Simple IoT Application Report

Dr Mohamed Shalan
Youmna Sabik
900160339

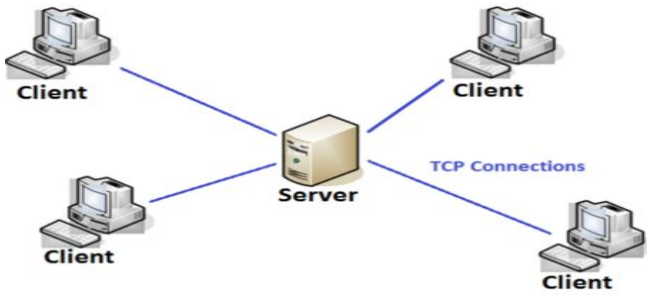
Content

Project Description

The project aims to control the I/O operations on the STM32 module using the RTC module to retrieve the Date and time through a web page interface.

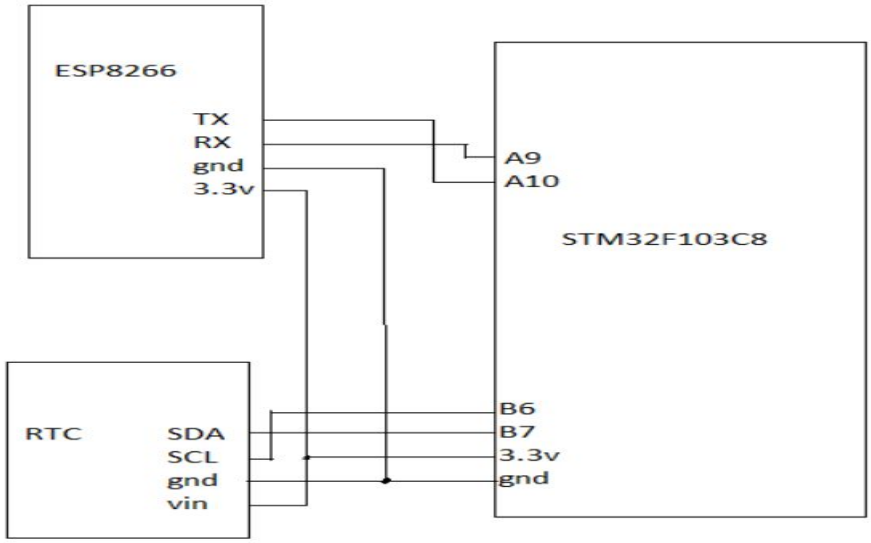
-It is a client server architecture where the ESP8266 module acts as a WiFi access point that connects the Server, STM32 module, to display the Time and date to the client, web page,based on his request, where the Time and date are retrieved from the STM32 module.

-It also controls the LEDs of STM32 module

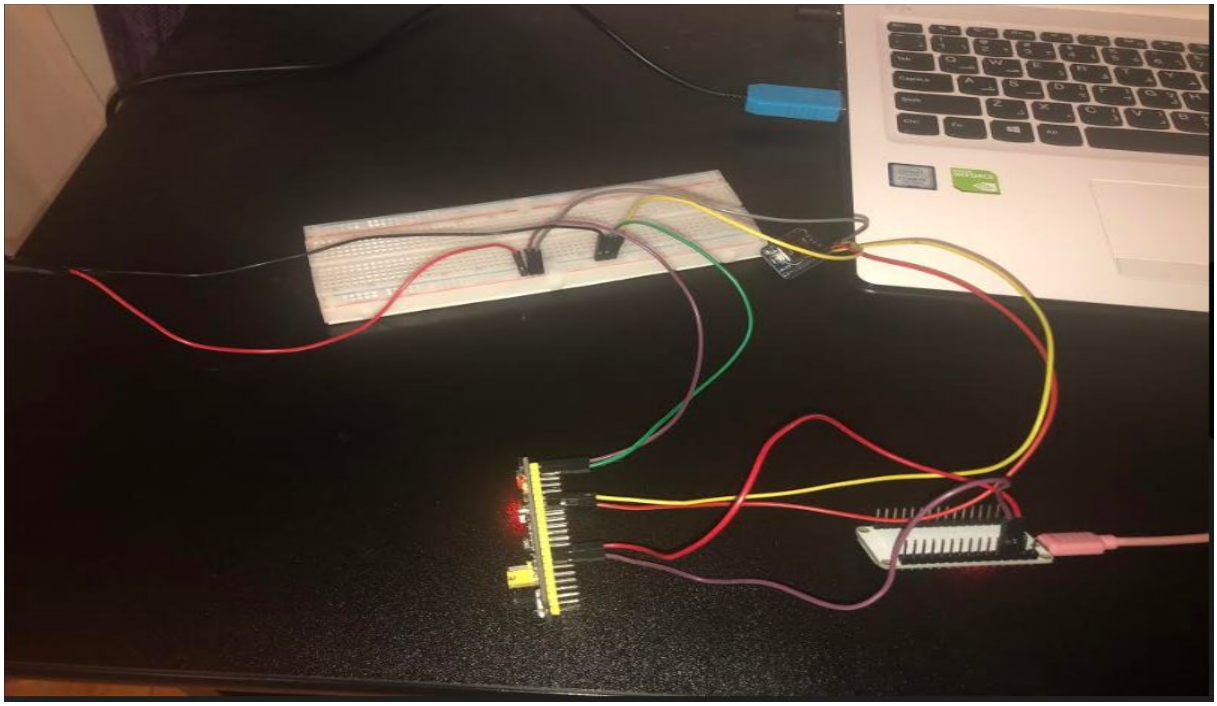


Connection

*Block Diagram



* Circuit



RTC Module

ADDRESS	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0 LSB	FUNCTION	RANGE
00h	0	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12/$\overline{24}$	$\overline{AM}$/PM 20 Hour	10 Hour	Hour				Hours	1–12 + $\overline{AM}$/PM 00–23
03h	0	0	0	0	0	Day			Day	1–7
04h	0	0	10 Date		Date				Date	01–31
05h	Century	0	0	10 Month	Month				Month/ Century	01–12 + Century
06h	10 Year				Year				Year	00–99

-I initially filled the value of the register to be Tuesday 26-5-2020 at 9 Am and then RTC continues clock based on these initialization once connected to power source

-**Seconds** register: is all set to zeros as set to 9AM sharp

-**Minutes** register: is all set to zeros as set to 9AM sharp

-Hours register: 9Am = 9, so will be using the 24 hour therefore set bit 6 to 0, bit 5 to 0 because 9 hours less than 20. Also set bit 4 to zero because less than 10 hours. Then filling bit 0 to bit 3 by "9" 00001001

-DAY register: Based on my assumption, Tuesdayis set to be 3 so will fill register with value 0x03

-DATE register: since the day is 26, therefore bit 5 and 4 set to 10 corresponding to 20 days, and from bit0 to bit 3 filled with remaining 6 days 00(10)0110

-Month register: since its May so its value will be set to 0x05

-Year register: since it is 2020, so its value is set to 0x20

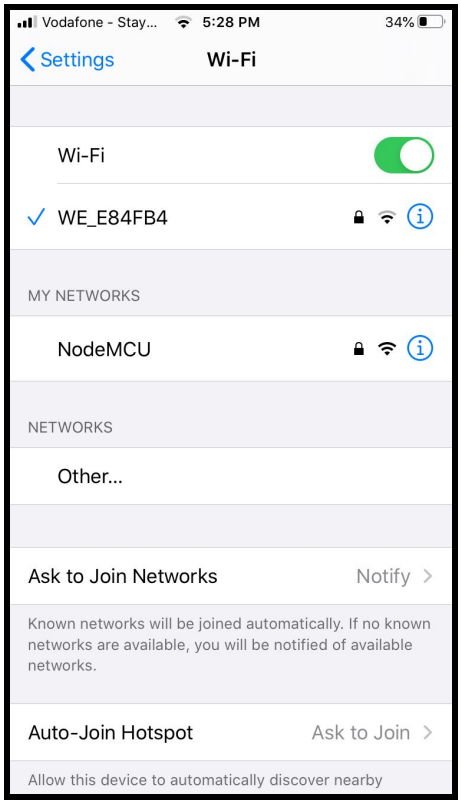
Output: uint8_t out[] =
{'D','A','Y','=',0,0,',',0,0,'-',0,0,'-',0,0,'\r','\n','T','I','M','E','=',0,0,':',0,0,':',0,0,'\r','\n'};
-This is the format of the output from keil code which shows that it is 29 characters.

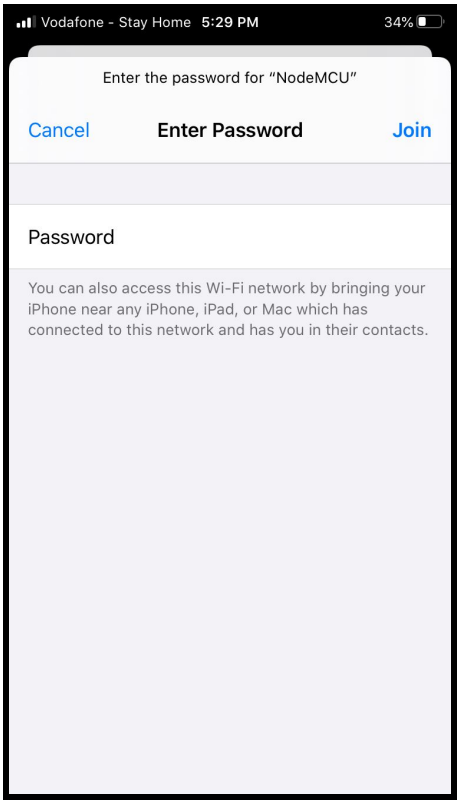
Establishing connection of ESP8266

Wifi

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>

/* Put your SSID & Password */
const char* ssid = "NodeMCU";  // Enter SSID here
const char* password = "12345678";  //Enter Password here

/* Put IP Address details */
IPAddress local_ip(192,168,1,1);
IPAddress gateway(192,168,1,1); //each router has gateway, make devoce connect to router
IPAddress subnet(255,255,255,0); //masking (up to 255 decice can connect to the router)

ESP8266WebServer server(80);  //default port number
```

Initially, setting SSID name and password. In addition, setting local_IP, gateway so that each router has a gateway to make the devices connect to router and subnet which is used to define how many devices can be connected to the router and here they are 255 devices.





Assumptions

D	**Retrieves date and time from STM32 module**
O	**Turns on led**
F	**Turns off led**

```
HAL_UART_Receive(&huart1,&Request, sizeof(Request),50); //receiving the request

    if (Request == 'D')
    {
      HAL_UART_Transmit(&huart1,out, sizeof(out),500); //display the date/time
      Request = '\0';
    }

    else if (Request == 'O')
    {
      HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13,0); //on
      Request = '\0';
    }

    else if (Request == 'F')
    {
      HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13,1); //off
      Request = '\0';
    }
```

When the user requests date and time , wifi module transmits ‘D’ to STM32 module and as a result, STM32 transmits current date and time to user. In Addition, when the STM32 module receives ‘O’ it turns the led of the stm32 module on. Finally, when the STM32 module receives ‘F’ it turns the led of the stm32 module off.

Days assumption

temp[5]	1	2	3	4	5	6	7
weekday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
void handle_DATE_TIME() {
  Serial.print('D');
  delay(1000);
  while(Serial.available()>0)
{temp+= char(Serial.read());}
 if(temp[5]=='1')
 {day=="Sunday";}
 else if(temp[5]=='2')
 {day=="Monday";}
 else if(temp[5]=='3')
 {day="Tuesday";}
    else if(temp[5]=='4')
 {day=="Wednesday";}
 else if(temp[5]=='5')
 {day="Thursday";}
    else if(temp[5]=='6')
 {day=="Friday";}
 else
 {day="Saturday";}

 Final_Date="Day="+day;
 for (int k =6 ; k < 16; k++)
 {Final_Date
 +=temp[k];}
 for (int j =16 ; j < 30; j++)
 {Final_Time+=temp[j];}
server.send(200, "text/html", SendHTML(!LED1status));
}
```

- When the user clicks on the **Date & Time** button,the **handle_DATE_TIME** function is executed so that it transmits **'D'** to STM32 module and based on this the STM32 transmits the current date and time to the user. The stm32 module transmits week days as numbers and based on the assumption as mentioned before. In this function, I make conditions on the 5th character of **temp** string as it is the character that identifies the week day based on the order of output of keil code.

{'D','A','Y','=',0,0,',',0,0,'-',0,0,'-',0,0,'\r','\n','T','I','M','E','=',0,0,':',0,0,':',0,0,'\r','\n'}

- It is already known from the keil code that output of date and time is 29 characters and that time part starts from character 16 to 29. The date part starts from character 6 to character 15.

- The Final_date= day(which is already known from if conditions based on 5th character) + characters from 6 to 15.

HTML Function

```
uint8_t LED1pin = D0;
bool LED1status = LOW; //ACTIVE LOW

String temp="";
String day="";
String Final_Date="";
String Final_Time="";
```

```
 if(led1stat)
{ptr +="<p>LED Status: ON</p><a class=\"button button-off\" href=\"/led1off\">OFF</a>\n";}
else
{ptr +="<p>LED Status: OFF</p><a class=\"button button-on\" href=\"/led1on\">ON</a>\n";}

ptr +="<a class=\"button button-off\" href=\"/DATEANDTIME\">DATE&Time</a>\n";

  String date = "<h3>" +Final_Date+ "</h3>\n";
  ptr +=date;

  String Time = "<h3>" +Final_Time+ "</h3>\n";
  ptr +=Time;

  temp="";
  day="";
  Final_Time="";
  Final_Date="";
```

String temp is used to store all the output(date and time) received from STM32 module. While string day is used to store only one bit of temp which identifies the day based on the assumption i made. Finally, I display the output on the website on two lines, one representing date and other representing time.

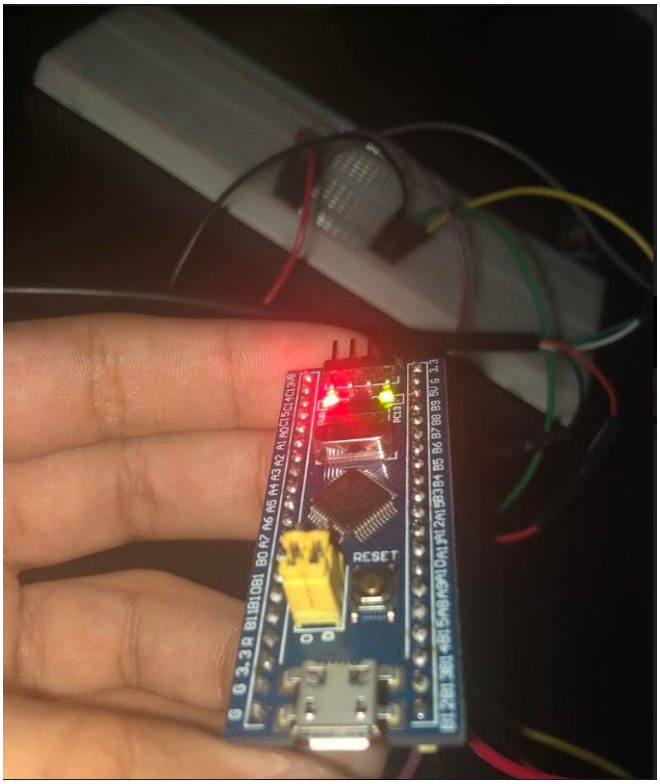
This shows how the date and time are outputted on the web page when the user requests them by clicking on a button. And after they are displayed, I clear all the strings (temp,day,Final_Time,Final_Date)

Controlling LEDS

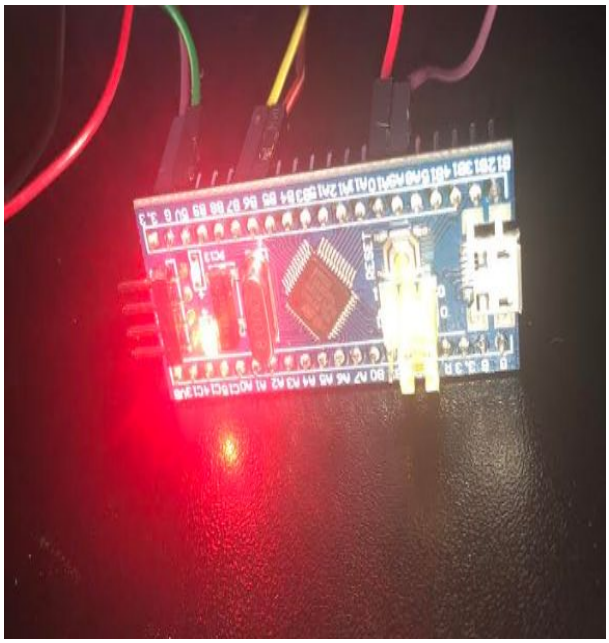
```
void handle_OnConnect() {
  LED1status = LOW;

  server.send(200, "text/html", SendHTML(LED1status));
}

void handle_led1on() {
  Serial.print('O');
  LED1status = LOW;//ACTIVE LOW
  //Serial.println("GPIO7 Status: ON");
  server.send(200, "text/html", SendHTML(true));

}

void handle_led1off() {
  Serial.print('F');
  LED1status = HIGH;//ACTIVE HIGH
  //Serial.println("GPIO7 Status: OFF");
  server.send(200, "text/html", SendHTML(false));

}
```

For turning led on, ‘O’ is transmitted to the STM32 module and this turns the led on and for turning led off, ‘F’ is transmitted to the STM32 module and this turns the led off.

LED Status: ON



LED Status: OFF



DATE&Time

Demo Link

https://drive.google.com/drive/folders/1-6FqKXK60jvp6SmCQqOgCML7yRkypbw0

References

https://lastminuteengineers.com/creating-esp8266-web-server-arduino-ide/

https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

https://blackboard.aucegypt.edu/bbcswebdav/pid-1577852-dt-content-rid-11605851_1/courses/CSCE430204_2020Sp/DS3231%20datasheet.pdf

https://blackboard.aucegypt.edu/bbcswebdav/pid-1558189-dt-content-rid-11337585_1/courses/CSCE430204_2020Sp/Description%20of%20STM32F1%20HAL%20and%20low-layer%20drivers%20%28en.DM00154093%29.pdf