Digital Scoreboard Using ESP32

AUTHOR: SHUBHAM KUMAR KANAUGIA, SHUBHAM KUMAR, VISHAL GOYAL, VIRANCH DADHEECH Dr. AKHIL PANDEY, Dr. VISHAL SHRIVASTAVA

Department of Artificial Intelligence and Data Science, Student of Artificial Intelligence and Data Science Arya College of Engineering & I.T. India , Jaipur

Abstract: This work presents the design and implementation of a real-time digital scoreboard using Internet of Things (IoT) technology. The scoreboard gets score updates from a specially designed web interface via an ESP32 microcontroller with built-in Wi-Fi. Shown on an LED screen, the scores offer a dynamic and readable presentation for live sports events. All the hardware uses general wiring; the system operates on a 5V power adapter. Lightweight, cheap, easy to add into other sports, the system is The work demonstrates how web technologies and embedded systems might be combined to provide smart, efficient, real-time solutions for sports management.

Index Terms: ESP32, Digital Scoreboard, Internet of Things (IoT), Web Interface, Real-time Score Update, LED Display, Embedded Systems

1.Introduction:

In contemporary sports and competitions, digital scoreboards play a crucial role in giving real-time information to viewers and players. The conventional scoreboard is usually costly and inflexible. With the evolution of IoT and embedded technology, low-cost Wi-Fi-enabled microcontrollers such as the ESP32 provide new prospects for designing intelligent and configurable scoreboard systems. This project involves creating a digital scoreboard that can accept inputs from a web interface and project them onto an LED screen, allowing effective management of scores from remote locations.

2. Literature Review :

Existing work identifies the use of microcontrollers and wireless communication in the implementation of smart scoreboards as a growing trend. A lot of the systems use Arduino Uno or Raspberry Pi boards to implement their systems, but both of these platforms lack native Wi-Fi, which adds system cost and complexity.

ESP32 is a dual-core system-on-chip microcontroller with inbuilt Wi-Fi and Bluetooth. Its low cost, low power, and support for the Arduino IDE make it well-suited for IoT applications. Researchers have proved its effectiveness in home automation, weather stations, and remote data logging systems.

As far as display systems are concerned, 7-segment displays and LED matrices are

widely applied to embedded design applications because they are easy to read and modular. This paper improves upon such basic technologies by integrating them in a visually appealing, remotely operable digital scoreboard.
Earlier projects had usually employed manual switches or infrared remote controls, yet these have some limitations regarding use and range. Incorporating an internet-based interface provides greater user access and accessibility via smartphones, laptops, or any web-accessible device.

3. Objective:

- The primary goals of this project are:
- To implement a wireless digital scoreboard with ESP32.
- To show team scores on an LED display.
- To provide real-time score updates through a web interface.
- To employ a 5V power supply for effective operation.
- To provide system scalability and portability.

4. Methodology:

4.1 Hardware Components-

ESP32: Wi-Fi microcontroller for communication and control.
LED Display (Matrix or 7-segment): For displaying scores of teams legibly.
Power Adapter (5V): For powering ESP32 and LED display.
Connecting Wires: Basic jumper wires to connect the parts.

4.2 Software Tools-

Arduino IDE: For coding the ESP32.
HTML/CSS/JavaScript: For building the web interface.
Wi-Fi Communication: Facilitates real-time data transfer between the web page and the ESP32.

4.3 System Workflow-

ESP32 gets connected to Wi-Fi.
There is a web interface hosted either locally or on ESP32.
User enters scores of two teams through the website.
ESP32 accepts the data and displays the LED display in real time.

5. System Design and Implementation

5.1 Circuit Diagram-
ESP32 is linked to the LED display through GPIO pins.
Both ESP32 and LED are powered by a 5V adapter.
All connections are established through standard male-to-female jumper cables.

5.2 Web Interface-
There's an easy HTML webpage to input scores for Team A and Team B.
There are buttons to increment, decrement, or reset scores.
JavaScript makes requests to the ESP32 to update the display accordingly.

5.3 ESP32 Code-
ESP32 has a web server.
Handles the GET/POST requests for updating scores.
Displays

6. Benefits:

- Portable and low cost.
- Wireless control in real time.
- Simple to use using web browser.
- Customizable for various sports.
- Multi-display scalable.

7. Drawbacks:

- Constrained within Wi-Fi range (local network).
- Manual score entry required.
- Slightly lags under poor signal conditions. values on the LED screen with the respective libraries.

8. Results and Discussion:

The last deployment of the digital scoreboard system delivered successful results that achieved the desired project goals. After completing hardware integration and software deployment, the system was subjected to several cycles of testing in controlled indoor conditions and semi-outdoor environments. The performance was measured in terms of connectivity, responsiveness, precision of score refreshes, visibility, and usability.

8.1 Wi-Fi Connectivity and Web Interface Performance-

The ESP32 microcontroller joined the local Wi-Fi network without lag (usually in less than 5 seconds). After joining the network, the microcontroller operated a light-weight web server hosting the scoreboard control page. The web page could be accessed over IP address using any smartphone, laptop, or tablet on the same network.

The responsiveness of the interface was appreciated. Score entries made via the web interface were calculated and appeared on the LED display within a period of less than 200 milliseconds. With low latency like this, updates were made on the scores real time without recognizable lag, perfect for live match environments. Interface consistency and use was tested under different browsers and devices.

8.2 Display of Scores and Legibility-

The LED display employed in the installation could show numeric scores for two teams (Team A and Team B). The format of the display was made simple and minimal to prevent confusion during a live match. Brightness and font size of the LED display were set for maximum readability, with the display being visible from a distance of 10–15 meters under daylight and indoor lighting conditions.

When tested in a mock match setting, scores were continuously updated (e.g., following each goal or point), and the display correctly updated the changes without failure. The modular design of the LED display also supports future upgrades (e.g., displaying game timer, team names, or match status).

8.3 Power Consumption and Hardware Stability-

Powered by a typical 5V/2A power adapter, the system maintained its stability during around-the-clock operation for more than 2 hours without heating up or slowing down. Both the ESP32 and LED display functioned properly with very low current consumption, suggesting the possibility of battery-powered models in the future.

Wiring between pieces was subjected to durability and electrical stability testing. There was no notable interference or power loss during tests, even with managing high-speed score changes.

8.4 User Experience and Feedback-

Test users such as students, coaches, and enthusiasts informed that the web interface was very easy to use and intuitive. Increment, decrement, and reset buttons immediately responded. In contrast to the old manual scoreboards or remote-controlled ones, it is possible with this system for scorekeepers to update from anywhere within Wi-Fi range, enhancing operational convenience.

Users also enjoyed the ease of setup. Because no extra software or mobile application was required, anyone with general technical skills could use the scoreboard through a browser alone.

8.5 Overall System Evaluation-

The computerized scoreboard system was an inexpensive, stable option that can be applied for small-scale sports activities, like school games, practice sessions, and leisure games. Although it does not equal the feature richness of commercial scoreboards employed in professional stadiums, it is excellent in terms of accessibility, mobility, and flexibility.

The modular design and open-source-based nature of the system further boost its suitability for educational applications, allowing students and hobbyists to tinker with, learn, and expand its functionality.

.

9. Conclusion:

Effective use of an IoT-driven digital scoreboard demonstrates the potential of integrating embedded systems with advanced web technologies. The ESP32 microcontroller was a stable core module, which allowed real-time wireless communication between the output display and the input interface. The LED display ensured that score updates were clearly and instantly visualized, and the system operated in a smooth manner during test matches.

This system fills the gap between performance and affordability, making smart scoreboards available even in grassroot-level sporting events. It also shows how simple IoT concepts—connectivity, automation, and remote control—can be used to improve user experience and operational efficiency in real-world applications.

The project has potential to be explored more in the area of smart sports management systems, perhaps integrating more sensors, automated score recognition, or scaling the solution to multiple-game domains. It is a modest but valuable contribution to the broader ecosystem of IoT applications in education, sport, and events.

10. References:

- Espressif Systems. (2023). *ESP32 Technical Reference Manual* (Version 4.5). Espressif Systems. https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf
- Arduino Project Hub. (2022). *ESP32 Web Server Tutorial*. Arduino. https://projecthub.arduino.cc
- Adafruit Industries. (2021). *Adafruit LED Backpack: LED Matrix Display Guide*. Adafruit Learning System. https://learn.adafruit.com/adafruit-led-backpack/
- Mozilla Developer Network (MDN). (2023). *JavaScript Reference*. Mozilla Foundation. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference
- Shah, M., & Patel, A. (2020, July). *Open-source platforms for IoT applications: A comparative study*. In *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)* (pp. 1–6). IEEE. https://doi.org/10.1109/CONECCT50063.2020.9198503
- Sethi, P., & Sarangi, S. R. (2017). *Internet of Things: Architectures, protocols, and applications*. *Journal of Electrical and Computer Engineering*, 2017, Article ID 9324035, 1–25. https://doi.org/10.1155/2017/9324035
- Vujović, V., & Maksimović, M. (2015). *Raspberry Pi as a sensor web node for home automation*. *Computers & Electrical Engineering*, 44, 153–171. Elsevier. https://doi.org/10.1016/j.compeleceng.2015.01.019
- Kumar, A., & Patel, M. (2021). *IoT-based scoreboard using ESP8266 and web interface*. *International Journal of Innovative Research in Computer and Communication Engineering*, 9(4), 345–350. http://www.ijircce.com/upload/2021/april/65_IOT.pdf
- Kurniawan, A. (2018). *Internet of Things Projects with ESP32: Build exciting and powerful IoT projects using the ESP32 microcontroller board*. Packt Publishing. ISBN: 9781789958039
- Banzi, M., & Shiloh, M. (2014). *Getting Started with Arduino* (3rd ed.). Maker Media, Inc. ISBN: 9781449363338