International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 5, Sep-Oct -2026, ISSN (Online): 2395-566X

A Literature Survey on Iot-Based Embedded Systems for Monitoring and Control

Bhalerao Atharv Ashok[1], Gunjal Ayush Santosh[2], Gaikwad Darshan Dattatray[3]
Prof. P. S. Dumbre[4], Prof. M. S. Nalawade[5], Dr. V. M. Dhede[6]
Department Of Electronics And Telecommunication Engineering, Jaihind College Of Engineering, Pune, Maharashtra, India

Abstract—IoT-based embedded systems integrate sensors, microcontrollers, communication networks and software platforms to connect physical devices with digital applications. They sense the environment, process the readings locally, and transmit useful data to a monitoring platform; depending on the application, the controller can also drive an actuator automatically or act on a remote user command. This paper surveys the literature on such systems under six headings: IoT foundations and reference architectures, low-cost Wi-Fi/Bluetooth microcontrollers such as the ESP32, wireless and messaging protocols, cloud and edge platforms, application domains (smart homes, agriculture, healthcare, industry and environmental monitoring), and security. Twenty-two peer-reviewed works are compared in a summary table. The survey finds that low-cost hardware and lightweight protocols such as MQTT have made monitoring-and-control systems easy to build, but that power consumption, sensor calibration, reliable connectivity and device security remain weak points in most reported prototypes. Based on these gaps, a general ESP32-based monitoring-and-control system is proposed, with local threshold-based automation, MQTT/HTTP data transfer to a cloud dashboard, and design rules for reliability, low power and secure communication.

Keywords—Internet of Things (IoT), embedded systems, ESP32, sensors, MQTT, cloud computing, edge computing, smart home, smart agriculture, IoT security.

I. INTRODUCTION

The Internet of Things (IoT) extends the Internet from computers and phones to everyday physical objects that can sense, compute and communicate. Atzori et al. [1] described IoT as the convergence of three visions — "things-oriented" (sensors, RFID tags), "Internet-oriented" (addressing and networking) and "semantic-oriented" (representing and reasoning over data). Gubbi et al. [2] added a cloud-centric view in which sensor data are stored, analysed and visualised on remote servers, and Al-Fuqaha et al. [3] mapped the protocols and enabling technologies used at each layer of an IoT system.At the device level, every IoT node is an embedded system:a microcontroller that reads sensors, runs a fixed program, and drives outputs under tight limits on cost, memory and power. Adding a network interface turns this local controller into a node that can report readings to a dashboard and accept commands from a remote user. This combination — embedded sensing and control plus Internet connectivity — is what this paper calls an IoT-based embedded system.Such systems now appear in smart homes, farms, hospitals, factories and environmental monitoring stations. Because the field is broad and fast-moving, a structured review is needed before designing a new system. The objectives of this survey are:

- To review reference architectures and the main hardware, communication and platform choices for IoT-based embedded systems.
- To summarise how these systems have been applied in major domains and what results were reported.
- To identify open problems — power, reliability, sensor accuracy and security — that a new design should address.
- To propose a general ESP32-based monitoring-and-control system based on these findings.

The rest of the paper is organised as follows. Section II describes the general architecture. Section III reviews the literature by topic, and Section IV compares the surveyed works. Section V lists research gaps, Section VI presents the proposed system and methodology, and Section VII concludes the paper.

II. ARCHITECTURE OF IOT-BASED EMBEDDED SYSTEMS

A typical IoT-based embedded system has five functional blocks — sensing, processing, communication, platform and actuation — matching the layered models in [3]–[5]. Fig. 1 shows how they connect.

© 2026 IJSRET

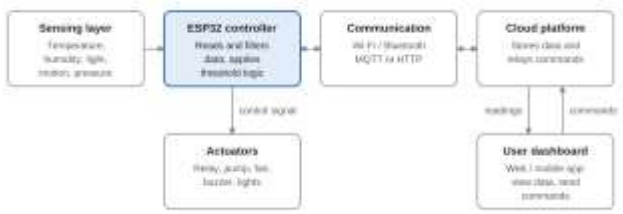


Fig. 1. General architecture of an IoT-based embedded monitoring-and-control system.

The **sensing layer** converts physical quantities such as temperature, humidity, light intensity, motion or pressure into electrical signals. The **processing unit**, here an ESP32 microcontroller, reads these signals through its ADC, I²C or digital inputs, filters them, and compares them with programmed thresholds. If a threshold is crossed, it drives an **actuator** — a relay, pump, fan, buzzer or light — without waiting for the cloud.

The **communication interface** sends selected readings over Wi-Fi or Bluetooth using a protocol such as MQTT or HTTP. The **cloud or application platform** stores the data, displays it on a web or mobile dashboard, and passes user commands back to the controller. Data therefore flow upward from sensors to the user, while commands flow downward from the user or the local logic to the actuators.

III. LITERATURE SURVEY

3.1 Iot Foundations And Reference Architectures

Atzori et al. [1] gave one of the earliest broad surveys of IoT. They grouped enabling technologies into identification (RFID), sensing (wireless sensor networks) and middleware, and listed open issues in standardisation, addressing, and security and privacy. Gubbi et al. [2] proposed a cloud-centric IoT in which sensor networks feed a cloud that provides storage, analytics and visualisation, and stressed that data presentation is as important as data collection.

Al-Fuqaha et al. [3] reviewed IoT protocols layer by layer, including application protocols (MQTT, CoAP, AMQP, XMPP, DDS) and network protocols (6LoWPAN, RPL). They also linked IoT with big-data analytics and fog computing. Lin et al. [4] surveyed IoT architecture together with fog/edge computing, and treated security and privacy as a cross-layer concern. Sethi and Sarangi [5] compared three-layer (perception, network, application) and five-layer architectures and described protocols and applications for each layer.

Together these works establish the layered model followed in this paper: sensing, processing, communication, platform and application.

3.2 Low-Cost Microcontrollers For Iot Nodes

The choice of controller decides cost, power and connectivity. Maier, Sharp and Vagapov [6] compared the ESP32 with the ESP8266, TI CC3200 and Digi XBee Wi-Fi modules. They noted the ESP32's dual-core 32-bit processor, built-in Wi-Fi and Bluetooth, and rich peripheral set at low cost, and demonstrated a portable wireless oscilloscope with a mobile app.

Babiuch, Foltýnek and Smutný [7] reported practical experience developing measurement and data-processing applications on several ESP32 boards, and compared development environments such as the Arduino IDE and Espressif's ESP-IDF. In a follow-up work, Foltýnek, Babiuch and Šuránek [8] split the workload across the ESP32's two cores — one for data processing and one for wireless communication — showing that software design can extract more performance from low-end hardware.

Aghenta and Iqbal [9] used an ESP32 board with the open-source Thinger.IO platform to build a low-cost SCADA system for remote monitoring and control. Their work shows that an ESP32 can serve as a remote terminal unit at a fraction of the cost of commercial SCADA hardware.

3.3 Communication And Messaging Protocols

Naik [10] evaluated four application-layer protocols — MQTT, CoAP, AMQP and HTTP — on message size, overhead, power use, bandwidth, latency, reliability, QoS, interoperability and security. The study concluded that no single protocol suits every IoT system: MQTT and CoAP are lighter for constrained devices, while HTTP is widely supported but heavier. CoAP itself is specified in IETF RFC 7252 [11] as a REST-style protocol over UDP for constrained nodes and networks.

For long-range, battery-powered nodes, low-power wide-area networks (LPWANs) are an alternative to Wi-Fi. Mekki et al. [12] compared Sigfox, LoRa and NB-IoT and found that Sigfox and LoRa lead on battery life, capacity and cost, while NB-IoT leads on latency and quality of service. Raza, Kulkarni and Sooriyabandara [13] reviewed the LPWAN design goals of long range, low power and low cost, and the techniques used to reach them.

3.4 Cloud And Edge Platforms

Cloud platforms provide storage, dashboards and remote access, but sending every reading to the cloud costs bandwidth and adds delay. Shi et al. [14] argued for edge computing, where data are processed close to where they are generated, to reduce latency and bandwidth, extend battery

© 2026 IJSRET

life and protect privacy. Lin et al. [4] similarly described fog nodes as an intermediate layer between devices and the cloud. For small monitoring systems, this suggests doing threshold checks and control decisions on the microcontroller itself and sending only summaries or alerts upward.

3.5 Application Domains

Smart homes. Piyare [15] built a low-cost home control and monitoring system using an embedded micro-web server with IP connectivity, controlled from an Android app, without a dedicated server PC. Kodali et al. [16] built a wireless home security system that alerts the owner over the Internet when an intrusion is detected and can sound an alarm, reusing the same sensors for home automation.

Agriculture and environment. Ojha, Misra and Raghuwanshi [17] reviewed wireless sensor networks in agriculture, covering irrigation, crop and soil monitoring and greenhouse control, and identified power, cost and field robustness as practical barriers. Talavera et al. [18] systematically reviewed IoT in agro-industrial and environmental fields, grouped applications into monitoring, control, logistics and prediction, and compiled the results into a reference architecture.

Healthcare. Islam et al. [19] surveyed IoT-based healthcare, including network architectures, applications such as remote vital-sign monitoring, and security requirements, threat models and attack taxonomies specific to medical data.

Industry. Xu, He and Li [20] reviewed IoT in industries, describing a service-oriented architecture, key enabling technologies and industrial applications, and noted challenges in standards, scalability and security.

3.6 Security And Privacy

Sicari et al. [21] reviewed security, privacy and trust in IoT, covering authentication, access control, data confidentiality, middleware security and policy enforcement. They concluded that many solutions existed only as proposals and that lightweight, standardised mechanisms were still lacking. Kolias et al. [22] analysed the Mirai botnet, which took over large numbers of IoT devices such as cameras and routers through default credentials and used them for large distributed denial-of-service attacks. This case shows that weak device-level security is a practical risk, not only a theoretical one.

IV. COMPARATIVE SUMMARY OF SURVEYED WORKS

Table I summarises the twenty-two works reviewed above. Most focus either on architecture and protocols or on a single application prototype; few combine low-cost hardware, local control, low power and security in one design.

TABLE I Summary of Surveyed Literature

Ref.	Authors (Year)	Focus	Technology / Method	Key Contribution	Limitation / Gap
[1]	Atzori et al. (2010)	IoT survey	RFID, WSN, middleware	Three IoT visions; enabling technologies	Early work; little on low-cost MCUs
[2]	Gubbi et al. (2013)	Cloud-centric IoT	WSN + cloud	Cloud storage, analytics, visualisation	Cloud dependence; latency not addressed
[3]	Al-Fuqaha et al. (2015)	Protocols survey	MQTT, CoAP, 6LoWPAN, RPL	Layer-wise protocol review	No hardware implementation
[4]	Lin et al. (2017)	Architecture, fog, security	Fog/edge + IoT	Fog layer; cross-layer security	Conceptual; no prototype
[5]	Sethi & Sarangi (2017)	Architectures, protocols	3- and 5-layer models	Clear layered models	Generic; no design guidance
[6]	Maier et al. (2017)	MCU comparison	ESP32 vs ESP8266, CC3200, XBee	ESP32 capability; wireless oscilloscope	Power use in field not studied
[7]	Babiuch et al. (2019)	ESP32 data processing	ESP32 boards, Arduino IDE, ESP-IDF	Practical development experience	Lab-scale only
[8]	Foltýnek et al. (2019)	Dual-core ESP32	Core-wise task split	Better use of low-end hardware	Single platform
[9]	Aghenta & Iqbal (2019)	Low-cost SCADA	ESP32 + Thinger.IO	Open-source remote monitoring and control	Limited security discussion

© 2026 IJSRET

Ref.	Authors (Year)	Focus	Technology / Method	Key Contribution	Limitation / Gap
[10]	Naik (2017)	Messaging protocols	MQTT, CoAP, AMQP, HTTP	Multi-criteria protocol comparison	Qualitative; no device tests
[11]	Shelby et al. (2014)	CoAP standard	REST over UDP	Protocol for constrained nodes	Standard, not an evaluation
[12]	Mekki et al. (2019)	LPWAN comparison	Sigfox, LoRa, NB-IoT	Technology fit per scenario	Low data rate only
[13]	Raza et al. (2017)	LPWAN overview	LPWAN design techniques	Long range, low power design goals	Network-level; no node design
[14]	Shi et al. (2016)	Edge computing	Edge processing	Lower latency and bandwidth	Vision paper
[15]	Piyare (2013)	Smart home	Micro-web server + Android	Remote control without server PC	Limited security
[16]	Kodali et al. (2016)	Home security and automation	Wi-Fi sensor node + Internet alerts	Intrusion alerts; shared sensors	Small-scale prototype
[17]	Ojha et al. (2015)	Agriculture WSN	WSN for irrigation, greenhouse	Field-practice review	Power and robustness issues
[18]	Talavera et al. (2017)	Agro-industrial and environmental IoT	Systematic review	Application classes; reference architecture	Interoperability, power open
[19]	Islam et al. (2015)	Healthcare IoT	Architectures, security models	Health-specific threat model	Survey; no device design
[20]	Xu et al. (2014)	Industrial IoT	Service-oriented architecture	Industrial applications	Standards and security open
[21]	Sicari et al. (2015)	Security, privacy, trust	Authentication, access control	Security requirement survey	Lightweight solutions lacking
[22]	Kolias et al. (2017)	IoT botnets	Mirai analysis	Real attack via default passwords	Mitigation not standardised

V. RESEARCH GAPS AND CHALLENGES

The survey points to six recurring gaps that a new IoT-based embedded system should address.

1. **Power consumption.** Wi-Fi nodes such as the ESP32 draw far more current while transmitting than while sleeping. Prototypes in [6], [7], [15] and [16] are mostly mains-powered and rarely report battery life, while LPWAN studies [12], [13] show that range and battery life must be traded against data rate.

2. **Reliability and connectivity loss.** Cloud-centric designs [2] stop working when the Internet link fails. Edge processing [4], [14] lets the controller keep making safe local decisions, but few application prototypes implement it.

3. **Sensor accuracy and calibration.** Low-cost sensors drift and are sensitive to noise and placement. Agricultural and environmental reviews [17], [18] identify field robustness as a practical barrier, yet calibration is seldom described in prototype papers.

4. **Protocol choice.** No single protocol suits every system [10]. Many student and prototype systems use plain HTTP even where MQTT or CoAP [11] would lower overhead.

5. **Security.** Default passwords, unencrypted links and missing firmware updates remain common. The Mirai case [22] shows the consequences, and surveys [19], [21] note that lightweight, standard security for small devices is still lacking.

6. **Interoperability and scalability.** Proprietary platforms and ad-hoc data formats make it hard to extend a single prototype to many nodes or to move between cloud services [18], [20].

VI. PROPOSED SYSTEM AND METHODOLOGY

Based on the gaps above, a general ESP32-based monitoring-and-control system is proposed. It follows the layered

© 2026 IJSRET

architecture of Section II and can be adapted to smart-home, agriculture, healthcare, industrial or environmental use by changing the sensors and actuators.

6.1 System Description

- Sensing: sensors for temperature, humidity, light intensity, motion or pressure, chosen per application.
- Processing: an ESP32 reads and filters sensor values, compares them with thresholds, and decides on local actions. Following [8], one core can handle sensing and control while the other handles Wi-Fi communication.
- Communication: Wi-Fi with MQTT for periodic readings and alerts, and HTTP for configuration where needed [10]; Bluetooth for local setup.

4. Platform: a cloud or web/mobile dashboard that stores data, plots trends and sends user commands back to the node [2], [9].

5. Actuation: relays, fans, pumps, buzzers or lights driven automatically by thresholds or manually by the user.

6.2 Proposed System Architecture

Fig. 2 arranges these blocks into the four-layer architecture used in this work, adapted from the layered models in [3]–[5].

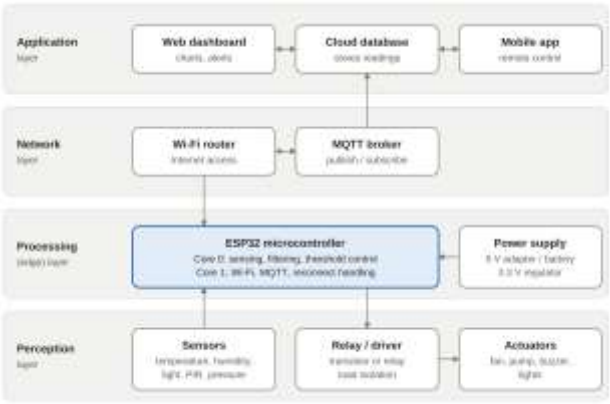


Fig. 2. Layered architecture of the proposed ESP32-based monitoring-and-control system.

- Perception layer: sensors measure the physical parameters; a relay or transistor driver isolates high-power actuators from the controller.
- Processing (edge) layer: the ESP32 runs sensing, filtering and threshold control on one core and Wi-Fi/MQTT communication on the other [8], so local control continues even if the network fails [14]. A regulated 3.3 V supply powers the board.
- Network layer: the ESP32 connects through a Wi-Fi router to an MQTT broker, which uses a publish/subscribe model to pass readings up and commands down [10].
- Application layer: a cloud database stores readings, while the web dashboard and mobile app display data, raise alerts and send user commands back to the node.

6.3 Design Methodology

- **Requirement analysis:** define the parameters to monitor, the sampling rate, the actions to automate and the expected power source.
- **Component selection:** choose sensors by range and accuracy, the ESP32 as controller [6], and actuators with suitable driver circuits.
- **Hardware design:** interface sensors over ADC, I²C or digital pins; isolate high-power loads through relays or transistor drivers; add protection and a stable power supply.
- **Firmware development:** read and filter sensors, apply threshold logic locally (edge processing [14]), and publish data over MQTT with reconnect handling so local control continues if the network fails.
- **Platform and dashboard:** configure the cloud/broker, design the dashboard, and add remote command handling.
- **Security:** unique credentials instead of defaults, TLS-encrypted MQTT/HTTPS where possible, and authenticated user access [21], [22].
- **Power optimisation:** use ESP32 sleep modes and reduce transmission frequency for battery-powered nodes.
- **Testing and calibration:** calibrate sensors against reference instruments, test response time, data accuracy and behaviour during network loss.

6.4 Expected Advantages And Limitations

The proposed design offers real-time monitoring, remote access and automatic control at low cost, and continues safe local operation during network outages. Its limitations are dependence on Wi-Fi coverage for remote access, higher power draw than LPWAN-based nodes [12], and the need for periodic sensor calibration and firmware updates.

VII. CONCLUSION

This paper reviewed twenty-two works on IoT-based embedded systems, covering architectures, low-cost microcontrollers, communication protocols, cloud and edge platforms, application domains and security. The literature

© 2026 IJSRET

shows that devices such as the ESP32 and lightweight protocols such as MQTT have made real-time monitoring and remote control affordable and easy to prototype. It also shows that most reported systems give limited attention to power consumption, behaviour during network failure, sensor calibration and device security. The proposed ESP32-based monitoring-and-control system addresses these gaps through local threshold-based control, MQTT communication with reconnect handling, sleep-mode power management and basic security measures. Future work will implement the system for a chosen application, measure its accuracy, response time and power consumption, and compare the results with the surveyed works.

REFERENCES

1. L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," Computer Networks, vol. 54, no. 15, pp. 2787–2805, 2010, doi: 10.1016/j.comnet.2010.05.010.
2. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, 2013, doi: 10.1016/j.future.2013.01.010.
3. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015, doi: 10.1109/COMST.2015.2444095.
4. J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1125–1142, Oct. 2017, doi: 10.1109/JIOT.2017.2683200.
5. P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, protocols, and applications," Journal of Electrical and Computer Engineering, vol. 2017, Art. no. 9324035, 2017, doi: 10.1155/2017/9324035.
6. A. Maier, A. Sharp, and Y. Vagapov, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the Internet of Things," in Proc. 7th Int. Conf. Internet Technologies and Applications (ITA), Wrexham, U.K., Sep. 2017, pp. 143–148.
7. M. Babiuch, P. Foltýnek, and P. Smutný, "Using the ESP32 microcontroller for data processing," in Proc. 20th Int. Carpathian Control Conf. (ICCC), May 2019, pp. 1–6.
8. P. Foltýnek, M. Babiuch, and P. Šuránek, "Measurement and data processing from Internet of Things modules by dual-core application using ESP32 board," Measurement and Control, vol. 52, no. 7–8, pp. 970–984, 2019, doi: 10.1177/0020294019857748.
9. L. O. Aghenta and M. T. Iqbal, "Low-cost, open source IoT-based SCADA system design using Thinger.IO and ESP32 Thing," Electronics, vol. 8, no. 8, Art. no. 822, 2019.
10. N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," in Proc. IEEE Int. Systems Engineering Symp. (ISSE), Vienna, Austria, Oct. 2017, pp. 1–7, doi: 10.1109/SysEng.2017.8088251.
11. Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," IETF RFC 7252, Jun. 2014.
12. K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," ICT Express, vol. 5, no. 1, pp. 1–7, 2019.
13. U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," IEEE Communications Surveys & Tutorials, vol. 19, no. 2, pp. 855–873, 2017, doi: 10.1109/COMST.2017.2652320.
14. W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.
15. R. Piyare, "Internet of Things: Ubiquitous home control and monitoring system using Android based smart phone," International Journal of Internet of Things, vol. 2, no. 1, pp. 5–11, 2013.
16. R. K. Kodali, V. Jain, S. Bose, and L. Boppana, "IoT based smart security and home automation system," in Proc. Int. Conf. Computing, Communication and Automation (ICCCA), Greater Noida, India, Apr. 2016, pp. 1286–1289.
17. T. Ojha, S. Misra, and N. S. Raghuwanshi, "Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges," Computers and Electronics in Agriculture, vol. 118, pp. 66–84, 2015.
18. J. M. Talavera et al., "Review of IoT applications in agro-industrial and environmental fields," Computers and Electronics in Agriculture, vol. 142, pp. 283–297, Nov. 2017, doi: 10.1016/j.compag.2017.09.015.
19. S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, "The Internet of Things for health care: A comprehensive survey," IEEE Access, vol. 3, pp. 678–708, 2015, doi: 10.1109/ACCESS.2015.2437951.
20. L. D. Xu, W. He, and S. Li, "Internet of Things in industries: A survey," IEEE Transactions on Industrial

© 2026 IJSRET

Informatics, vol. 10, no. 4, pp. 2233–2243, Nov. 2014, doi: 10.1109/TII.2014.2300753.

21. S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," Computer Networks, vol. 76, pp. 146–164, 2015, doi: 10.1016/j.comnet.2014.11.008.
22. C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," Computer, vol. 50, no. 7, pp. 80–84, 2017, doi: 10.1109/MC.2017.201.

© 2026 IJSRET