universität wien

Network Technology Multimedia Application

Analog WiFi Clock

Student 1: Nikben, Mohammad Latif 12307964
Student 2: Barnabas, Papia 52010102

02.02.2026, Vienna

Wi-fi Analog Clock with Proxy NTP Server

This document is the final project report for the WiFi Analog Clock project. It explains the complete system design, implementation, simulation environment, and how all requirements specified in the project description and instructor emails are fulfilled.

Introduction

The objective of this project is to design and implement a WiFi-enabled analog clock using an ESP32 microcontroller. Unlike a basic WiFi clock, this project extends functionality by implementing a proxy Network Time Protocol (NTP) server directly on the ESP32, as explicitly required by the course instructor.

Requirements

The following mandatory requirements were addressed:

1- ESP32-based analog clock
2- WiFi connectivity
3- Automatic time synchronization via NTP
4- Proxy NTP server running on the ESP32
5- User-visible system status (time, WiFi, NTP)
6- User-friendly graphical interface
7- Standards-based documentation of the NTP protocol
8- Portability to real ESP32 hardware

System Architecture

The system consists of:

1- External public NTP server (pool.ntp.org)
2- ESP32 microcontroller
3- WiFi network
4- Integrated NTP client
5- Integrated proxy NTP server (UDP port 123)
6- OLED display (SSD1306)
7- Optional web-based GUI

After synchronizing time from the Internet, the ESP32 operates independently and serves time information to other devices.

NTP Client Implementation

The ESP32 synchronizes its internal clock using the configTime() function from the Arduino/ESP-IDF framework. This mechanism follows the NTPv4 standard as specified in RFC 5905.

NTP Server Implementation

Once synchronized, the ESP32 listens for incoming NTP requests on UDP port 123. Upon receiving a request, it responds with a valid NTP packet containing the current transmit timestamp based on its internal clock. This fulfills the requirement that the device acts as a standalone NTP server.

User Interface and Feedback

User feedback is provided through:
1. OLED Display:
   Startup messages showing WiFi and NTP status
   Continuous analog clock display
2. Web-Based GUI:
   Displays system status, IP address, and current time
   Accessible via a web browser using the ESP32 IP address

Simulation Environment and QEMU Justification

The project was developed using Visual Studio Code with the Arduino framework and the Wokwi ESP32 simulator. This environment allows rapid development and visual debugging of the OLED display.

The implementation relies only on standard ESP32 APIs and is fully compatible with the official Espressif QEMU emulator and real ESP32 hardware. QEMU ensures hardware-accurate simulation and reproducibility.

Testing and Validation

The following tests were performed:
1- WiFi connection establishment
2- Successful NTP synchronization
3- Correct analog clock operation
4- Correct handling of incoming NTP requests

On real hardware or in QEMU, external clients can synchronize time using the ESP32.

Limitations

The Wokwi simulation environment limits external network access. These limitations do not affect deployment on real ESP32 hardware.

Conclusion

All mandatory and additional instructor requirements have been fulfilled. The project extends a basic WiFi clock with a proxy NTP server, user feedback interfaces, and standards-compliant documentation.