important lacuna in current solutions where the use of SMS or push notification can result in missed or delayed alarms. In addition to this, the utilization of open-source hardware and software dramatically reduces the entry point, making advanced security technology more accessible to a wide range of stakeholders.

7.2 Limitations and Challenges

Even though the system has its strengths, it has limitations as well. The first-order problem is false alarms, especially in residential areas where pets or environmental conditions are changing. While the short verification loop is an excellent help in preventing this problem, customers might have to spend time on sensor placement and calibration to get the best result. Further, the reliance on network connectivity—over Bluetooth, Wi-Fi, or GSM—means that service outages could render the system unable to initiate a call or send notifications.

Another concern is energy consumption. While the system can be operated through a standard electrical plug, installations in distant or off-grid areas may need battery or solar power options. Providing a stable and continuous power supply is essential to ensure the reliability of the system. Additionally, in multi-story buildings or big commercial complexes, extra hardware like signal repeaters or multiple microcontrollers might be needed to effectively cover all concerned zones.

7.3 Ethical and Legal Considerations

Moral grounds mandate that whatever surveillance or monitoring technology is applied, it must be done so responsibly and according to local legislations. Even though the current design does not inherently demand the utilization of cameras, video or audio capture add-ons in future extensions would need far-reaching consideration regarding privacy legislations.What's more, persistent false alarms leading to police calls can be taxing on public resources and even result in legal penalties for the user. Therefore, user training and system calibration are key to preventing the dispatch of calls or alarm fatigue.

8. Conclusion and Future Work

This paper proposed a detailed design, implementation, and evaluation of a Smart Home Security System with open-source microcontrollers, that is, Arduino or Node MCU, PIR motion sensor, and wireless modules for intrusion detection and automatic phoning in the case of detection. A low-cost, but highly effective solution for home use and small business premises is integrated with a verification loop for ascertaining whether motion events were indeed true or false and the mobile application is used for setup and notifications.

Experimental results from laboratory and residential deployments underscore the system's reliability, with detection accuracy approaching 98% and a typical response time of around four seconds. Although the level of false positives can be reduced by accurate calibration and sensor placement, the system performance is resilient even in varied environmental settings. Compared to standard CCTV systems and standard alarms, the automatic voice call feature is a standout feature, delivering important alerts in voice-based form that demands immediate user attention.

9. References

1. A. K. Srivastava, J. P. Sharma, and L. R. Jha, "Evolution of security systems: From traditional to smart solutions," *IEEE Access*, vol. 7, pp. 97830–97845, 2019.
2. S. R. Dhumal and P. N. Mahalle, "IoT-based system design for smart home applications," in *Proc. 3rd Int. Conf. Internet of Things (ICIoT)*, Chennai, India, 2019, pp. 45–52.
3. A. Z. Alkar, A. Karaca, and E. Halici, "An internet-based wireless home automation system for multifunctional devices," *IEEE Trans. Consum. Electron.*, vol. 51, no. 4, pp. 1169–1174, Nov. 2018.
4. A. Gupta, "Real-time responsiveness in embedded security systems," *Int. J. Embedded Syst.*, vol. 12, no. 2, pp. 95–105, 2020.
5. R. J. Garcia, *Physical Security Systems Handbook*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2017.
6. J. T. Smith and M. S. Krishnan, "Analyzing the limitations of CCTV-based surveillance," *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1407–1415, Mar. 2019.
7. H. M. Lee, D. H. Park, and K. H. Lee, "Enhancing human factor in CCTV monitoring: A cognitive approach," *IEEE Trans. Hum.-Mach. Syst.*, vol. 49, no. 4, pp. 310–320, Aug. 2019.