JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Implementation constraints and Efficient-Hardware for the Realization of an Embedded System for IOT Applications

[1]P.K.Sharma,[2]Jayant Bhardwaj,[3]Paras Sharma, [4]Shivam Singh

[1]Associate Professor,[2]Assistant Professor,[3,4]Students
[1]Department of Electronics and Communication Engineering,
[1]Bhagwan Parshuram Institute of Technology, New Delhi, India

Abstract : The exponential expansion of Internet of Things (IoT) devices has given rise to an escalating demand for embedded systems that prioritize energy efficiency. These systems serve a crucial function in facilitating a diverse array of applications, ranging from smart cities and industrial automation, healthcare monitoring and environmental sensing. In the realm of IoT deployments, energy efficiency emerges as a paramount concern, given that numerous devices rely on batteries with limited capacity or harness energy from their immediate surroundings. Nowadays, the Internet of Things (IoT) has garnered significant attention in various domains, such as healthcare, smart living, wearable technology, and defense applications. Despite the myriad of applications, many IoT applications necessitate energy-efficient devices, akin to battery-powered smart watches and healthcare devices. The power consumption of such devices can be mitigated by employing energy-efficient components like sensors, microcontrollers, peripheral devices, and memory, among others. This scholarly article directs its attention towards the development of efficient hardware for the real-time execution of IoT applications.

IndexTerms **- Internet of Things (IoT), IoT application, IoT challenge, wearable sensors, standards, challenges.**

I. INTRODUCTION

The proliferation of the Internet of Things (IoT) has brought about a new era of connectivity, transforming ordinary objects into intelligent entities capable of exchanging information and performing tasks autonomously. As IoT technologies continue to reshape industries and lifestyles, the demand for energy-efficient embedded systems has become increasingly evident. These systems, which serve as the foundation of IoT devices, determine their operational longevity, performance, and environmental impact [1]. This introduction sets the stage for comprehending the importance of energy efficiency in the context of embedded systems for IoT, emphasizing the challenges posed by limited power resources and the crucial role of optimized design strategies [2][3].

Embedded systems have long been the unsung heroes of various applications, seamlessly integrating hardware and software to carry out specific functions. The emergence of IoT has propelled these systems into the spotlight, transforming them into interconnected nodes that collectively constitute the framework of smart environments [4][5][6]. However, the convergence of computing and connectivity brings forth a unique set of challenges, most notably the need to minimize energy consumption. Unlike traditional embedded systems, IoT devices are often constrained by limited battery capacity or the availability of harvested energy, necessitating innovative approaches to maximize operational lifespans [7][8]. Energy efficiency emerges as a critical foundation in the design and deployment of IoT devices. These devices encompass a wide range, from small sensors to complex edge computing platforms, each with distinct energy requirements. The inherent trade-off between computational performance, responsiveness, and power consumption complicates the pursuit of optimal energy usage [9][10]. In order for IoT devices to truly be transformative, they must not only offer intelligent capabilities but also operate reliably and sustainably for extended periods. Achieving this balance requires a comprehensive approach that includes hardware architecture, software optimization, communication protocols, and power management strategies [11]. Addressing the energy challenge necessitates a multidimensional strategy that covers the entire system lifecycle. Hardware design principles play a crucial role in shaping power efficiency. Low-power processors, dynamic voltage and frequency scaling (DVFS), and specialized hardware accelerators are key facilitators of energy-efficient computation. Additionally, innovative sensor technologies, such as MEMS (Micro-Electro-Mechanical Systems), contribute to minimizing energy consumption during data acquisition [12][13].

Software design also exerts a significant influence on energy efficiency. Advanced techniques in power management, such as sleep modes and duty cycling, ensure that devices transition intelligently between active and low-power states. Furthermore, edge computing paradigms optimize data processing and analysis at the device level, thereby reducing the necessity for continuous data transmission to centralized servers. Through data aggregation, sensor fusion, and local decision-making, the consumption of energy-intensive wireless communications is minimized, while enabling real-time responsiveness [14][15].

Within the study, case studies and practical examples are presented to demonstrate the application of energy-efficient design principles in real-world scenarios involving the Internet of Things (IoT) [16]. The impact of these strategies on system performance, responsiveness, and overall user experience is discussed. Additionally, the research paper addresses the challenges associated with maintaining security and reliability when optimizing for energy efficiency.

II. CHALLENGES FOR IMPLEMENTING REAL-TIME EFFICIENT IOT SYSTEMS

Due to the presence of diverse hardware devices and communication protocols, as well as the need for real-time data exchange, connectivity to the cloud, remote control capabilities, and interoperability, security, and reliability within the expansive and varied ecosystem of Internet of Things (IoT) applications, the role of IoT is of utmost importance. The implementation of energy-efficient embedded systems for IoT poses numerous challenges due to the limited power resources and the requirement to optimize various components of the system. Several key challenges can be identified:

IoT Standards: IoT devices have the capability to connect through multiple networks, such as Wi-Fi, cellular, Bluetooth, LoRaWAN, and others. To facilitate seamless communication across different technologies, IoT standards must be able to accommodate this network heterogeneity. One of the primary objectives of IoT standards is to enable interoperability among devices manufactured by different companies. However, achieving this objective can be challenging given the wide range of communication protocols and data formats. IoT encompasses a vast array of devices, technologies, and industries, each having its own specific requirements and limitations [17]. The development of a standard that encompasses all possible use cases becomes challenging due to this diversity. Certain IoT applications demand low-latency or real-time communication, such as industrial automation or autonomous vehicles. Consequently, standards should address these latency requirements while working within the constraints of the involved devices and networks. Numerous IoT deployments need to integrate with existing legacy systems and technologies. Therefore, IoT standards should offer solutions to bridge the gap between the old and the new systems. IoT systems often consist of a large number of devices distributed across various geographical locations. Hence, standards should possess scalability in order to support the increasing number of devices and the growing volume of data. Many IoT devices possess limited processing power, memory, and energy resources [5][18]. Consequently, standards must take into account these limitations to ensure effective communication among devices with varying capabilities. Moreover, since many IoT devices operate on battery power, it is necessary to have energy-efficient communication protocols in place in order to prolong the lifespan of the devices without frequent battery replacements.

Software Design for IoT Devices and Applications: The design of software for IoT devices and applications, which effectively manages power consumption while providing the necessary functionality, is a complex task. It necessitates a multidisciplinary approach that combines knowledge of embedded systems, networking, security, and software architecture. Moreover, this task is accompanied by unique challenges arising from the diversity of devices, communication protocols, and scalability requirements [17][19].

Communication protocols must be carefully selected for the purpose of data exchange between devices and the cloud or among devices. The IoT community frequently demonstrates the use of popular protocols such as constrained application protocol (CoAP), message queue telemetry transport (MQTT), extensible messaging and presence protocol, advanced message queuing protocol, data distribution service, low power wireless personal area network, Bluetooth low energy, and ZigBee in the development of intelligent IoT applications [4][19].

Real-time responsiveness is a critical requirement for certain IoT applications, such as healthcare, security, surveillance, and industrial control of robotic machines. Existing communication technologies are capable of meeting the demands for reliability, connectivity, data rate, and latency. However, the complex nature of real-time applications necessitates the integration of multiple technologies to achieve end-to-end solutions. The challenge lies in achieving both low latency and energy efficiency, which may require the utilization of hardware acceleration [4]. Real-time IoT applications typically operate on battery power or energy harvesters. To make real-time decisions based on energy estimations, it is crucial to consider the power profile of the power source specific to the application. Both energy modeling and energy consumption estimation techniques must take this into account [20][21].

The utilization of low-power hardware components, such as microcontrollers and sensors, can significantly reduce energy consumption in IoT devices. Therefore, designing and utilizing such components is crucial for achieving energy efficiency in IoT devices. Addressing the challenges associated with energy efficiency in IoT necessitates a multidisciplinary approach that encompasses expertise in low-power hardware design, software development, data management, communication protocols, and power management strategies. Overcoming these challenges is imperative for fully realizing the potential of real-time IoT systems, while also ensuring their sustainability and longevity.

III. FUTURE DEMAND FOR EMBEDDED HARDWARE IN IOT APPLICATIONS

The demand for embedded boards, modules, and systems at the hardware level has experienced a notable increase. This is due to a combination of factors, including the expansion of both greenfield and brownfield computing deployments, as well as original equipment manufacturers (OEMs) opting to outsource more hardware integrations in order to concentrate internal resources on core software, cloud services, and application-specific support.

Embedded hardware suppliers that are able to navigate these challenges effectively are poised to benefit from increased demand and enhance their competitive position. The landscape of the embedded space has already witnessed significant movement since 2020, which is particularly noteworthy as this sector has traditionally been slow-moving. The rapid pace of innovation in enterprise/IT and consumer sectors has now permeated into the embedded and edge domain, resulting in heightened competitive pressure for embedded hardware suppliers. As a result, embedded form factors and hardware architectures are evolving to support more efficient and high-performance computing applications in various field environments.

Embedded hardware now serves as the foundation for processing new edge/IOT workloads, leading to greater convergence with the enterprise/IT domain and increasing the importance of hardware differentiation. In addition to meeting the traditional

requirements of performance, power consumption, connectivity, networking, hardening, and security, embedded hardware must also provide targeted workload support, including hardware accelerations and supporting software development.

IV. Hardware used for real-time IOT applications

1. Arduino Board for IOT applications

Arduino boards are the preferred option among researchers due to their user-friendly development environment, rendering them an exceptional choice for novices and individuals with limited knowledge of electronics and programming. The Arduino Integrated Development Environment (IDE) offers a straightforward and easily comprehensible interface. These boards are the primary selection for researchers embarking on projects related to the Internet of Things (IoT), owing to their user-friendly nature, versatility, and a thriving community of developers. Their popularity stems from their open-source platform for constructing electronics projects and real-time IoT projects. All Arduino IoT-enabled products are endorsed on the Arduino IoT Cloud, which permits the logging, visualization, and analysis of sensor data, the initiation of events, and the automation of residential or commercial spaces. Arduino boards have gained significant popularity due to their cost-effectiveness, and this affordability proves advantageous when it comes to prototyping and executing small-scale IoT projects. Arduino boasts an extensive ecosystem comprising boards, shields, and libraries that seamlessly integrate into IoT projects. This streamlines the process of hardware and software development, as numerous components and sensors are readily accessible. Several Arduino boards are specifically designed to operate with minimal power consumption, making them well-suited for battery-powered or energy-efficient IoT devices. Additionally, it is possible to implement power-saving techniques to optimize energy usage. The Arduino IDE is compatible with a wide array of operating systems, including Windows, macOS, and Linux, ensuring its accessibility across various platforms. Arduino boards can be conveniently expanded through the incorporation of various shields (plug-in modules) that confer supplementary functionality, such as wireless communication, GPS, and motor control, among other capabilities. This modular approach simplifies IoT hardware development. Arduino supports programming in C/C++, rendering it suitable for developers well-versed in coding. Nevertheless, its simplified syntax and extensive libraries also make it accessible to beginners.

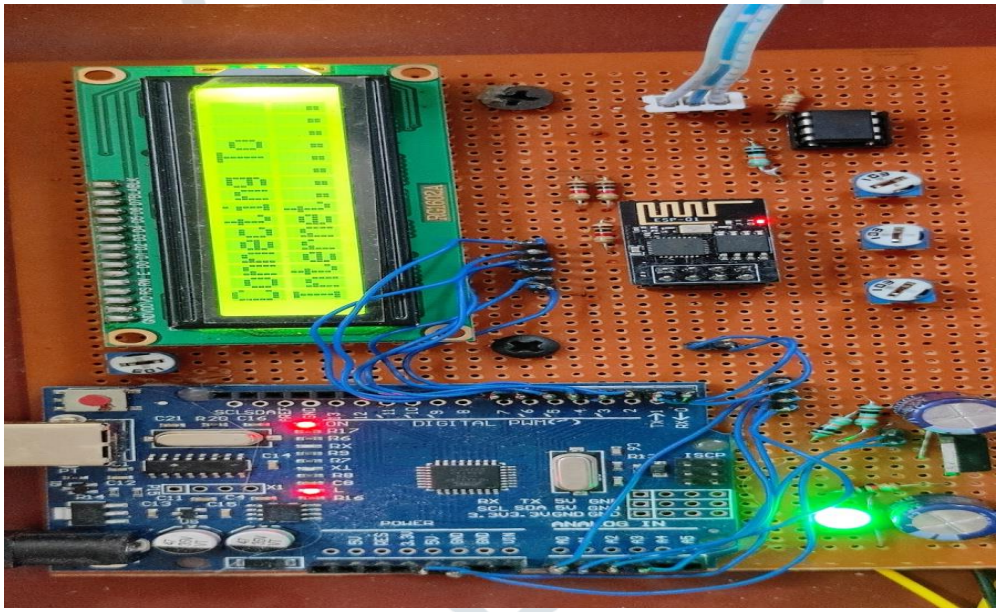

Fig.1 Arduino board used with Wi-Fi module ESP01

As demonstrated in Figure 1, Arduino exhibits compatibility with a wide array of sensors and actuators, thus facilitating its integration with diverse environmental and input devices, including temperature sensors, accelerometers, cameras, and other similar components. The simplicity and adaptability of Arduino render it an exceptional selection for swift prototyping, enabling developers to promptly iterate and assess Internet of Things (IoT) concepts. Arduino provides real-time control capabilities, which prove indispensable for applications necessitating precision timing and responsiveness, such as robotics or sensor data acquisition. Arduino has the capacity to communicate with additional devices, platforms, and services through the utilization of standardized communication protocols like Wi-Fi, Bluetooth, Zigbee, and MQTT. Consequently, Arduino can be effectively adapted to various IoT ecosystems. Although Arduino is commonly associated with hobbyist and educational utilization, it also encompasses commercial and industrial variations, such as the Arduino Industrial line, which caters to more demanding IoT applications in professional settings. The user-friendly approach, affordability, modularity, expansive ecosystem, and community support inherent to Arduino attribute to its compelling nature for IoT applications, particularly for the development and prototyping of IoT solutions that prioritize ease of use and accessibility.

In the domain of IoT applications utilizing Arduino, several limitations can be identified:

1. Arduino boards are typically founded on microcontrollers, such as the ATmega series, which possess limited processing power when compared to more potent microprocessors. This constraint may pose challenges when handling intricate tasks or data-intensive computations within IoT applications.
2. Arduino boards are commonly equipped with limited RAM and flash memory, thereby imposing restrictions on the size and complexity of executable programs. Consequently, the management of voluminous datasets or the implementation of advanced algorithms may prove arduous.

3. IoT devices are often susceptible to security threats. Arduino boards may not inherently possess robust security features, and the implementation of stringent security measures can be intricate.
4. The administration of a large number of individual Arduino devices can become unwieldy in the context of expansive IoT deployments. In such scenarios, more streamlined solutions, such as Raspberry Pi or specialized IoT development platforms, ma y prove better suited.
5. Certain IoT applications necessitate strict real-time capabilities. Arduino boards may not be optimally suited for tasks that mandate precise timing or low-latency communication.
6. Divergent from certain IoT platforms that operate on comprehensive operating systems, Arduino boards directly execute single-threaded programs on the hardware. This characteristic can impose limitations on multitasking capabilities and complicate the management of intricate software architectures.
7. Numerous Arduino boards lack built-in Wi-Fi or cellular connectivity, which are commonly indispensable for IoT applications. Although it is possible to incorporate external shields or modules to enable connectivity, this augmentation amplifies the cost and complexity of the system.
8. Power consumption represents a salient concern within IoT devices, particularly when considering their protracted operation on battery power. Certain Arduino boards may exhibit a propensity for high power consumption, rendering the optimization of power usage a challenge.

2. NODEMCU ESP8266 for IOT applications

Espressif's Node MCU ESP8266 is a widely utilized development board that is founded upon the ESP8266 Wi-Fi module. It has gained popularity in the realm of Internet of Things (IoT) applications due to its affordability, integrated Wi-Fi, GPIO pins, strong community support, programming flexibility, and modular design, rendering it an exceptional choice for a diverse range of IoT applications. Its user-friendly nature and extensive ecosystem contribute to its value as a valuable tool for IoT developers. The ESP8266 module comes equipped with essential built-in Wi-Fi capabilities that are necessary for IoT applications requiring wireless communication and internet connectivity. This simplifies the process of connecting devices to the internet and IoT platforms.
Node MCU ESP8266 boards possess the capability to be programmed using various programming languages such as the Arduino IDE, Lua, Micro Python, among others, which provides developers with different skill sets and preferences the flexibility they desire. By incorporating a multitude of sensors and modules, the functionality of NodeMCU ESP8266 boards can be expanded, thereby rendering them suitable for a vast array of IoT applications. Due to their ease of use, built-in Wi-Fi, and readily available libraries, NodeMCU ESP8266 boards are highly advantageous for rapid prototyping and development purposes. This enables developers to swiftly iterate and test their IoT concepts.
The ESP8266 module has been deliberately designed to consume minimal power, making it a suitable option for battery-powered IoT devices. It supports sleep modes and power-saving techniques that effectively maximize battery life. By implementing Over-the-Air (OTA) updates on Node MCU ESP8266 devices, individuals are granted the ability to remotely update and maintain their IoT applications without necessitating physical access to the devices themselves. Additionally, Node MCU ESP8266 can seamlessly integrate with popular IoT platforms and cloud services, such as MQTT, AWS IoT, Google Cloud IoT, and others, facilitating data storage, analysis, and remote control.
The ESP8266 is equipped with real-time programming capabilities, rendering it suitable for applications that mandate precise timing and responsiveness, such as sensor data acquisition and control systems. When appropriate safeguards are implemented, NodeMCU ESP8266 can also be employed in commercial and industrial applications.

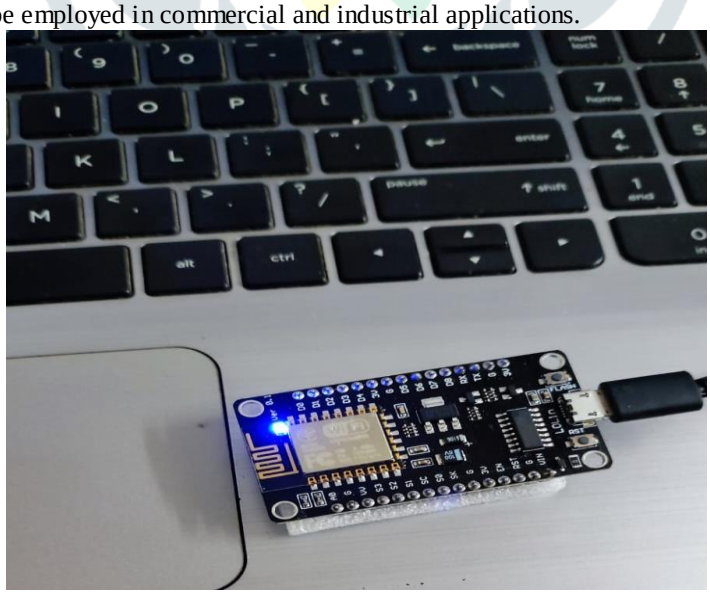

Fig.2 wifi connection establishment using ESP 8266

The Node MCU ESP8266 is a well-liked option for Internet of Things (IoT) ventures due to its economical price, integrated Wi-Fi capabilities, and user-friendly nature, as depicted in figure 2. Nonetheless, like any hardware platform, it possesses certain constraints that can impact its suitability for specific IoT implementations. Presented below are some common limitations associated with the Node MCU ESP8266:
The ESP8266 possesses a relatively modest central processing unit (CPU) and random access memory (RAM), which can restrict its capacity to handle intricate computations or tasks necessitating substantial processing power. It may encounter difficulties with computationally intensive IoT applications. The Node MCU ESP8266 has only one analog input pin, which may prove insufficient for ventures requiring multiple analog sensors. The embedded flash memory of the ESP8266 is limited, which can

pose a limitation if there is a need to store large volumes of data or firmware updates. This limitation can impact Over-The-Air (OTA) updates for IoT devices. The Node MCU ESP8266 typically functions at a 3.3V voltage level, which can restrict the selection of sensors and peripherals that require varying voltage levels. Furthermore, it may not be suitable for battery-powered applications due to its power consumption. The ESP8266 lacks advanced security features that can be found in more contemporary IoT platforms. Implementing strong security measures for IoT devices may necessitate additional effort and external components. The Wi-Fi range of the ESP8266 may prove inadequate for long-range IoT applications. The utilization of external Wi-Fi range extenders or alternative wireless communication options such as LoRa or NB-IoT may be necessary. The ESP8266 does not possess hardware memory protection and isolation, meaning that a flaw or malfunction in one section of the code can potentially crash the entire device.

Despite these limitations, the Node MCU ESP8266 can still be a feasible choice for numerous IoT applications, particularly those with moderate processing and memory requirements. However, for more demanding or specialized IoT projects, it may be advisable to consider more powerful and feature-rich microcontrollers or microcontroller units (MCUs) such as the ESP32, Raspberry Pi, or other IoT platforms that offer enhanced scalability and performance.

3. Hardware implementation of real-time IOT applications using ESP 32

The ESP32 is a widely used microcontroller that integrates Wi-Fi and Bluetooth capabilities, making it well-suited for a diverse range of Internet of Things (IoT) applications. With its dual-core Tensilica Xtensa LX6 processor, the ESP32 enables efficient multitasking and simultaneous handling of various tasks, which is particularly advantageous for real-time processing and communication in IoT applications. Its built-in Wi-Fi and Bluetooth capabilities support both classic Bluetooth and Bluetooth Low Energy (BLE), enabling seamless wireless connectivity to the internet and other devices, making it an ideal choice for IoT projects.

The ESP32 is specifically designed to ensure low power consumption, making it suitable for IoT devices powered by batteries. It offers several power-saving modes, such as deep sleep, that effectively prolong the battery life of IoT devices. Security is a critical aspect of IoT applications, and the ESP32 addresses this by incorporating features like hardware acceleration for cryptographic operations, secure boot, and flash encryption, thereby enhancing the security of IoT devices. Additionally, the ESP32 supports various communication protocols, including MQTT, CoAP, HTTP, and more, making it compatible with a wide range of IoT ecosystems and cloud platforms.

One of the distinguishing characteristics of the ESP32 is its adherence to open-source hardware and software principles. This means that users have access to the design files and can modify the firmware as necessary for their specific application, as illustrated in Figure 3. Another notable feature is Over-the-Air (OTA) Updates, which allows for remote firmware updates of IoT devices. The combination of wireless connectivity, processing power, low power consumption, and extensive community support makes the ESP32 a robust choice for a broad spectrum of IoT applications, ranging from home automation and sensor nodes to industrial IoT and wearable devices. However, it is important to be aware of the limitations of the ESP32 when considering it for a project.

While the ESP32 offers low-power modes, its power consumption can be relatively high when the Wi-Fi and Bluetooth radios are active. This can be a concern for battery-powered IoT devices, necessitating careful power management to optimize battery life. Additionally, the ESP32 has limited RAM compared to more powerful microcontrollers and microprocessors. This limitation can be a constraint when working with large data sets or complex applications that require a substantial amount of memory. Furthermore, although the dual-core processor is suitable for many IoT tasks, it may not provide sufficient computational power for applications that require intensive computation, such as image processing or machine learning. In such cases, a more powerful microcontroller or dedicated hardware may be necessary.

While the ESP32 is a powerful and versatile option for numerous IoT applications, it is essential to carefully consider its limitations, particularly in terms of power consumption, processing power, and range, when selecting it for a specific project. Evaluating the requirements and constraints of the project is crucial in determining whether the ESP32 is the appropriate choice or if an alternative platform would be more suitable.

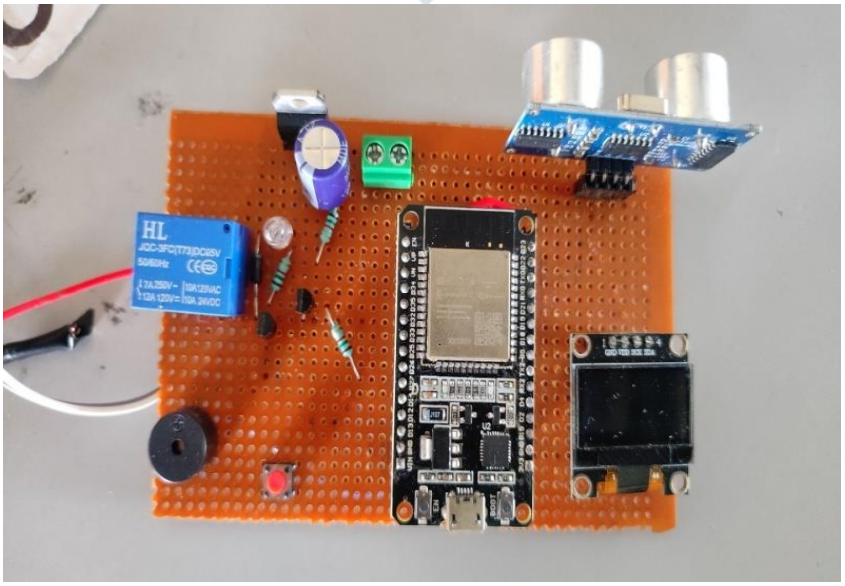

Fig.3 An ESP-32 development board

IV. RESULTS AND DISCUSSION

In this paper, first we identified the different constraints that will degrade the performance of real-time IOT applications. Second, we have focused on the hardware mostly used now a day for real-time IOT applications. We found that the hardware used by the researchers have certain constraints depending upon the system designed for a specific application using real-time IOT. For example, in IOT applications like health care devices, Industries, etc. energy efficient and low latency system are required. In this paper, we have compared and analyzed the performance of different hardware normally used for real-time IOT applications. We analyzed that the hardware boards Arduino UNO, NodeMCU ESP 8266, ESP 32 , on the basis of different constraints like security, operating system, Communication Latency, wireless connectivity, processing power, low power consumption, etc. We found that ESP 32 has good features like wireless connectivity, processing power, low power consumption, and extensive community support make it a strong choice for a wide range of IoT applications, from home automation and sensor nodes to industrial IoT and wearable devices.

REFERENCES

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.
[2] N. Kaur and S. K. Sood,"An energy-efficient architecture for the internet of things (IoT)," *IEEE Syst. J.*, vol. 11, no. 2, pp. 796–805, Jun. 2017.
[3] J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, Feb.2014.
[4] J. Sheth and B. Dezfouli, "Enhancing the energy-efficiency and timelinessof iot communication in wifi networks," IEEE Internet of Things Journal,vol. 6, no. 5, pp. 9085–9097, 2019.
[5] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, "IoT middleware:Asurvey on issues and enabling technologies," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 1–20, Feb. 2017.
[6] T. Lewis, M. Perkowski, and L. Jozwiak, "Learning in hardware: Architectureand implementation of an FPGA-based rough set machine," in*Proc. 25th IEEE EUROMICRO Conf.*, 1999, pp. 326–334.
[7] G. Suciu, T. Us, urelu, C. Beceanu, and M. A. Dobrea, "Iot and energy efficiency for smart agriculture using adcon telemetry devices," in 2018International Symposium on Fundamentals of Electrical Engineering (ISFEE).IEEE, 2018, pp. 1–6.
[8] N. Kaur and S. K. Sood, "An energy-efficient architecture for the internet of things (IoT)," *IEEE Syst. J.*, vol. 11, no. 2, pp. 796–805, Jun. 2017.
[9] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, ``Asurvey on Internet of Things: Architecture, enabling technologies, securityand privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5,pp. 1125_1142, Oct. 2017.
[10] Andrea, C. Chrysostomou, and G. Hadjichristo_, ``Internet of Things: Security vulnerabilities and challenges," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2015, pp. 180_187.
[11] L. Lyu, C. Chen, S. Zhu, and X. Guan, "5g enabled codesign of energyefficienttransmission and estimation for industrial iot systems," IEEETransactions on Industrial Informatics, vol. 14, no. 6, pp. 2690–2704,2018.
[12] W. Wang, H. Yang, Y. Zhang, and J. Xu, "Iot-enabled real-time energyefficiency optimisation method for energy-intensive manufacturing enterprises,"International Journal of Computer Integrated Manufacturing,vol. 31, no. 4-5, pp. 362–379, 2018.
[13] W. T. Hartman, A. Hansen, E. Vasquez, S. El-Tawab, and K. Altaii,"Energy monitoring and control using internet of things (iot) system," in2018 Systems and Information Engineering Design Symposium (SIEDS).IEEE, 2018, pp.13–18.
[14] F. Karim Shaikh, S. Zeadally, and E. Exposito, "Enabling technologiesfor green Internet of Things," *IEEE Syst. J.*, vol. 11, no. 2, pp. 983–994,Jun. 2017.
[15] J. Ding, M. Nemati, C. Ranaweera, and J. Choi, "IoT connectivitytechnologies and applications: A survey," *IEEE Access*, vol. 8,pp. 67646–67673, 2020.
[16] S. Popli, R. K. Jha, and S. Jain, "A survey on energy efficient narrowbandInternet of Things (NBIoT): Architecture, application and challenges,"*IEEE Access*, vol. 7, pp. 16739–16776, 2019.
[17] Z. Sheng, D. Tian, and V. C. M. Leung, "Toward an energy and resourceefficient Internet of Things: A design principle combining computation,communications, and protocols," *IEEE Commun. Mag.*, vol. 56, no. 7,pp. 89–95, Jul. 2018.
[18] O. Hahm, E. Baccelli, H. Petersen, and N. Tsiftes, "Operating systems forlow-end devices in the internet of things: A survey," *IEEE Internet ThingsJ.*, vol. 3, no. 5, pp. 720–734, Oct. 2016.
[19] Research and Markets, *Embedded Internet of Things (IoT) Ecosystem:Next Gen Embedded System Hardware, Software, Tools, andOperating Systems 2016-2021*, 2016. [Online]. Available: http://www.researchandmarkets.com/research/cqnpdc/embedded-internet, Accessed:Nov. 20, 2016.
[20] P.Stenumgaard, J.Chilo, Challenges and conditions for wirelessmachine-to-machine communications in industrial environments, IEEECommunications Magazine, vol.51, no.6, pp.187-192, 2019.
[21] S. Verma, Y. Kawamoto, Z. M. Fadlullah, H. Nishiyama, and N. Kato,``A survey on network methodologies for real-time analytics of massiveIoT data and open research issues," *IEEE Commun. Surveys Tuts.*, vol. 19,
no. 3, pp. 1457_1477, 3rd Quart., 2017.