Name: Khaled Soliman

Student ID: 900162031

Embedded Project Option 4

# Project Prompt

In this project we are going to utilize the ESP8266 module to create a small IoT application that can enable the user to perform I/O operations with the STM32 module through a web interface. The I/O operations include:

• Retrieving the date/time from RTC module connected to the STM32 module

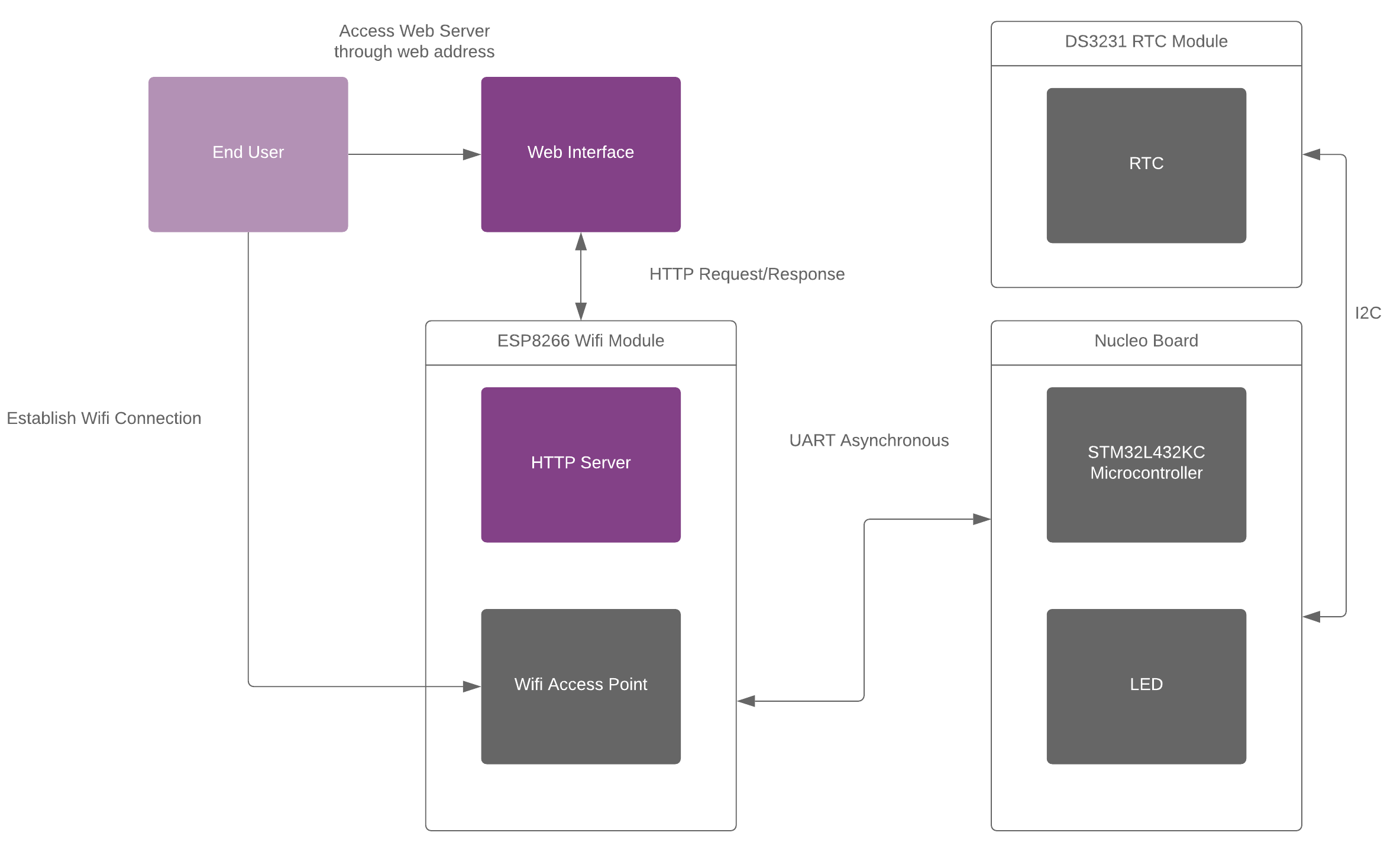
Control the STM32 module LEDs status to build this application:

• The ESP8266 module will be programed to act as a WiFi Access Point and run a simple HTTP server.

• The ESP8266 module communicates with the STM32 module using Asynchronous Serial link (using UART)

• The STM32 module runs code that receives commands from the ESP8266 module to control the LEDs and to retrieve the current time/date from the i2c RTC module attached to it

# Architectural Components:



# Time Plan

**Monday 5/4**

* **Identifying Project Hardware Components [Detailed in a later section]**
* **Installing Prerequisite Software [Detailed in a later section]**

**Thursday 5/7**

* **Configuring HTTP Server on Wifi Module**
* **Establish UART Communication between HTTP Server and uC**
* **Estbalish I2C Bus Communication between RTC and uC**
* **Build 2 APIs:**
  + **Retrieve Time/Date Information**
  + **Toggle LED**

**Identifying Project Hardware Components:**

**STM32 Microcontroller:**

Model:

* STM32L432KC

User Manual:

* <https://www.st.com/resource/en/user_manual/dm00231744-stm32-nucleo32-boards-mb1180-stmicroelectronics.pdf>

Datasheet:

* <https://www.st.com/resource/en/datasheet/stm32l432kc.pdf>

**Wifi Module:**

Model:

* ESP8266MOD

User Manual:

* <http://wiki.ai-thinker.com/_media/esp8266/esp8266_series_modules_user_manual_v1.1.pdf>

Datasheet:

* <https://datasheet.lcsc.com/szlcsc/1811141522_Ai-Thinker-ESP-12F-ESP8266MOD_C82891.pdf>

Version:

* The Wemos D1 mini

Documentation:

* <https://arduino-esp8266.readthedocs.io/en/2.7.0/>

**RTC Module:**

Model:

* DS3231

Datasheet:

* <https://components101.com/sites/default/files/component_datasheet/DS3231%20Datasheet.pdf>

**Installing Prerequisite Software:**

**Arduino IDE:**

* <https://downloads.arduino.cc/arduino-1.8.12-windows.exe>

**Python:**

* <https://www.python.org/downloads/>

**CH340 Driver:**

* <https://github.com/wemos/ch340_driver/raw/master/CH341SER_WIN_3.5.ZIP>

**Installing the esp8266 platform with Arduino Boards Manager:**

* Starting with 1.6.4, Arduino allows installation of third-party platform packages using Boards Manager. We have packages available for Windows, Mac OS, and Linux (32 and 64 bit).
* Install the current upstream Arduino IDE at the 1.8.7 level or later. The current version is on the Arduino website.
* Start Arduino and open the Preferences window.
* Enter https://arduino.esp8266.com/stable/package\_esp8266com\_index.json into the Additional Board Manager URLs field. You can add multiple URLs, separating them with commas.
* Open Boards Manager from Tools > Board menu and install esp8266 platform (and don't forget to select your ESP8266 board from Tools > Board menu after installation).

**Microcontroller Flashing and Debugging IDE Options:**

* Free Version of ARM Keil (Max 32 KB Debugging)
* STM32CubeIDE

**Configuring STM32 Nucleo Board:**

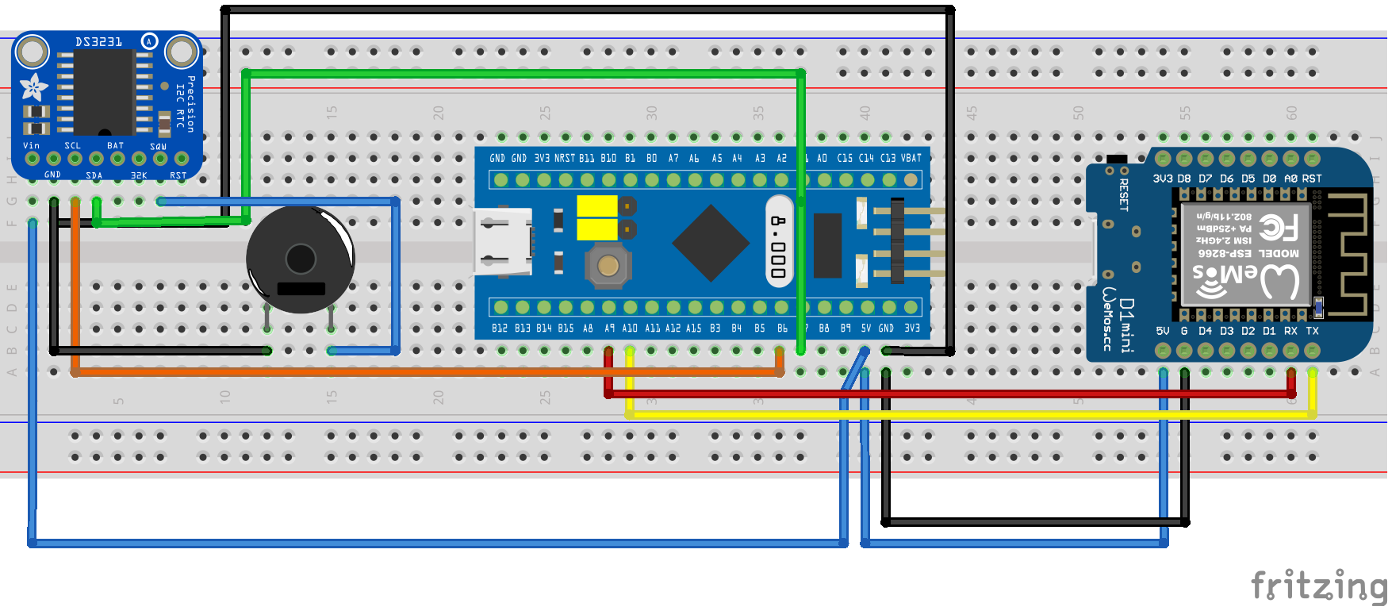
* STM32CubeMX

# Revised Architecture

A screenshot of a cell phone

Description automatically generated

# Connections



# Implementation

## Web Interface

Built with bootstrap 4 css3 plugin, JavaScript, and HTML it provides the interface for the functionality implemented in APIs.

Sample Excerpt Codes:

Creating the socket connection on port 81

websock = new WebSocket('ws://' + window.location.hostname + ':81/');

Hook for on message event, handles any messages received from WebSocket.

websock.onmessage = function(evt) {…}

## Web Server

Built with Arduino C++ it deploys three listener functions one on port 80 for HTTP web server communications and the other on port 81 for WebSocket communications, and the third on the UART Serial bus.

Sample Excerpt Codes:

Loop function for the three listeners.

|  |
| --- |
| void loop(void) { |
| webSocket.loop(); |
| serialListen(); |
| server.handleClient(); |
| } |

## STM Microcontroller

Implements a single listener function on its UART Serial port.

|  |
| --- |
| while (1) |
| { |
|  |
| listen(); |
| } |

# Communication Structure And Functionality

A screenshot of a cell phone

Description automatically generated

# Operation Demo Screenshots

A screenshot of a cell phone screen with text

Description automatically generated

A screenshot of a cell phone

Description automatically generated

# Limitations

* SPIFFS is deprecated and should migrate to LittleFS for better directory structure
* Web Server serves clients on a public socket (Improve websocket communications)
* If any of the HTML or CSS3 files get too large web server faces issues in serving them (Stream data by chunks)
* Implement FreeRTOS or similar functionalities in the microcontroller for better response time/extensibility
* Sync Functionality