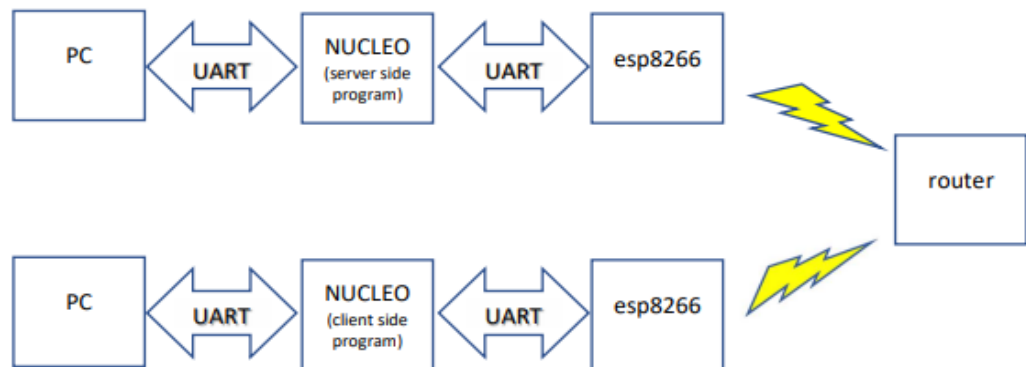


Embedded System-Final Project

By: Saeed Jaber, Ameer Jaber

The following program consists of two programs, a server, and a client. By connecting two STM32 nucleo systems using an ESP8266 modem for each piece, a connection is possible through the Wi-Fi network.

The connection allows the two systems to send and receive message from one another, the main features of this project allow the boards to change each other's LEDS by pressing the blue button.



The above figure details how the connection is made, the PC (Tera term) sends and receive data to the nucleo using USART2 and the nucleo sends and receive data from the ESP8266 modem using USART1.

By transferring the data from the modem using USART1 to the nucleo and from the nucleo to the pc using USART2 the PC can receive and transmit data to the ESP8266.

The router connects both boards together and establish a connection between the two sides.

LED states:

- **OFF:** The LED is turned off.
- **ON:** The LED is turned on (a green light)
- **Blinky:** The LED is blinking, 1 second on and 1 seconds off (frequency of 1/2 HZ).

Main Modules used:

- **Watchdog:** Independent watchdog (IWDG) that will reset the system if it gets stuck an infinite loop, to test it out the user can type "watchdog" command which will generate an infinite loop that will trigger the watchdog to rest the system.
- **Cyclic Buffer:** A buffer for that handles the commands received from USART1 and USART2 in a more efficient way than a regular buffer, is it has pointers to keep track of the data that has been written and read.
- **Event Queue:** A cyclic buffer to handle the events that are generated by the interrupts, used to better manage the events, events such as: button pressed, message has been received from USART1 and USART2
- **USART1:**is used to transmit and receive data from the ESP8266 modem.
- **USART2:**is used to transmit and received data from the PC aka Tera Term.

How to connect cables from modem to board?

- A cable from the modem 3v3 pin to a 3v3/VDD pin.
- A cable from the modem EN pin to a 3v3/VDD pin.
- A cable from the modem GND pin to a GND.
- A cable from the modem RX pin to PA9 pin (USART1 TX)
- A cable from the modem TX pin to PA10 pin (USART1 RX)

How to use?

- **Using STM32CUBEIDE install each program to the board.**
- **For each board open Tera Term.**
- **Click serial and choose the port for the corresponding board.**
- **Click Setup→Terminal**
- **Set the Receive and Transmit to LF and enable Local echo**
- **Ensure Serial port is set at 9600(Setup→Serial Port)**
- **Type "config" and then will be asked to type Wi-Fi name and password.**
- **Do the previous step for both the server and the client.**
- **On the client type "connect" which will then ask for the server IP (it can be found by typing "show" in the server side).**
- **If the connection is successful start pressing the blue button which will send a command to the modem to change the LED state.**

Client-Side Commands:

- **config: a command to set Wi-Fi name and password**
- **show: a command to display IP address.**
- **connect: a command to connect to server by typing its IP address.**
- **watchdog: a command to trigger an infinite loop which will then trigger the board's watchdog and rest it.**
- **Any commands that are supported by the ESP8266 modem**

Server-Side Commands:

- **config:** a command to set Wi-Fi name and password
- **show:** a command to display IP address.
- **watchdog:** a command to trigger an infinite loop which will then trigger the board's watchdog and rest it.
- **Any commands that are supported by the ESP8266 modem**

Configuration:

- **The server side is configured as follows:**
- **AT+CIPWMODE=1**
- **AT+CIPMUX=1**
- **AT+CIPSERVER=1,80**
- **The client side is configured as follows:**
- **AT+CIPWMODE=1**
- **AT+CIPMUX=0**
- **AT+CIPSTART="TCP","/server ip",80**

Extra notes:

- **In case of failure press board black button to rest it**
- **If a link is invalid after many attempts, try to disconnect by typing AT+CWQAP.**
- **Most of the job was done together as a team through zoom meetings. As for parts for each student to explain:**
 - **USART1: Saeed Jaber**
 - **Cyclic Buffer: Ameer Jaber**
 - **Event Queue: Saeed Jaber**
 - **Watchdog: Ameer Jaber**
 - **LED Switch states: Ameer Jaber**
 - **Handling Input: Saeed Jaber**
 - **Communication: Saeed Jaber + Ameer Jaber**