System on Chip Architecture
Lab 3 Report

Daniele Castro S253244
System-on-chip architecture
Politecnico of Turin

CONTENTS

LIST OF FIGURES

System on Chip Architecture
Lab 3 Report

***Abstract*—In this lab I have implemented a simple web server using the MCU STM32F051R8 on the discovery board and the WiFi module SPWF01SX.11 communicating with the microcontroller and using AT commands. Web pages are updated by sending HTML code through UART (RS-232) with TTL voltage levels.**

I. Introduction

During the development of this lab I have met some issues regarding both the code structure and even its behaviour. At first I thought to simply go ahead in the development of my custom commands but, then, I decided to fix, some misbehaviour. Maybe it can be useful to someone in the future.

II. Background

The SPWF01SX.11 module is a MCU itself with his own firmware. So, to make this laboratory faster, it would be better to program the module itself instead of making a communication using the serial UART or SPI port to send commands from the STM32 to the module. Anyway, nowdays these kind of modules are going to be very popular because developing an IOT embedded system is really easy with these modules. We can really control a lot of features with its AT commands such as its ADC peripheral.

III. Proposed Solution

The main problem with the behaviour of this template was that the HTML code sent appeared to be corrupted: full of strange non ASCII characters. Such charachters were so many that some browsers (like chrome) did not visualize the retrieved HTML file and downloaded it, like in this picture:

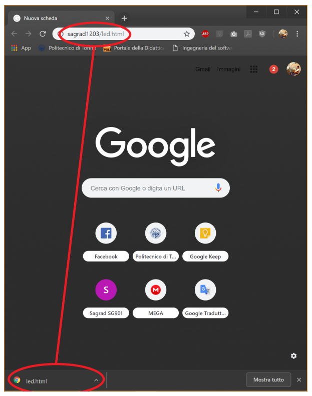
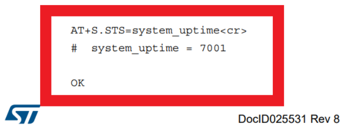


Fig. 1. wrong code download

while trying to fix this issue I also noticed that HTML web pages were stored in the C code with ASCII characters: in embedded systems they are commonly stored in hexadecimal values so that HTML code can be even stored being formatted without modify any character to store it as a C array. So I decided to rewrite the ”led.html” web page with a modern one and to change the storing name from ”led.html” to ”index.html” so that when we simply write in the browser ”http://sagrad1203” we will be automatically redirected to ”http://sagrad1203/index.html” (sagrad1203 is the DNS name avoiding to rewrite the IP address that dinamically changes at every reconnection). Anyway, even fixing what showed a strange character was appearing every time at the beginning of every serially uploaded webpage. I discovered that when serially passing the C string:

```c
uint8_t TxBuffer_Prepare_led_page_upload[] = "at+s
    .fsa=/index.html,1023\r\n";
```

the `'\r'` was stored at the beginning of every serially passed HTML file (maybe because of a firmware bug of the module). So, I removed that character. As a result I obtained a well displayed web page but If I attached my serial port to the communication bus I saw a bed formatting of the serial AT commands. In any case these commands are not normally shown, so it does not matter. At this point there were only few compilation warnings given by an unexplicited C type casting that I corrected. Other behavioural modifications I have made were in adding the form of cgi_demo.html in index.html so that I can directly send commands from that web page without hand write cgi_demo.html. Of course I even had to modify this file because, after test.cgi has been called by the button click, it calls back cgi_demo.html that, now, contains only a waiting message. After 5 seconds page will be redirected back to index.html. During this time the MCU has enough time to re-upload the index.html with the updated LED status. Now, concerning the ”dispall” custom command I have made, I had to send the AT command serially ”AT+S.STS=ip_ipaddr” that, as datasheet explains, would show me the only ”ip_ipaddr” variable. Unfortunally, again maybe because of a firmware bug of the modul, whatever I put after ”=” the ”AT+S.STS” showed me the entire variable environment of the module. So I had to post-process the command on the MCU.



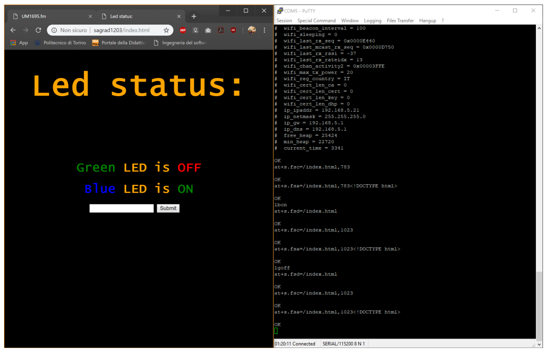
```
AT+S.STS=system_uptime<cr>
#  system_uptime = 7001

OK
```

DocID025531 Rev 8

Fig. 2. User Manual screenshot

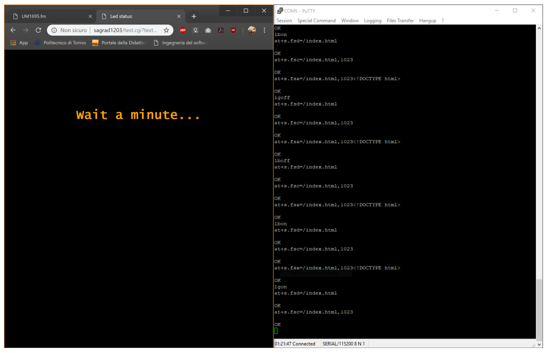
IV. Results and Discussion

Here are some screenshots I have made to show the final result. On the right there is the serial console showing what is happening on the TX pin of the WiFi module so that we can also see the echo of the AT commands sent from the MCU and the module response but not the HTML code sent because the module does not echo it back during a file transfer. On the left, instead, there is a browser showing the web pages:


Led status:
Green LED is OFF
Blue LED is ON
Submit
COM5 - PuTTY
Session Special Command Window Logging Files Transfer Hangup ?

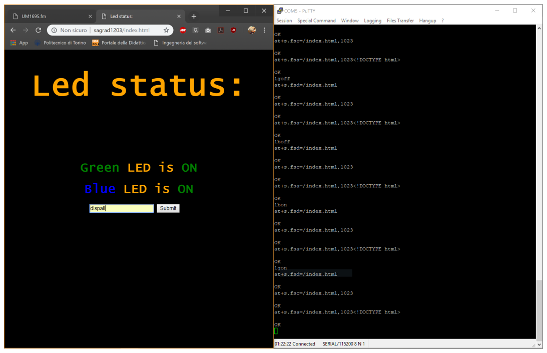
```
#  wifi_beacon_interval = 100
#  wifi_sleeping = 0
#  wifi_last_rx_seq = 0x0000E440
#  wifi_last_mcast_rx_seq = 0x0000D750
#  wifi_last_rx_rssi = -37
#  wifi_last_rx_rateidx = 13
#  wifi_chan_activity2 = 0x00003FFE
#  wifi_max_tx_power = 20
#  wifi_reg_country = IT
#  wifi_cert_len_ca = 0
#  wifi_cert_len_cert = 0
#  wifi_cert_len_key = 0
#  wifi_cert_len_dhp = 0
#  ip_ipaddr = 192.168.5.21
#  ip_netmask = 255.255.255.0
#  ip_gw = 192.168.5.1
#  ip_dns = 192.168.5.1
#  free_heap = 25424
#  min_heap = 22720
#  current_time = 3341

OK
at+s.fsc=/index.html,783

OK
at+s.fsa=/index.html,783<!DOCTYPE html>

OK
lbon
at+s.fsd=/index.html

OK
at+s.fsc=/index.html,1023

OK
at+s.fsa=/index.html,1023<!DOCTYPE html>

OK
lgoff
at+s.fsd=/index.html

OK
at+s.fsc=/index.html,1023

OK
at+s.fsa=/index.html,1023<!DOCTYPE html>

OK

01:20:11 Connected   SERIAL/115200 8 N 1
```

Fig. 3. index web page


Wait a minute...
COM5 - PuTTY
Session Special Command Window Logging Files Transfer Hangup ?

```
OK
lbon
at+s.fsd=/index.html

OK
at+s.fsc=/index.html,1023

OK
at+s.fsa=/index.html,1023<!DOCTYPE html>

OK
lgoff
at+s.fsd=/index.html

OK
at+s.fsc=/index.html,1023

OK
at+s.fsa=/index.html,1023<!DOCTYPE html>

OK
lboff
at+s.fsd=/index.html

OK
at+s.fsc=/index.html,1023

OK
at+s.fsa=/index.html,1023<!DOCTYPE html>

OK
lbon
at+s.fsd=/index.html

OK
at+s.fsc=/index.html,1023

OK
at+s.fsa=/index.html,1023<!DOCTYPE html>

OK
lgon
at+s.fsd=/index.html

OK
at+s.fsc=/index.html,1023

OK

01:21:47 Connected   SERIAL/115200 8 N 1
```

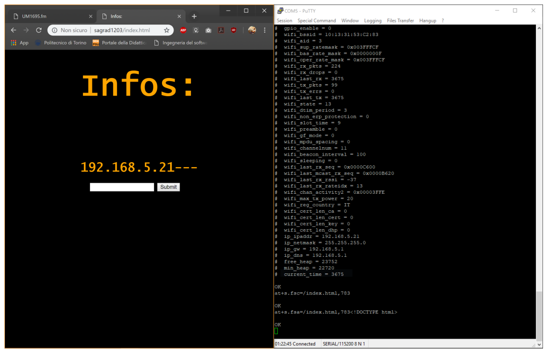
Fig. 4. wait web page



```

OK
at+s.fsc=/index.html,1023

OK
at+s.fsa=/index.html,1023<!DOCTYPE html>

OK
lgoff
at+s.fsd=/index.html

OK
at+s.fsc=/index.html,1023

OK
at+s.fsa=/index.html,1023<!DOCTYPE html>

OK
lboff
at+s.fsd=/index.html

OK
at+s.fsc=/index.html,1023

OK
at+s.fsa=/index.html,1023<!DOCTYPE html>

OK
lbon
at+s.fsd=/index.html

OK
at+s.fsc=/index.html,1023

OK
at+s.fsa=/index.html,1023<!DOCTYPE html>

OK
lgon
at+s.fsd=/index.html

OK
at+s.fsc=/index.html,1023

OK
at+s.fsa=/index.html,1023<!DOCTYPE html>

OK

01:22:22 Connected   SERIAL/115200 8 N 1
```

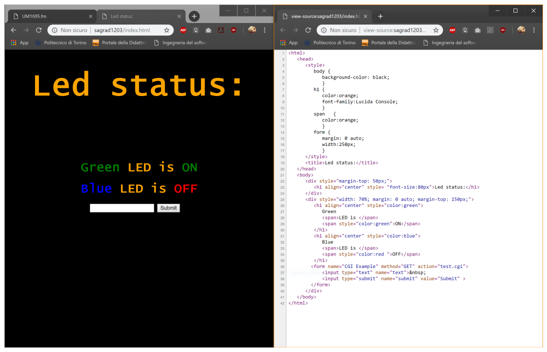
Fig. 5. updated index web page

Note the output of the serial console command ”AT+S.STS” that has to be post processed.


Infos:
192.168.5.21---
Submit
COM5 - PuTTY
Session Special Command Window Logging Files Transfer Hangup ?

```
#  gpio_enable = 0
#  wifi_bssid = 10:13:31:53:C2:83
#  wifi_aid = 3
#  wifi_sup_ratemask = 0x003FFFCF
#  wifi_bas_rate_mask = 0x0000000F
#  wifi_oper_rate_mask = 0x003FFFCF
#  wifi_rx_pkts = 224
#  wifi_rx_drops = 0
#  wifi_last_rx = 3675
#  wifi_tx_pkts = 99
#  wifi_tx_errs = 0
#  wifi_last_tx = 3675
#  wifi_state = 13
#  wifi_dtim_period = 3
#  wifi_non_erp_protection = 0
#  wifi_slot_time = 9
#  wifi_preamble = 0
#  wifi_gf_mode = 0
#  wifi_mpdu_spacing = 0
#  wifi_channelnum = 11
#  wifi_beacon_interval = 100
#  wifi_sleeping = 0
#  wifi_last_rx_seq = 0x0000C600
#  wifi_last_mcast_rx_seq = 0x0000B620
#  wifi_last_rx_rssi = -37
#  wifi_last_rx_rateidx = 13
#  wifi_chan_activity2 = 0x00003FFE
#  wifi_max_tx_power = 20
#  wifi_reg_country = IT
#  wifi_cert_len_ca = 0
#  wifi_cert_len_cert = 0
#  wifi_cert_len_key = 0
#  wifi_cert_len_dhp = 0
#  ip_ipaddr = 192.168.5.21
#  ip_netmask = 255.255.255.0
#  ip_gw = 192.168.5.1
#  ip_dns = 192.168.5.1
#  free_heap = 23752
#  min_heap = 22720
#  current_time = 3675

OK
at+s.fsc=/index.html,783

OK
at+s.fsa=/index.html,783<!DOCTYPE html>

OK

01:22:45 Connected   SERIAL/115200 8 N 1
```

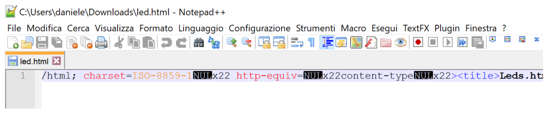
Fig. 6. IP address show web page


Led status:
Green LED is ON
Blue LED is OFF
Submit
view-source:sagrad1203/index.ht
Non sicuro | view-source:sagrad1203...

```html
<html>
   <head>
      <style>
         body {
            background-color: black;
            }
         h1 {
            color:orange;
            font-family:Lucida Console;
            }
         span   {
            color:orange;
            }
         form {
            margin: 0 auto;
            width:250px;
            }
      </style>
      <title>Led status:</title>
   </head>
   <body>
      <div style="margin-top: 50px;">
         <h1 align="center" style= "font-size:80px">Led status:</h1>
      </div>
      <div style="width: 70%; margin: 0 auto; margin-top: 150px;">
         <h1 align="center" style="color:green">
            Green
            <span>LED is </span>
            <span style="color:green">ON</span>
         </h1>
         <h1 align="center" style="color:blue">
            Blue
            <span>LED is </span>
            <span style="color:red ">OFF</span>
         </h1>
        <form name="CGI Example" method="GET" action="test.cgi">
            <input type="text" name="text">&nbsp;
            <input type="submit" name="submit" value="Submit" >
        </form>
      </div>
   </body>
</html>
```

Fig. 7. well formatted HTML code retrieved

I have decided to format the code even if formatting characters are useless for the browser and only slow the transfer but, since real bottle neck here is the UART web page transfer at 115200 baudrate, I decided to maintain formatting characters: for small web pages like this speed is not compromised.


C:\Users\daniele\Downloads\led.html - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro Esegui TextFX Plugin Finestra ?
led.html

```
/html; charset=ISO-8859-1NULx22 http-equiv=NULx22content-typeNULx22><title>Leds.ht
```

Fig. 8. led.html code retrieved

Here, I finally show what I was retrieving when typing ”http://sagrad1203/led.html”: file was downloaded instead of being showed like I sad before and a lot of strange characters were displayed. As I told if I had time I would also reorganized the code because, instead of putting in memory four different possible web page for each possible LED combination like it was done, I would simply memorize a single page divided in two pieces and I would concatenate them with the small piece of code to be changed during the transfer to the WiFi module like I have done for the info web page. I do not write the code here because it is too big. I would have also used `printf()` ; and `scanf();` functions I implemented in past laboratories instead of manually play with the RX buffer. That would have reduced code size and optimized it a lot .

V. Conclusions

I have matched the assignment requests and, as an extra, I have fixed some template bugs restayling webpages with new HTML 5 code.

References

[1] STM32F051R8 Reference Manual (RM0091), [pdf] Available at: Link.
Accessed on: Feb. 10, 2019.

[2] STM32F0DISCOVERY User Manual (UM1525), [pdf] Available at: Link.
Accessed on: Feb. 10, 2019.

[3] SPWF01SX.11 User Manual (UM1695), [pdf] Available at: Link.
Accessed on: Feb. 10, 2019.