Webology, Volume 18, Number 5, 2021
ISSN: 1735-188X
DOI: 10.29121/WEB/V18I5/72

Integrating Low-Cost Devices into An Internet-Of-Things Security and Fire Prevention System

Durgaprasad Gangodkar[1], Devesh Pratap Singh[2], Dibyahash Bordoloi[3]

[1]Department of Computer Science & Engineering,Graphic Era Deemed to be University, Dehradun, Uttarakhand India, 248002
[2]Department of Computer Science & Engineering,Graphic Era Deemed to be University, Dehradun, Uttarakhand India, 248002
[3]Head of the Department, Department of Computer Science & Engineering,Graphic Era Hill University, Dehradun, Uttarakhand India, 248002

ABSTRACT

To warn of an intruder or suspicious object, modern electronic security systems may do much more than simply sound a loud, piercing beep. Additionally, they protect the owner and his belongings from harm by including a plethora of other useful features. Monitoring for fire, gas leak, flooding, assault, and medical emergencies are examples of the essential capabilities that fall under this category. Alternatively, more sophisticated mechanisms like a network of motion detectors, changes in the room's pressure and temperature, a camera system with facial and fingerprint recognition, a retinal scan, etc. Each potentially life-threatening situation is evaluated, and the results are communicated to the user application by text message or the Internet. The essay lays out a complex system for safeguarding people and their possessions using innovative technical solutions. In this paper, we go over the process of implementing a web interface for system control and management, as well as setting up a safe and fireproof system using the microcontroller Arduino.

Keywords:

INTRODUCTION

The Internet of Things security devices have started to coalesce into a complex network in which each node is in constant dialogue with the other nodes and with the external peripherals that provide backup and reliability. These systems have been redesigned so that the user can control and monitor a network of components from his desk at home using his smartphone. The home's lighting, temperature, and blinds, as well as the heating of the pool, are all integrated into the system, as are the sensors and smart locks. Numerous microelectronic devices, such as Arduino and Raspberry PI, are also compatible with the Internet of Things. Using the Internet and wireless technology, the IoT creates a remote-control system environment that may be utilised to set up an automated home or workplace setting.

A wireless sensor network is a type of network made up of nodes that work together to track changes in physical parameters in real time. Multiple applications are possible for the data collected.

Webology, Volume 18, Number 5, 2021
ISSN: 1735-188X
DOI: 10.29121/WEB/V18I5/72

It is possible to incorporate hundreds of sensor nodes, collector nodes, and gate nodes into the network. A variety of sensors, microcontrollers, and transceivers are embedded within each knot. The knot conducts a scan, and the results are relayed through the gate and into the network. In this analogy, the knots represent the limited storage, processing speed, and power of the gadgets.

Internet-based communication networks are a crucial pillar of the IoT's underlying architecture, making it possible for devices to connect from anywhere in the world at any time.

The focus of this essay is on the current state of commercially accessible security technologies and the design of an all-encompassing system to safeguard people and their possessions with the help of cutting-edge, cost-effective innovations. The microcontroller capabilities of Arduino were put to use in the creation of a safety and security surveillance network. The system was then put through its paces in realistic circumstances, and its individual parts were examined using cutting-edge statistical methods**.**

LITERATURE SURVEY

With Bluetooth technology, ***Chung et al***. demonstrated a smart home prototype. Every sensor and device in the system is controlled by an Android-powered smartphone that does not have access to the internet. Despite this, the user has very limited remote control over the system, and the system costs a considerable amount of money.

Raja et al. built a simple security system with an Arduino Uno acting as the brains (CPU). Infrared (PIR) sensors were used to detect any unwelcome movement inside the protected object, and magnetic sensors were set up at the building's entryways and exits. When this is completed, the information may be analysed to see if any of the windows or doors were left ajar. The LM35 temperature sensor is used to check if the temperature has exceeded an established limit, providing an additional layer of protection against fire.

It was proposed by ***Gulve et al.*** to use an IoT-based intelligent monitoring and recording application to store and retrieve video footage (IoT). As a result of technological advancements, it is now possible to detect the presence of almost anyone nearby. OpenCV is a programme that can be used for pattern recognition and other similar tasks (for video monitoring). A GSM module is used to send messages over the 900 MHz network (for both the SMS and the email notice).

Kader et al. state that there are several options for protecting a smart home's access points. To determine whether or not a person is authorised to enter a secure area, a fingerprint scanner reads their fingerprint and compares it to a database of stored fingerprints; if the fingerprints match, the door is unlocked. An Arduino ATMega328P microcontroller has been used as the system's brain. Using an RFID reader for security and PIR sensors for motion detection, ***Santoso et al***. have developed a smart house system that is controlled by an Arduino microcontroller. Wi-Fi is used for connection to the control unit, and the control unit itself is operated via a web app with security measures in place. The control unit Arduino ATMega 2560 was used in Prasetyo et alintelligent .'s office system. With the help of the camera attached to the Raspberry Pi, the suggested system is able to recognise the threat in the event of a violent incursion into the restricted area using potentially

Webology, Volume 18, Number 5, 2021
ISSN: 1735-188X
DOI: 10.29121/WEB/V18I5/72

harmful things made of metal, shattering the glass, starting a fire, or getting unauthorised access to a person. To ensure the identity of the visitor, a keypad or RFID reader may be utilised.

PROPOSED METHODOLOGY

The whole alarm security and distress system, including hardware, software, and housing. The device was constructed using the Arduino platform and its component parts. The shield is fashioned with the help of modelling tools. Solidworks and PrusaControl are only two examples. Schematics were created using draw.io, an online diagramming tool, and plans were drawn up in Friting, a programme. The system's software consisted of Arduino code as well as website code. We used the cross-platform Arduino IDE (Integrated Development Environment), which is written in Java. C and C++ were used in the creation of the Arduino software. The IDE's function library made it easier to programme routine hardware tasks. There were only two user-defined functions needed to make the app completely functioning.

The ultimate security solution involved three parts. During development, the hardware for each part of the system needed to be tackled separately. The Arduino boards and shields used in this project were wired together in the ways detailed in the preceding chapters. An advanced solution architecture was designed to contain all the necessary components after their appropriate operation was confirmed. The plan called for the use of a microSD card, an RFID reader, a keyboard, an LCD screen, an Arduino Mega board, and an Ethernet shield.

The application required a library to be set up for each component to guarantee appropriate operation and communication with the motherboard. Data loading, user authentication, sensor state monitoring, and data logging were all planned for an earlier version of the software. The database was created on the website examine.sk, and it had charts recording the sensors and security status, the users, and any new disruptions or alarms. Notepad++ was used to programme the PHP code used by the forms to store data in the database. A PHP file is in constant communication with Arduino, and any time a sensor value changes or an alert is created, the database is synchronised.

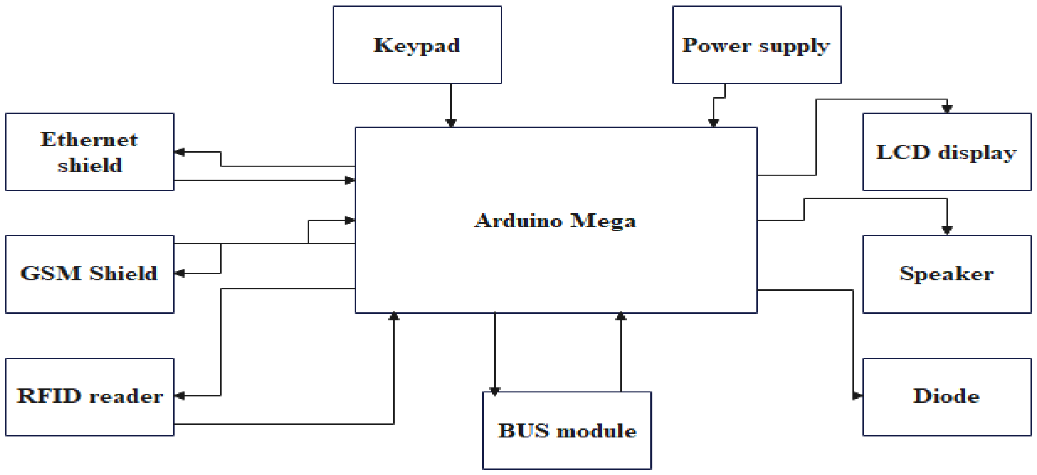

Fig 1: Proposed Architecture

Webology, Volume 18, Number 5, 2021
ISSN: 1735-188X
DOI: 10.29121/WEB/V18I5/72

After that, we moved on to creating the system's online interface so that users can enter data and check on its progress. This project was completed using Notepad++. We started by building the foundation of HTML and CSS. The design process also saw the production of the visual environment used to represent each sensor. The most recent five entries appeared in a new side panel on their own.

There were tabs labelled Control and Statement in the top right corner of the UI. There was a panel labelled "Control" where one could activate or deactivate various safety features, such as alarms and sprinklers. From the Statement tab of the interface, users could also view a full text statement of the data stored in the database. The logs were updated once an hour whenever there was a change, as notified by Arduino. Forms written in PHP that exchanged information with the server's back end were used to enter all the information and components into the design. The next critical step was to build a user login system within the website's front end. The user interface has been modified so that a login name and password entered into the system are now necessary to access any data. The password was protected by using a hash function, which is a way to convert data strings into easily readable output strings.

There were many moving parts and channels of communication in the created security system. Asynchronous or synchronous serial transmission was employed for some of the interactions. The components of the system were divided between the access panel and the switch board.

The control panel consisted of a 44-keypad LCD display, RFID reader, LED diode, and speaker. The Arduino Mega served as the system's nerve centre for inter-system communication. The LCD screen could communicate with the Arduino board via the I2C bus, which called for two data connections and two power wires. Connecting the screen to pins 20 and 21, we were able to use the LiquidCrystal I2C library to do the heavy lifting. There was just one bus that the real-time DS1307 modules used to communicate across. The character input from the keyboard was sent to an Arduino Mega board, which, with the aid of the Keypad library, was able to read it. Using the SS (Slave Select) and RST (Reset) pins, the RFID reader was linked to the board through a synchronous peripheral interface (SPI). The MFRC522 library's inclusion allowed it to read RFID tags and cards. Connecting diodes to digital pins 45 and 43 results in an LED. There was an Arduino Mega, an Ethernet Shield, a GSM Shield, a Real Time Module DS1306, a Bus Module, and a Power Supply all connected to digital pin 49 on the switch board, which was used to connect the speaker. Turning the module on and off, as well as establishing SPI bus communication between the Ethernet shield and the Arduino board, were accomplished by means of pins ICSP and SSpin 10. Thanks to the Ethernet library, we were able to connect to the requested server and send information to it. The LibrarySoftware Serial and asynchronous communication (pins 7 and 8) allowed the GSM shield to send and receive SMS messages. Connecting a single wire to pin 2 of the module's bus allowed us to read the temperature from its built-in sensors. Pins A11 and A13 received the sensor readings via two extra balanced current loops.

The control programme managed the safety and security apparatus. Both circuits are interconnected and built with the Arduino IDE. One component was the user's physical access panel, via which they controlled and managed the system. The second part of the control programme was a hub for information exchange between the sensors and the user interface, which operated invisibly in the background. It served double duty as an SMS warning system and a secure barrier. After inserting the RFID card and entering the correct password, the access panel's security mechanism would be deactivated, unlocking the panel's menu and functionality. Every circuit's menu structure, submenu, and user interface have been consolidated onto a single Arduino Mega 2560. Because of the Void protocols, all of the circuits could communicate with one another. Each action called a specific method, guaranteeing that the right code always ran. When the programme was first turned on, it loaded all of the necessary components for communicating with other devices. The system then adjusted its settings based on its analysis of the environmental circumstances under which each component was operating. There was a loading of sensor data and conditions. The system would periodically (every 30 seconds) check for fresh input from the user interface and update its database accordingly. After identifying whose input was disrupted, the algorithm would relay that information to the ID problem evaluated by the sending functions. ID was pre-set to receive an appropriate text message.

The security system could be managed and inspected by the user via an accessible internet interface. Web-based security system communication is constructed on top of a SQL database. All of the necessary information for the interface was retrieved from the database, which also collected information about each page visited. The website required a login before it could be accessed. An authorization form was submitted by the user and checked by the database. After the user's credentials were confirmed, they were able to access the page. On the homepage, users were able to submit requests for sensor and database reports. After some time spent fetching database information, the page was shown.

RESULTS AND SIMULATION RESULTS

The safety and fire prevention features of a home were evaluated. We checked the system's dependability, the frequency with which it turned on and off, and the accuracy with which it reported its current status. The 22-day trial ran with two separate safety circuits. The online interface was used to programme the security circuit to activate at 7 a.m. and deactivate at 3 p.m. The security system kept a log of the system's activity every hour during this time. Every hour of every day, the state of the fire system circuit was recorded. Sensor data has been entered into the database.

Users were informed of security system activity and disruptions via these statuses. The name of the activated circuit or sensor was displayed. Value then reported the on/off states of the circuit and the sensor.

The system's on/off status was only recorded when the power was actively toggled. The system would automatically save the current state in the event of a disruption. A data matrix was developed to ensure accurate processing. Data from an actual security system was cleared into the matrix.

If the system was turned on or off, it would be reflected in the Value field's data. It was in LogValue where the transformations between the binary values on (one) and off (zero) were recorded. As a fictitious backbone for the data structure, we posited the field Supposition. This metric reflected the typical operation of the system and, by extension, its predicted values. Log-Supposition, a binary

Webology, Volume 18, Number 5, 2021
ISSN: 1735-188X
DOI: 10.29121/WEB/V18I5/72

field, was formed after the field's values were computed. Next, we compared the observed binary field data with our model's predictions. If the data were consistent across time, a 1 was entered into the field Comparison, and a 0 was entered if they weren't. The determined field percentages are shown in Table 1.

Table 1. The calculation of conformity and discrepancy for the system

Values	**Number of records**	**Conformity**	**Discrepancy**
Numbers	754	750	4
Percentage	100.00%	99.47%	0.53%

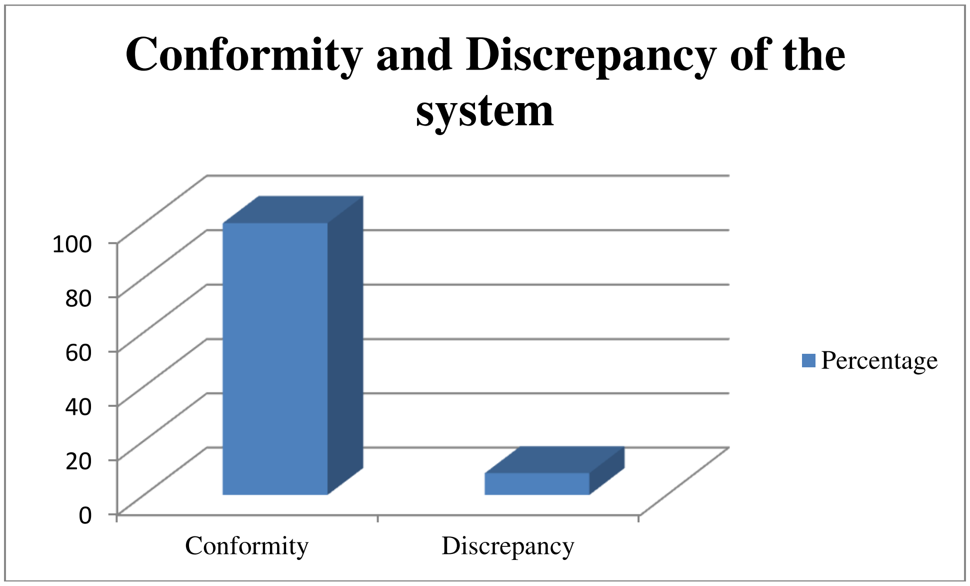

Fig 2:Comparing Discrepancy and conformity of the system

The total number of rows in the table was 754, which is 100% of the information. With 750 copies, 99.47% of the functional data was duplicated. Only four records out of a total of a million and fifty-three were not exact duplicates. The percentage of successful attempts for each security measure was determined and then compared. In an Excel data matrix, specific information was extracted from the Status field of the fire system. A field labelled "Comparison" then showed the results after being filtered.

Table 2. The calculation of conformity and discrepancy for the fire system

Values	**Number of records**	**Conformity**	**Discrepancy**
Numbers	518	516	2
Percentage	100.00%	99.61%	0.39%

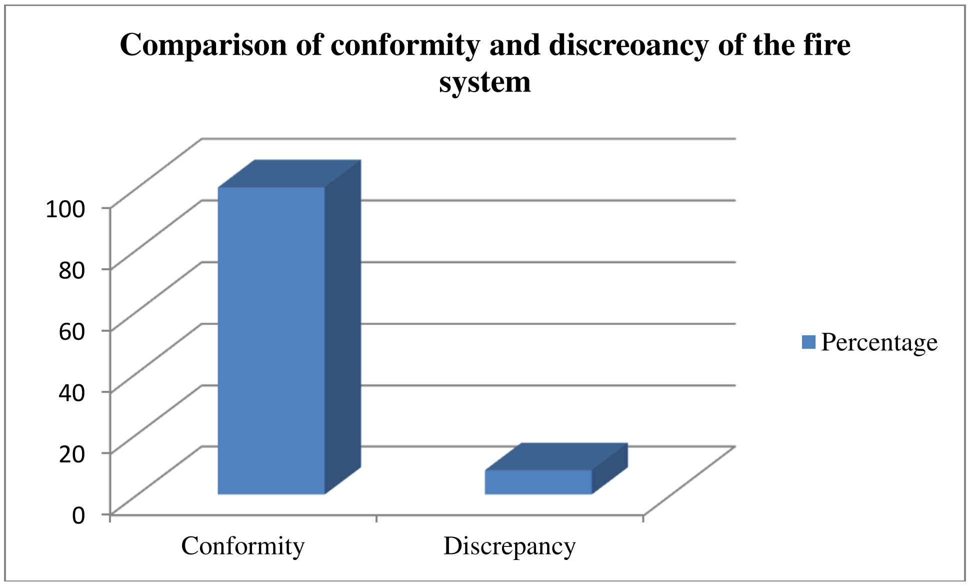

Fig 3: Comparing the Discrepancy and conformity of the fire system

Table 2 showed that there were 518 records for the fire department, which is 100%. As shown by these 516 documents, 99.61 percent of the data was shared. Only 0.39 percent of the data was new. As an example, the security system was employed. Table 3 details the results of applying a safety-system-specific filter to the data matrix.

Webology, Volume 18, Number 5, 2021
ISSN: 1735-188X
DOI: 10.29121/WEB/V18I5/72

Table 3. The calculation of conformity and discrepancy for the security system

Values	Number of records	Conformity	Discrepancy
Numbers	236	234	2
Percentage	100.00%	99.15%	0.85%

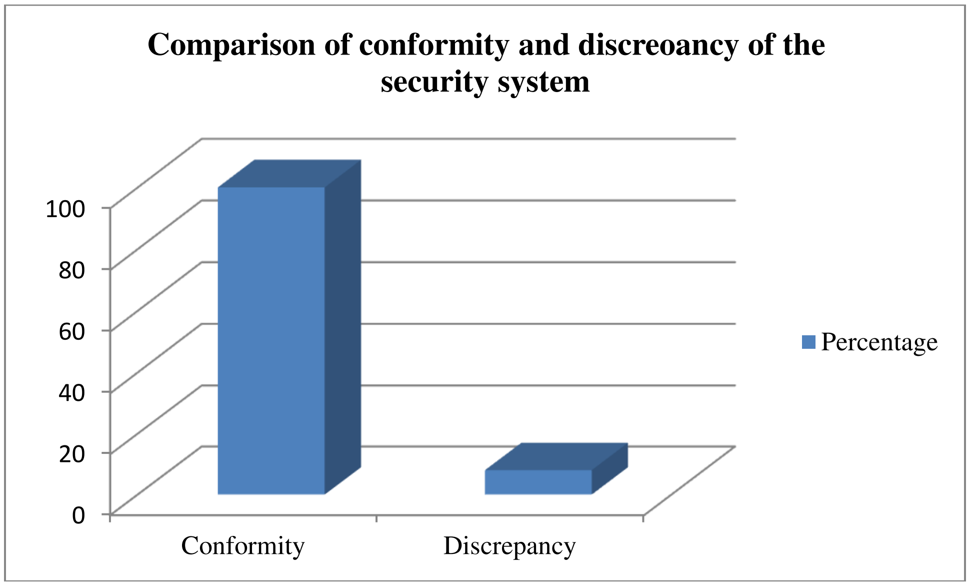

Fig 3: Comparing the Discrepancy and conformity of the security system

There were two inconsistencies that added up to 0.85% across the board across the security and fire systems. There were 234 records that were 100% similar, making up 99.15% of the total. There were a total of 236 records in the database. According to the findings of the investigation, the reliability of the security system in terms of both its function and its dependability over a period of time was 99%. This indicates that the system was activated at the time that had been programmed into it by the user.

CONCLUSION

The paper go through the steps of designing and building a security system with the help of the microcontroller Arduino. In the practical portion, the theoretical knowledge was put to use by creating an effective fire safety system. The concept called for a switchboard to connect the control room's physical buttons and web-based interface to remote access. The sensor module connected the two sets of sensors (one for fire safety, the other for security). A mobile Internet connection and the

Webology, Volume 18, Number 5, 2021
ISSN: 1735-188X
DOI: 10.29121/WEB/V18I5/72

SD card were needed to configure the system's alert notification. The security system's reliability was verified with the use of a data matrix that examined and rated the system's logs. Both the efficiency and error rate of the system were examined. The system's accomplishment at that test, which required only one failure out of a hundred, proved its dependability and effectiveness. When everything was said and done, we had an efficient and reliable automated security system. Its versatility as either a family house or an apartment was its strongest selling feature. Due to the user's ability to administer the system on their own, expert intervention is unnecessary.

REFERENCES

1. Chung, C.-C., Huang, C.Y., Wang, S.-C., Lin, C.-M.: Bluetooth-based Android interactive applications for smart living. In: 2011 Second International Conference on Innovations in Bio-inspired Computing and Applications, pp. 309–312. IEEE (2011)
2. Raja, K.B., Saranya, M., Shahana, R., Sreeraj, S., Vishnu, R.: Assorted security system to defend intrusion. Int. J. Adv. Res. Sci. Eng. (IJARSE) 7(2), 762–774 (2018)
3. Gulve, S.P., Khoje, S.A., Pardeshi, P.: Implementation of IoT-based smart video surveillance system. In: Behera, H.S., Mohapatra, D.P. (eds.) Computational Intelligence in Data Mining. AISC, vol. 556, pp. 771–780. Springer, Singapore (2017). https://doi.org/10.1007/978-981-10-3874-7_73
4. Kader, M., Haider, M.Y., Karim, M.R., Islam, M.S., Uddin, M.M.: Design and implementation of a digital calling bell with door lock security system using *fi*ngerprint. In: 2016 International Conference on Innovations in Science, Engineering and Technology (ICISET), pp. 1–5. IEEE (2016)
5. Santoso, F.K., Vun, N.C.H.: Securing IoT for smart home system. In: 2015 IEEE International Symposium on Consumer Electronics (2015)
6. Prasetyo, T., Zaliluddin, D., Iqbal, M.: Prototype of smart of*fi*ce system using based security system. J. Phys.: Conf. Ser. 1, 012189 (2018)