Automatic Target Tracking System with YOLO and Stepper Motor Control

Author: Alfonso Custodio

Date: 12/26/2024

Table of Contents

Introduction	**1**
Objective	**1**
Components and Materials	**2**
Methodology	**3**
Results and Analysis	**4**
Challenges and Solutions	**6**
Future Improvements	**6**
Real Life Applications	**7**
Conclusion	**8**
References	**9**

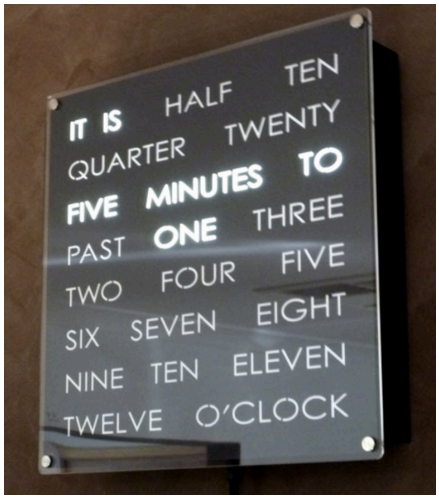
Introduction

Overview

This project is a custom-built Word Clock that tells time using words instead of numbers. It lights up phrases like “IT IS TEN PAST FIVE” using 130 tiny colored lights (LEDs), all controlled by a small computer chip called an ESP32. The clock checks the time from the internet when it turns on and once a day at 4 AM, then turns off its Wi-Fi to save energy. I also designed and built the case myself, using both 3D printing and laser-cut acrylic to make it look clean and modern.

Motivation

I started this project because I wanted to challenge myself beyond classroom theory and apply what I’ve learned to something tangible. While I’ve spent a lot of time studying mechanics, electronics, and design principles, I hadn’t yet combined them into a single, self-contained product. Building a word clock felt like the perfect opportunity to do that. I wanted to explore the complete process, from designing the enclosure in SolidWorks and manufacturing parts with 3D printing and laser cutting, to integrating the electronics and programming the firmware. This project wasn’t just about telling time in a unique way; it was about gaining confidence in my ability to turn an idea into a working, polished final product. It pushed me to think about power efficiency, layout planning, and system integration. And best of all, it was fun to see it all come together.



Objective

Primary Goal: Design and build a functional word clock that displays time in natural language using an ESP32 and 130 WS2812 LEDs.

Secondary Goal: Strengthen skills in CAD modeling, embedded programming, and digital fabrication by designing a custom enclosure and integrating electronics through 3D printing and laser cutting.

Components and Materials

Electronics:

- ESP32 microcontroller
- 130 × WS2812 addressable LEDs (60 LEDs/m density)
- 5V power supply (capable of supplying at least 3–5A)
- 1000 μF capacitor (for power smoothing)
- Female DC barrel jack to screw terminal adapter
- Connecting wires, resistors, and heat shrink tubing

Software:

- Arduino IDE
- FastLED library
- NeoPixelBus library
- Wi-Fi and NTP time libraries

Design and Fabrication Tools:

- SolidWorks (CAD assembly modeling and exploded view)
- 3D printer with PETG filament
- Laser cutter for precision-cut acrylic faceplate

Methodology

Step 1: Proof of Concept

- Got the ESP32 connected to Wi-Fi and successfully synced time using NTP
- Programmed the ESP32 to control five WS2812 LEDs using the FastLED library
- Verified LED signal output, color control, and brightness adjustment

Step 2: Design and Planning

- Mapped LED positions so each one corresponded to a specific word segment
- Created a struct-based system in code to manage each word's LED range
- Designed the enclosure and layout in SolidWorks, including an exploded view

Step 3: Fabrication

- 3D printed the internal structure and base using PETG
- Laser cut the acrylic faceplate to achieve clean, readable word cutouts
- Spray painted the faceplate and base to improve aesthetics and create a polished finish
- Tested all physical parts for fit and adjusted tolerances where needed

Step 4: Firmware Development

- Used the Arduino IDE to program the ESP32 with LED control and phrase logic
- Implemented time rounding to the nearest 5-minute interval
- Controlled LED segments using FastLED and NeoPixelBus libraries

Step 5: Time Synchronization and Power Management

- Synced time via NTP on startup and once daily at 4:00 AM
- Disabled Wi-Fi after sync to conserve power
- Added a 1000 µF capacitor to smooth power to the LEDs and prevent flicker

Step 6: Final Assembly and Testing

- Assembled all parts and routed wiring securely inside the enclosure
- Confirmed time-phrase accuracy through serial monitoring
- Tested for consistent performance and visual clarity during extended runtime

Results and Analysis

Time Synchronization:

- Successfully synced to NTP at startup and daily at 04:00 local time.
- Wi-Fi automatically disabled after sync to minimize power usage.

LED Performance:

- 130 WS2812 LEDs displayed time phrases with high accuracy and clarity.
- Phrases updated every 5 seconds based on the nearest 5-minute interval.
- Manual LED mapping ensured consistent segment activation and visual precision.

Mechanical Design:

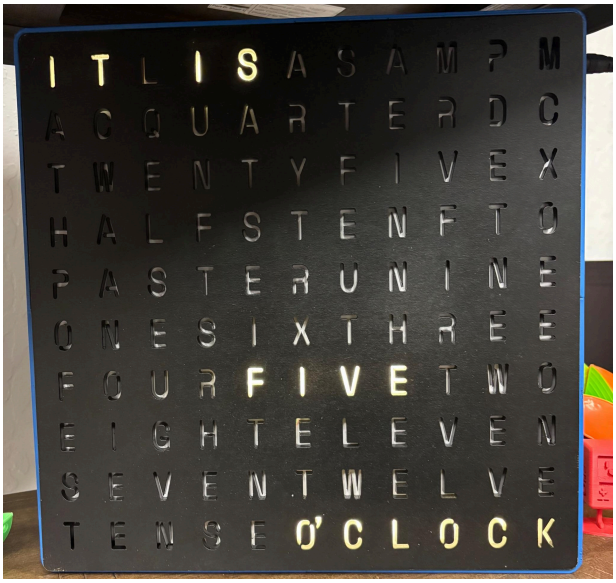
- Hybrid fabrication: laser-cut acrylic (faceplate) and 3D-printed PETG (frame and base).
- Designed in SolidWorks with internal compartments for wiring and ESP32 mounting.
- Snap-fit and screw-accessible structure for streamlined assembly and maintenance.

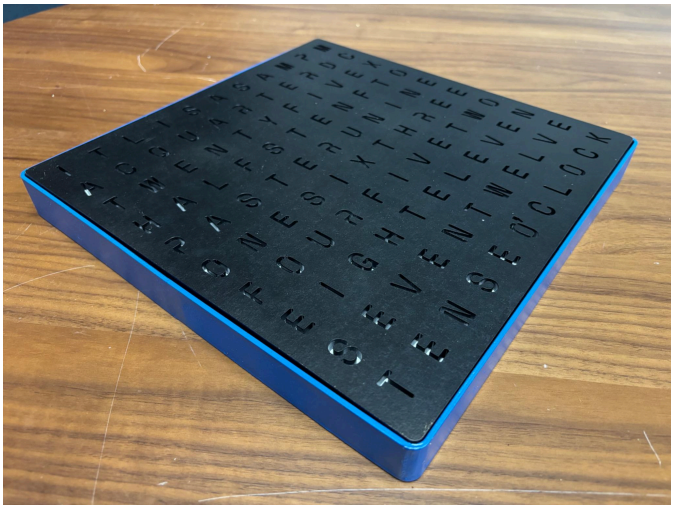
Enclosure Specifications:

- Material: PETG (3D printed) + Black Acrylic (laser cut)
- Dimensions: [12in × 12in × 1.25 in]
- Features: Modular design, internal LED light grid, and secure component housing.

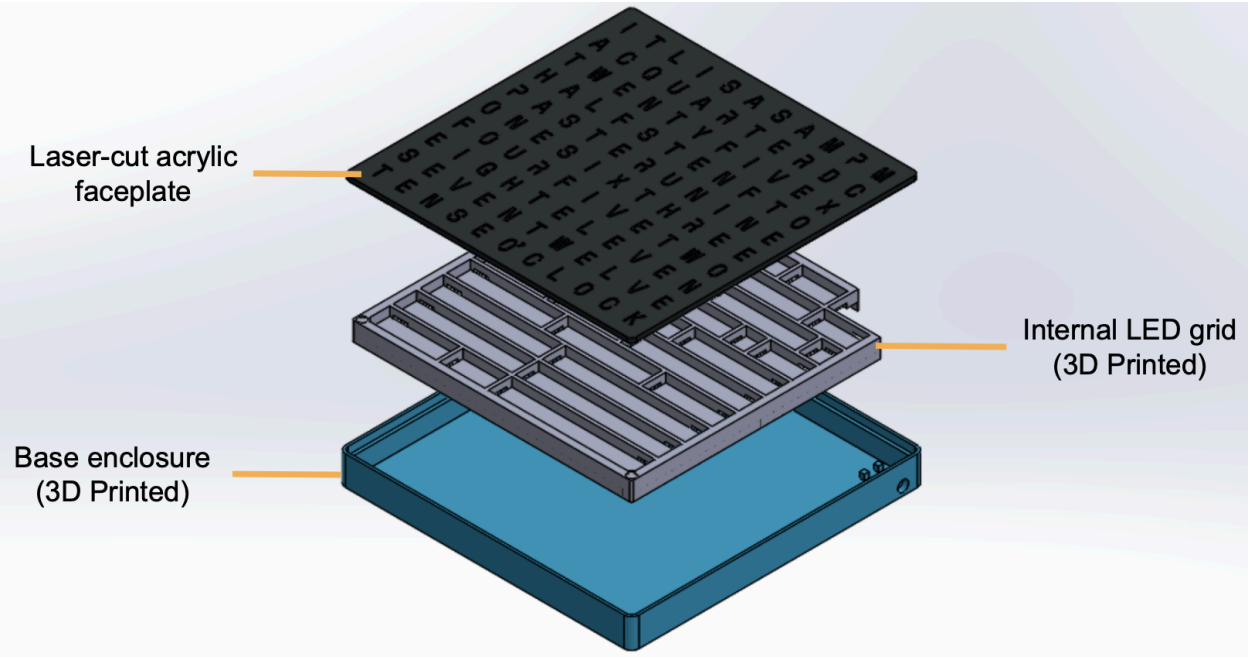
Visuals:

- Real-life setup:





- SolidWorks model:



Link to STP & Solidworks Files Here

Video Demonstration:

Watch the clock in action: YouTube Video Link

Challenges and Solutions

Challenge #1: Wi-Fi Interrupt Conflicts with WS2812 Timing

Description: The timing-sensitive WS2812 protocol conflicted with Wi-Fi radio interrupts on the ESP32, causing LED flicker and unreliable output.
Solution: Time was read via NTP before initializing LEDs. Wi-Fi was disabled before LED output began to ensure uninterrupted control.

Challenge #2: Internal Clock Drift

Description: The ESP32's internal RTC exhibited drift of ±1 second per day, leading to gradual desynchronization.
Solution: Scheduled daily time resynchronization at 04:00 AM local time to keep long-term accuracy without keeping Wi-Fi active.

Challenge #3: 3D Print Size Limitations

Description: The full enclosure exceeded the printable volume of the available 3D printer.
Solution: The model was strategically split into 8 interlocking parts and later joined using adhesive, preserving strength and appearance.

Future Improvements

1. Add a light sensor to enable automatic brightness adjustment based on ambient lighting conditions, improving visibility and energy efficiency.
2. Integrate a real-time clock (RTC) module with battery backup to maintain accurate time without daily NTP synchronization.
3. Replace adhesive joints with magnetic or screw-based assembly for easier maintenance and modularity in future iterations.
4. Improve light diffusion by adding internal light guides or frosted overlays to reduce LED hotspots and enhance uniformity.
5. Implement a settings interface (e.g., via Bluetooth or mobile app) to allow toggling between different time display formats such as 12-hour, 24-hour, or stylized output.
6. Utilize deep sleep modes on the ESP32 with timed wakeups to further lower power consumption during idle periods

Real Life Applications

Consumer Electronics Prototyping
By integrating a microcontroller with LEDs, timing logic, and sensors, I gained hands-on experience in the kind of systems used in smart lighting, wearables, and other connected consumer devices. I also learned to design power-efficient firmware that respects timing constraints, a critical skill in real-world product development.

IoT and Smart Infrastructure
Through combining Wi-Fi communication, real-time NTP synchronization, and low-power operation, I developed foundational IoT skills. These are directly applicable to systems like smart meters, public displays, and time-sensitive infrastructure where reliability and efficiency are essential.

Embedded Systems Development
While working with WS2812 LEDs and resolving conflicts with Wi-Fi interrupts, I deepened my understanding of timing-sensitive embedded systems. This experience translates well to fields like automotive electronics, robotics, aerospace, and medical devices, where real-time performance and stability are critical.

Digital Fabrication and Rapid Prototyping
Using SolidWorks, 3D printing, and laser cutting, I created a hybrid enclosure specifically tailored to house the electronics. These fabrication skills are essential in modern engineering environments where quick iteration and precision are necessary to develop and test functional prototypes.

Low-Power System Design
To avoid unnecessary power draw, I designed the system to disable Wi-Fi immediately after syncing the time. This taught me how to balance functionality with energy efficiency, a key principle in designing battery-powered or remote-deployed systems like environmental sensors and mobile platforms.

Multidisciplinary Engineering Collaboration
This project brought together mechanical design, electrical integration, and embedded programming, all areas I had to navigate and connect myself. It reinforced my ability to think across disciplines and design complete systems, a skill that’s valuable in any cross-functional engineering team.

Conclusion

This project demonstrates the successful integration of embedded systems, time synchronization, and digital fabrication. By combining software, electronics, and mechanical design, I created a fully functional and visually appealing word clock that operates with high accuracy and efficiency. It also emphasized the importance of thoughtful design, timing precision, and system-level problem solving.

Reflection

Working on this project deepened my understanding of real-time systems, low-level programming, and the trade-offs involved in power and timing management. I also gained valuable experience in hybrid fabrication techniques, combining 3D printing and laser cutting to bring my design to life. Most importantly, it showed me how even small projects can involve real-world engineering challenges and cross-disciplinary thinking. I'm excited to keep exploring embedded systems and smart device design, maybe I'll make a watch version of this one day.

References

- **GitHub Repository:** Word Clock
    - Access the full project code, 3D models, and supporting files in the GitHub repository
- **Personal Website:** Alfonso Custodio's Portfolio
    - Explore more of my projects, detailed documentation, and professional portfolio.