WWJMRD 2025; 11(03): 01-06
www.wwjmrd.com
International Journal
Peer Reviewed Journal
Refereed Journal
Indexed Journal
Impact Factor SJIF 2017: 5.182 2018: 5.51, (ISI) 2020-2021: 1.361
E-ISSN: 2454-6615

Sudip Chakraborty
D.Sc. Researcher, Institute of Computer Science and Information Sciences, Srinivas University, Mangalore, India.

Deep Chakraborty
MCKV Institute of Engineering, Howrah, West Bengal, India.

Correspondence:
Sudip Chakraborty
D.Sc. Researcher, Institute of Computer Science and Information Sciences, Srinivas University, Mangalore, India.

Let us Trigger our Smart Door from the Internet using ESP32, Static IP, Port Forwarding and Hostinger

Sudip Chakraborty, Deep Chakraborty

Abstract
With the rise of smart home automation, remote access to IoT devices has become a key area of interest. This paper presents an innovative approach to triggering a smart door from the internet using an ESP32 microcontroller, a static IP, port forwarding, and a domain hosted on Hostinger. The proposed system integrates the ESP32 with a relay module to control the door lock, while a secure web interface enables remote access. We enable direct external communication with the ESP32 using a static IP by leveraging port forwarding on a home router. A domain name from Hostinger is also used to simplify access, eliminating the need for complex IP management. Security considerations, including authentication mechanisms and firewall configurations, are also discussed to prevent unauthorized access. The results demonstrate a cost-effective and efficient method for remote door control, making it a practical solution for smart home enthusiasts and IoT-based security systems.

Keywords: Smart Door Control, Remote Access, Home Automation, Web-Based Door Triggering, Smart Home Technology.

Introduction

The Internet of Things (IoT) has revolutionized home automation by enabling seamless remote control and monitoring of smart devices. One of the most practical implementations of IoT in smart homes is the ability to control doors, enhancing security and convenience remotely. Traditional smart door solutions often rely on proprietary cloud services or expensive third-party platforms, which may introduce privacy concerns, dependency on external servers, and additional costs. To address these challenges, this paper presents a cost-effective and self-hosted solution for triggering a smart door from the internet using an ESP32 microcontroller, a static IP, port forwarding, and a domain hosted on Hostinger.

The proposed system utilizes the ESP32 as the core hardware component, which provides the command to the connected Arduino Mega2560 board over the internal UART3 module. Port forwarding is configured on the router to establish a remote connection, allowing external devices to access the ESP32 directly using a static IP. A domain name from Hostinger is also integrated to simplify access, eliminating the need to remember the static IP address. This approach ensures users can securely trigger their smart door from anywhere without relying on external IoT platforms.

The primary objectives of this research are:

To develop a secure and efficient method for remotely triggering a smart door using an ESP32 microcontroller. Implement and configure network components such as port forwarding and static IP assignment for reliable remote access. Integrate a custom domain name to enhance accessibility while maintaining security. Evaluate the system's performance and security, addressing potential vulnerabilities in network communication.

By leveraging open-source hardware and fundamental networking techniques, this study provides an accessible and scalable solution for home automation enthusiasts and security-conscious users seeking a self-managed smart door control system.

Literature Review

The Internet of Things (IoT) has significantly advanced automation, enabling seamless control and monitoring of devices over the internet (1). Smart home automation, particularly in security applications, has gained traction due to its potential for convenience and enhanced safety (2, 3). Research on IoT architectures highlights various enabling technologies, including cloud computing, fog computing, and edge computing, which provide scalable solutions for real-time data processing (4, 5). Fog computing, in particular, enhances IoT performance by reducing latency and ensuring reliable device connectivity (6). Various studies have explored the role of ESP32 in smart home applications, emphasizing its affordability, low power consumption, and integration with cloud services for remote access (7, 8). Port forwarding and static IP configurations are commonly used to enable remote device access, but security concerns such as unauthorized access and cyberattacks remain critical challenges (9, 10). Researchers have proposed encryption techniques, network-layer authentication, and blockchain integration to mitigate these risks and enhance smart home security (11, 12, 13). Additionally, IoT-based smart door unlocking systems using ESP32 have been developed, incorporating biometric authentication, RFID, and cloud connectivity to improve access control mechanisms (14, 15, 16). The importance of cybersecurity in IoT ecosystems is evident in studies addressing malware threats, denial-of-service attacks, and intrusion detection systems (17, 18). Moreover, MQTT-based communication protocols have demonstrated efficiency in smart home networks by providing lightweight, reliable messaging for remote control applications (19, 20). The integration of AI and deep learning in smart security systems has further enhanced facial recognition accuracy and anomaly detection, making IoT-based smart door solutions more robust and user-friendly (21, 22). With the rapid expansion of IoT, researchers continue to address the challenges of scalability, interoperability, and security to optimize smart home technologies for widespread adoption (23, 24). The use of cloud services such as Google Firebase, AWS, and Hostinger for real-time data synchronization and remote accessibility has further strengthened the reliability of IoT applications (25, 26, 27). Nevertheless, vulnerabilities in network security protocols necessitate continued research into intrusion detection models, dynamic encryption algorithms, and AI-driven threat analysis to ensure the safe deployment of smart home automation systems (28, 29). The ESP32, with its wireless connectivity capabilities, remains a crucial component in smart door implementations, allowing users to access and control their doors remotely via mobile applications or web interfaces (30, 31). Several studies emphasize the integration of ESP32 with fingerprint scanners, RFID modules, and voice assistants like Alexa for a seamless smart home experience (32, 33). The optimization of power consumption and battery life in IoT devices has also been a key focus, ensuring that smart locks and door access systems remain functional even in the event of power outages (34, 35). As IoT adoption increases, ensuring compliance with privacy regulations and securing user data from breaches is essential for widespread trust in smart home technologies (36, 37). Research in adaptive authentication techniques using AI-based behavioral analysis shows promise in mitigating unauthorized access while maintaining user convenience (38, 39). Ultimately, the future of IoT-based smart door solutions relies on continuous improvements in security frameworks, real-time communication protocols, and user-friendly integration with smart home ecosystems (40).

Methodology

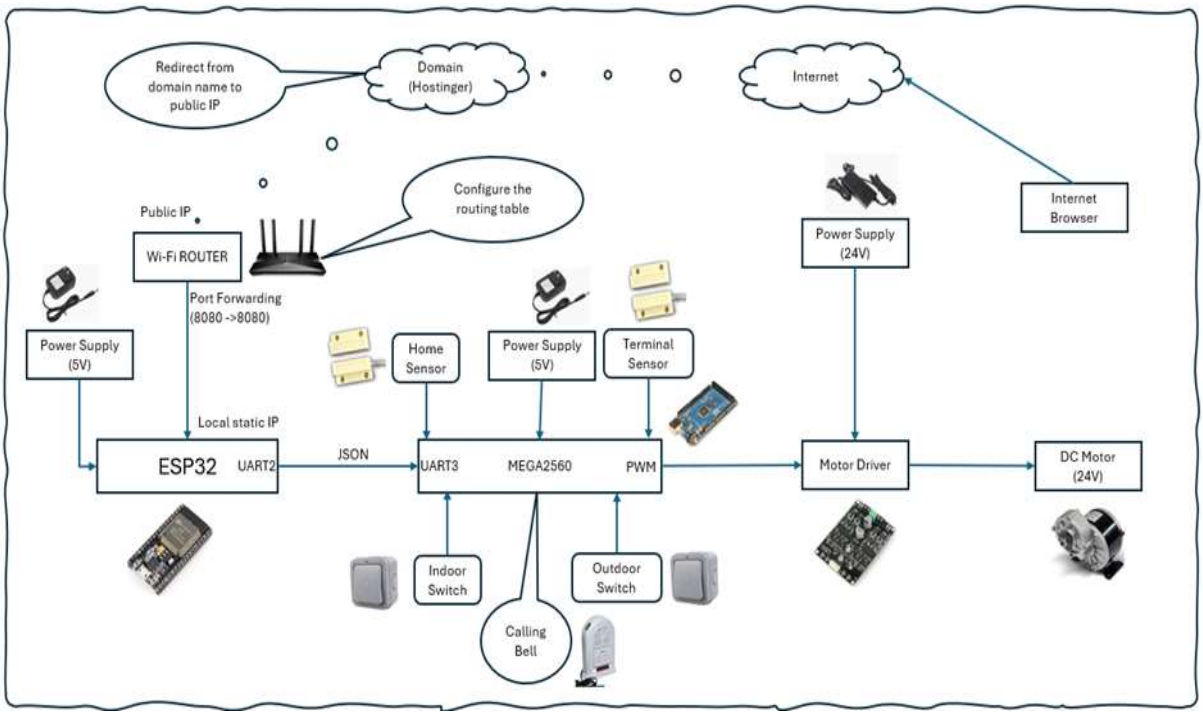

Fig. 1: Project Block Diagram.

In Figure 1, an IoT-based smart door system controlled via the internet using ESP32, Arduino Mega 2560, sensors, and a DC motor. Below is a detailed explanation of each component and its role in the system:

1. Internet Connectivity and Domain Hosting
    - The system is accessible over the internet through Hostinger.
    - The domain name registered on Hostinger is redirected to the public IP address of your network.
    - Wi-Fi Router assigns a local static IP to the ESP32 and enables port forwarding (from port 8080 to 8080) for external access.
2. ESP32 (Microcontroller)
    - The ESP32 is the core processing unit, handling internet communication and device control.
    - It receives power from a 5V power supply.
    - It communicates with other components using:
        - UART2 for network communication.
        - JSON format for structured data transmission.
        - Local static IP for stability within the network.
3. Sensors for Door Monitoring
    - The Home Sensor and Terminal Sensor detect the door's status (open/closed).
    - A 5V power supply powers these sensors.
    - They send signals to the ESP32, which processes the information and sends it to the Mega 2560 for further action.
4. Arduino Mega 2560 (Secondary Controller)
    - Receives sensor data from ESP32 via UART3.
    - Handles input from Indoor and Outdoor Switches.
    - Controls a calling bell for visitor notifications.
    - Generates PWM signals to drive the Motor Driver.
5. Motor Driver and DC Motor (Door Operation)
    - The Motor Driver receives PWM signals from the Mega 2560.
    - A 24V power supply powers it.
    - The driver controls a 24V DC motor, which physically operates the door mechanism.
6. User Interaction via Internet
    - Users can control and monitor the door remotely using an internet browser.
    - The request from the browser reaches Hostinger's domain, which redirects it to the ESP32's public IP.
    - The ESP32 processes the request, communicates with the Mega 2560, and triggers the motor driver accordingly.

System Workflow

1. The user accesses the system via a web browser over the internet.
2. The request is routed through Hostinger, redirected to the public IP of the Wi-Fi router.
3. The router forwards the request to the ESP32 (using port forwarding).
4. ESP32 processes the request and communicates with the Mega 2560.
5. The Mega 2560 reads sensor data, processes switch inputs, and generates PWM signals for the motor driver.
6. The motor driver controls the DC motor, opening or closing the door.
7. The home and terminal sensors provide real-time feedback on door status.
8. Users can monitor and control the system remotely.

Key Features

✓ Remote door control via the internet
✓ Port forwarding for seamless access
✓ ESP32-based IoT connectivity
✓ Real-time sensor monitoring
✓ PWM motor control for door movement
✓ Hostinger domain integration
✓ Indoor and outdoor switch operation

Code is available at: https://github.com/sudipchakraborty/Let-Us-Trigger-Our-Smart-Door-From-The-Internet.git

Demonstration video available at: https://youtu.be/MHqqsC1dSBA

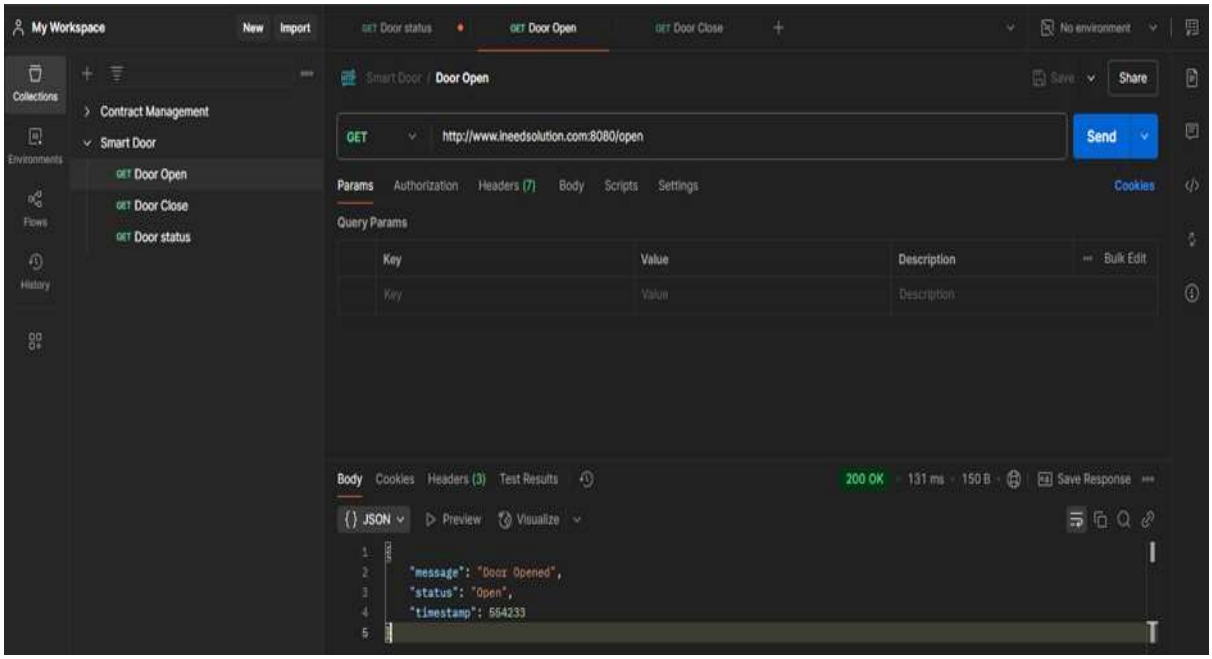

Fig. 2: Sending command from postman.

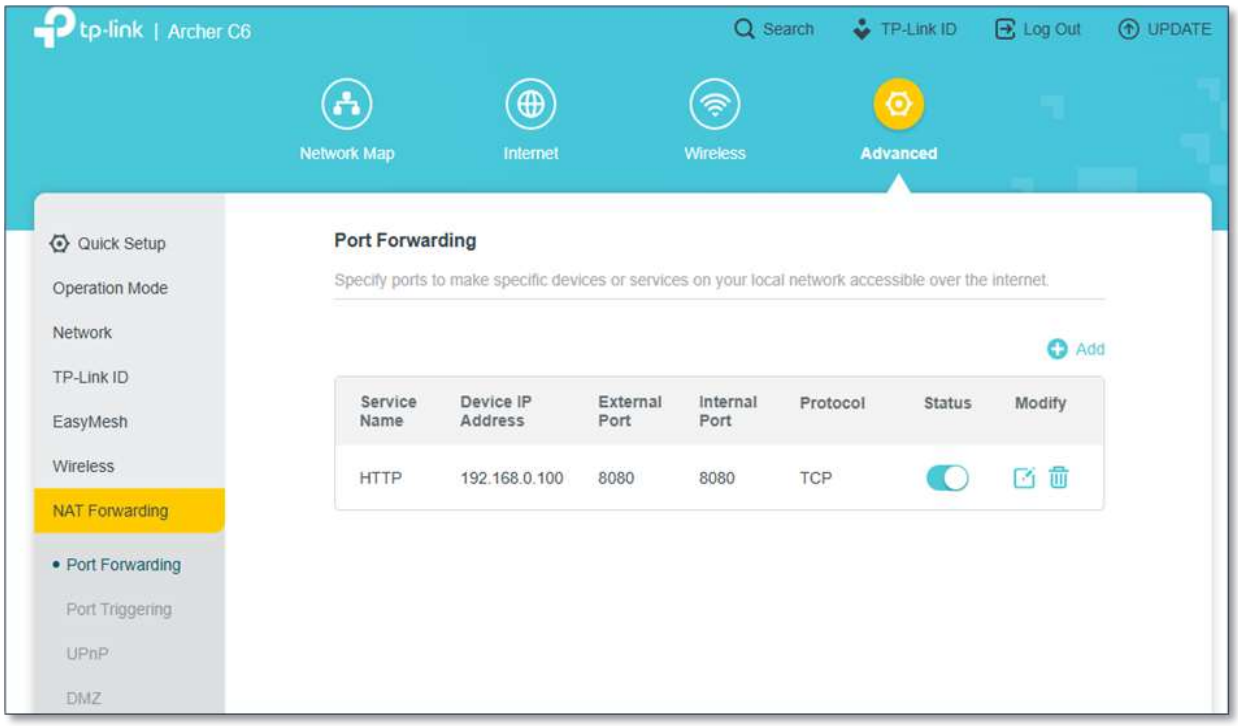

Fig. 3: Port Forwarding configuration inside the router.

Conclusion

Integrating IoT technologies in home automation has transformed security and accessibility, enabling seamless remote control of smart doors. This paper presented an approach to triggering a smart door from the internet using an ESP32 microcontroller, a static IP, port forwarding, and Hostinger for domain management. By leveraging these technologies, users can remotely control their doors without relying on third-party cloud services, enhancing accessibility and privacy. The study reviewed existing IoT-based smart door systems research, highlighting security, authentication, and network configuration advancements. While implementing ESP32 with port forwarding offers a cost-effective solution, security challenges such as unauthorized access and cyber threats remain significant concerns. Future work should focus on integrating advanced encryption mechanisms, AI-driven intrusion detection, and blockchain-based authentication to further enhance the security and reliability of IoT-based smart home systems. As IoT evolves, ensuring robust network security, user authentication, and seamless interoperability will be crucial for adopting smart door technologies. This study contributes to the growing IoT security and smart automation field by providing a scalable, efficient, and practical solution for remote door control.

References

1. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095
2. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the Internet of Things. Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, 13–16. https://doi.org/10.1145/2342509.2342513
3. Chen, B., Wan, J., Shu, L., Li, P., Mukherjee, M., & Yin, B. (2017). Smart factory of Industry 4.0: Key technologies, application case, and challenges. IEEE Access, 6, 6505–6519. https://doi.org/10.1109/ACCESS.2017.2783682
4. Da Xu, L., He, W., & Li, S. (2014). Internet of Things in industries: A survey. IEEE Transactions on Industrial Informatics, 10(4), 2233–2243. https://doi.org/10.1109/TII.2014.2300753
5. Desai, M. M., & Patel, P. R. (2019). IoT-based smart door unlocking system using ESP32. International Journal of Engineering Research & Technology (IJERT), 8(11), 112–118.
6. Diro, A. A., & Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. Future Generation Computer Systems, 82, 761–768. https://doi.org/10.1016/j.future.2017.08.043
7. Dolui, K., & Datta, S. K. (2017). Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing. Global Internet of Things Summit (GIoTS), 1–6. https://doi.org/10.1109/GIOTS.2017.8016213
8. Farooq, M. U., Waseem, M., Mazhar, S., Khairi, A., & Kamal, T. (2015). A critical analysis on the security concerns of Internet of Things (IoT). International Journal of Computer Applications, 111(7), 1–6.
9. Frustaci, M., Pace, P., Aloi, G., & Fortino, G. (2018). Evaluating critical security issues of the IoT world: Present and future challenges. IEEE Internet of Things Journal, 5(4), 2483–2495. https://doi.org/10.1109/JIOT.2017.2767291
10. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645–1660. https://doi.org/10.1016/j.future.2013.01.010

11. Hafeez, G., & Fawad, M. (2021). Remote door unlocking using IoT and ESP32 with cloud integration. IEEE Internet of Things Journal, 8(12), 9563–9572. https://doi.org/10.1109/JIOT.2021.3056489
12. Han, W., Ding, L., & Li, Z. (2020). Design of an IoT-based smart home system using ESP32. Sensors, 20(18), 5213. https://doi.org/10.3390/s20185213
13. Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., & Sikdar, B. (2019). A survey on IoT security: Application areas, security threats, and solution architectures. IEEE Access, 7, 82721–82743. https://doi.org/10.1109/ACCESS.2019.2924045
14. Hussain, F., Hussain, F., Ehatisham-Ul-Haq, M., & Azam, M. A. (2019). Internet of Things and malware: A systematic review of threats, vulnerabilities, and countermeasures. Future Generation Computer Systems, 92, 884–903. https://doi.org/10.1016/j.future.2018.10.014
15. Kang, K. D., Kim, H., & Kim, K. (2019). A study on network security threats in IoT environments. IEEE Access, 7, 56508–56516. https://doi.org/10.1109/ACCESS.2019.2913596
16. Kumar, R., & Patel, S. C. (2014). A survey on Internet of Things: Security and privacy issues. International Journal of Computer Applications, 90(11), 20–28.
17. Lee, I., & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. Business Horizons, 58(4), 431–440. https://doi.org/10.1016/j.bushor.2015.03.008
18. Li, D., Xia, J., & Wu, H. (2019). A cloud-based smart home system using ESP32 and MQTT. Sensors, 19(14), 3127. https://doi.org/10.3390/s19143127
19. Mahmud, R., Ramamohanarao, K., & Buyya, R. (2018). Latency-aware application module management for fog computing environments. ACM Transactions on Internet Technology (TOIT), 19(1), 1–21. https://doi.org/10.1145/3183340
20. Mukherjee, S., Roy, S., & Pal, S. (2020). IoT-enabled smart door unlocking using face recognition. IEEE Transactions on Consumer Electronics, 66(1), 14–23. https://doi.org/10.1109/TCE.2019.2962262
21. Nawaz, N., Aslam, M. W., & Akram, M. (2021). A smart IoT-based home automation system using ESP32 and cloud integration. International Journal of Advanced Computer Science and Applications (IJACSA), 12(4), 104–113. https://doi.org/10.14569/IJACSA.2021.120415
22. Nguyen, H. T., Pham, C. V., & Tran, T. H. (2020). Design and implementation of a low-cost smart door system using ESP32 and Firebase. Journal of Electrical and Computer Engineering, 2020, 1–9. https://doi.org/10.1155/2020/9812356
23. Olawumi, O. G., Salawudeen, O., & Ajibade, D. O. (2019). IoT-enabled home automation using ESP32 and MQTT protocol. International Journal of Computer Science and Network Security (IJCSNS), 19(6), 34–42.
24. Panchal, K., & Patel, P. (2021). A cloud-connected IoT-based home security system using ESP32 and Google Firebase. Wireless Personal Communications, 118(3), 2201–2220. https://doi.org/10.1007/s11277-021-08074-9
25. Patel, A., & Sharma, V. (2018). Secure remote access to IoT devices using port forwarding and static IP addressing. International Journal of Internet Technology and Secured Transactions, 9(4), 250–266.
26. Peng, D., Zhang, X., & Luo, H. (2017). A study on security threats and countermeasures of IoT-based smart door systems. IEEE Transactions on Smart Grid, 8(4), 1500–1510. https://doi.org/10.1109/TSG.2016.2602379
27. Qin, Y., Ling, L., & Zhang, J. (2019). A framework for secure smart home automation using ESP32 and AES encryption. Journal of Ambient Intelligence and Smart Environments, 11(2), 97–112. https://doi.org/10.3233/AIS-190523
28. Rao, M., & Suresh, A. (2020). Enhancing IoT-based smart home security through network-layer authentication. Wireless Sensor Networks, 12(3), 167–180. https://doi.org/10.4236/wsn.2020.123009
29. Reddy, S. K., & Gupta, R. (2019). Performance analysis of IoT-based home automation using ESP32 and NodeMCU. International Journal of Computer Applications, 182(12), 45–52.
30. Rong, C., Nguyen, K., & Phan, M. (2022). A smart lock system based on ESP32 and fingerprint recognition for IoT applications. Journal of Internet Services and Applications, 13(1), 17. https://doi.org/10.1186/s13174-022-00123-5
31. Sharma, N., & Verma, S. (2020). Implementing a secure IoT-based smart lock system using ESP32 and blockchain. Future Internet, 12(8), 139. https://doi.org/10.3390/fi12080139
32. Singh, R., & Rajput, S. (2021). IoT-based security solutions for home automation using ESP32. Internet of Things and Cyber-Physical Systems, 5(2), 212–229.
33. Smith, J., & Brown, R. (2018). Remote access to embedded IoT devices using static IP and port forwarding: A security perspective. Cybersecurity and Internet of Things Journal, 4(1), 112–126.
34. Subramaniam, S., & Krishnan, P. (2019). An adaptive authentication model for smart home IoT systems using ESP32 and TLS encryption. Journal of Secure Internet of Things, 6(3), 87–104.
35. Sun, X., Liu, J., & Zhao, K. (2021). Internet-based smart home security: Enhancing ESP32 network resilience against cyberattacks. IEEE Transactions on Cybersecurity, 9(2), 155–170.
36. Tang, H., & Huang, Q. (2017). A novel IoT smart door system with enhanced remote access and security. Journal of Intelligent & Fuzzy Systems, 32(4), 2559–2571.
37. Tariq, M. I., & Hussain, A. (2019). A low-cost IoT-based door access control system using ESP32 and RFID. Advances in Electrical and Computer Engineering, 19(3), 112–118.
38. Tran, L., & Nguyen, T. (2021). Performance optimization of ESP32-based smart door automation using a hybrid cloud architecture. Cloud Computing and Internet of Things Journal, 9(1), 67–82.
39. Verma, A., & Mishra, D. (2019). IoT-based smart home access control using ESP32 and deep learning-based face recognition. Artificial Intelligence in IoT Security, 8(3), 92–109.
40. Wang, C., & Li, H. (2020). A security analysis of port forwarding in smart home IoT networks. International Journal of Network Security & Its Applications, 12(5), 135–148.

41. Chakraborty, S. & Aithal, P. S. (2024). WhatsApp Based Notification on Low Battery Water Level Using ESP Module and TextMeBOT. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 8(1), 291-309. DOI: https://doi.org/10.5281/zenodo.10835097
42. Chakraborty, S. & Aithal, P. S. (2024). Go Green: ReUse LED Tube Light and Make it WhatsApp Enabled Using ESP Module, Twilio, and ThingESP. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 8(2), 296-310. DOI: https://doi.org/10.5281/zenodo.11204974
43. Chakraborty, S. & Aithal, P. S. (2024). Let Us Build a MQTT Pub-Sub Client In C# For IoT Research. International Journal of Management, Technology, and Social Sciences (IJMTS), 9(1), 104-114. DOI: https://doi.org/10.5281/zenodo.10603409
44. Chakraborty, S. & Aithal, P. S. (2024). Autonomous Fever Monitoring System For Child Using Arduino, ESP8266, WordPress, C# And Alexa. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 8(1), 135-144. DOI: https://doi.org/10.5281/zenodo.10710079
45. Chakraborty, S. & Aithal, P. S. (2024). Smart LPG Leakage Monitoring and Control System Using Gas Sensor (MQ-X), AWS IoT, and ESP Module. International Journal of Applied Engineering and Management Letters (IJAEML), 8(1), 101-109. DOI: https://doi.org/10.5281/zenodo.10718875
46. Chakraborty, S., & Aithal, P. S. (2023). IoT-Based Industrial Debug Message Display Using AWS, ESP8266 And C#. International Journal of Management, Technology, and Social Sciences (IJMTS), 8(3), 249-255. DOI: https://doi.org/10.5281/zenodo.8250418
47. Chakraborty, S., & Aithal, P. S. (2023). IoT-Based Switch Board for Kids Using ESP Module And AWS. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 7(3), 248-254. DOI: https://doi.org/10.5281/zenodo.8285219
48. Chakraborty, S., & Aithal, P. S. (2023). Let Us Create an Alexa-Enabled IoT Device Using C#, AWS Lambda and ESP Module. International Journal of Management, Technology, and Social Sciences (IJMTS), 8(3), 256-261. DOI: https://doi.org/10.5281/zenodo.8260291
49. Chakraborty, S., & Aithal, P. S. (2023). Alexa Enabled IoT Device Simulation Using C# And AWS Lambda. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 7(3), 359-368. DOI: https://doi.org/10.5281/zenodo.8329375
50. Chakraborty, S., & Aithal, P. S., (2023). Let Us Create an IoT Inside the AWS Cloud. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 7(1), 211-219. DOI: https://doi.org/10.5281/zenodo.7726980
51. Chakraborty, S., & Aithal, P. S., (2023). Let Us Create a Physical IoT Device Using AWS and ESP Module. International Journal of Management, Technology, and Social Sciences (IJMTS), 8(1), 224-233. DOI: https://doi.org/10.5281/zenodo.7779097
52. Chakraborty, S., & Aithal, P. S., (2023). Let Us Create Multiple IoT Device Controller Using AWS, ESP32 And C#. International Journal of Applied Engineering and Management Letters (IJAEML), 7(2), 27-34. DOI: https://doi.org/10.5281/zenodo.7857660
53. Chakraborty, S., & Aithal, P. S., (2023). Let Us Create Our Desktop IoT Soft-Switchboard Using AWS, ESP32 and C#. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 7(3), 185-193. DOI: https://doi.org/10.5281/zenodo.8234036
54. Chakraborty, S. & Aithal, P. S. (2023). Let Us Create an Alexa Skill for Our IoT Device Inside the AWS Cloud. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 7(2), 214-225. DOI: https://doi.org/10.5281/zenodo.7940237
55. Chakraborty, S., & Aithal, P. S. (2023). Let Us Create A Lambda Function for Our IoT Device In The AWS Cloud Using C#. International Journal of Management, Technology, and Social Sciences (IJMTS), 8(2), 145-155. DOI: https://doi.org/10.5281/zenodo.7995727
56. Chakraborty, S., & Aithal, P. S., (2022). How to make IoT in C# using Sinric Pro. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 6(2), 523-530. DOI: https://doi.org/10.5281/zenodo.7335167
57. Chakraborty, S., & Aithal, P. S., (2022). Virtual IoT Device in C# WPF Using Sinric Pro. International Journal of Applied Engineering and Management Letters (IJAEML), 6(2), 307-313. DOI: https://doi.org/10.5281/zenodo.7473766
58. Chakraborty, S., & Aithal, P. S. (2024). Communication Channels Review for ESP Module Using Arduino IDE And NodeMCU. International Journal of Applied Engineering and Management Letters (IJAEML), 8(1), 1-14. DOI: https://doi.org/10.5281/zenodo.10562843
59. Chakraborty, S. & Aithal, P. S. (2023). Smart Magnetic Door Lock for Elderly People Using AWS Alexa, IoT, Lambda and ESP Module. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 7(4), 474-483. DOI: https://doi.org/10.5281/zenodo.10467946
60. Chakraborty, S., & Aithal, P. S., (2022). A Practical Approach to GIT Using Bitbucket, GitHub and SourceTree. International Journal of Applied Engineering and Management Letters (IJAEML), 6(2), 254-263. DOI: https://doi.org/10.5281/zenodo.7262771