

System on Chip Architecture

Lab 3 Report

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

CONTENTS

I	Introduction	1
II	Background	1
III	Proposed Solution	1
IV	Results and discussion	2
V	Conclusions	3
	References	3

LIST OF FIGURES

1	wrong code download	1
2	User Manual screenshot	1
3	index web page	2
4	wait web page	2
5	updated index web page	2
6	IP address show web page	2
7	well formatted HTML code retrieved	2
8	led.html code retrieved	2

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, nowadays 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 characters were so many that some browsers (like chrome) did not visualize the retrieved HTML file and downloaded it, like in this picture:

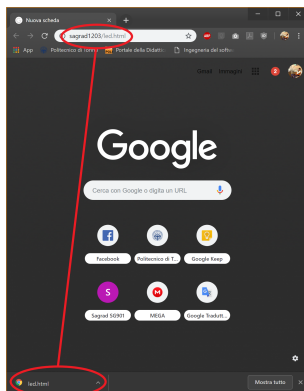


Fig. 1. wrong code download

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 dynamically 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:

```
uint8_t TxBuffer_Prepare_led_page_upload[] = "at+s\n\n.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 bad 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 "dispal" 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. Unfortunately, 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.

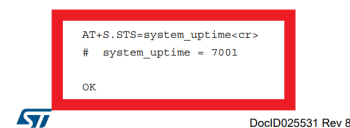


Fig. 2. User Manual screenshot

V. CONCLUSIONS

I have matched the assignment requests and, as an extra, I have fixed some template bugs restyling 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.