AN4902
Application note

SPWF01Sx – dynamic web pages

Introduction

This application notes describes how to create a dynamic web page using the SSI tags and integrated CGIs.

The SPWF01S module offers a set of tag and CGI pages which allow users to build dynamic web pages and interact with them.

Starting from a “Dynamic DEMO website”, this AN shows how to create every web page containing one or more SSI tags.

The HTML source code used in this AN is provided and can be freely used to build a custom web site.

The AN is aligned with the FW 3.5 (version SPWF0xS-160129-c5bf5ce).

Contents

1 Overview

1.1 Current CGIs (hardcoded in the module)

- *output.cgi*: can be used to push data over the module's UART
- *firstset.cgi*: can be used to perform the first configuration of the module
- *remote.cgi*: can be used to perform AT commands remotely (only GPIO control currently supported)
- *input.cgi*: can be used together with input SSI

1.2 Current SSIs (hardcoded in the module)

- *devsts SSI*: user can get module's status variables including this SSI into HTML pages
- *devconf SSI*: user can get module's configuration variables including this SSI into HTML pages
- *peers SSI*: user can get module's peers table including this SSI into HTML pages
- *input SSI*: user can get strings from HOST including this SSI into HTML pages
- *message SSI*: used for error message through HTML pages. This SSI can be ignored

Notes regarding SSI tags:

- Whitespace is allowed between the tag lead-in "<!--#" and the start of the tag name, and between the tag name and the lead-out string "-->".
- HTML file extensions that can correctly run the SSIs: **.shtml**, **.shtm**, **.ssi** or **.xml**.

1.3 CGI vs. SSI

CGI is a source code executed on demand, while SSI is an HTML replacement. These can be used at the same time. For example, in the input_demo.html, the input CGI and input SSI are both used. The first asks the HOST to provide a message for the client, while the second replaces the HTML comment (SSI) with the message provided by the HOST.

1.3.1 CGI & SSI enable/disable

The CGIs and SSIs integrated in the webserver can be enabled/disabled by acting on the ip_use_cgis and ip_use_ssis variables. By default, both variables are set to 0x0000000F (all enabled).

The CGIs and SSIs can be set as shown below:

- OUTPUT_CGI_BIT=0
- INPUT_CGI_BIT=1
- REMOTE_CGI_BIT=2
- FIRSTSET_CGI_BIT=3

- STATUS_SSI_BIT=0
- CONFIG_SSI_BIT=1
- PEERS_SSI_BIT=2
- INPUT_SSI_BIT=3

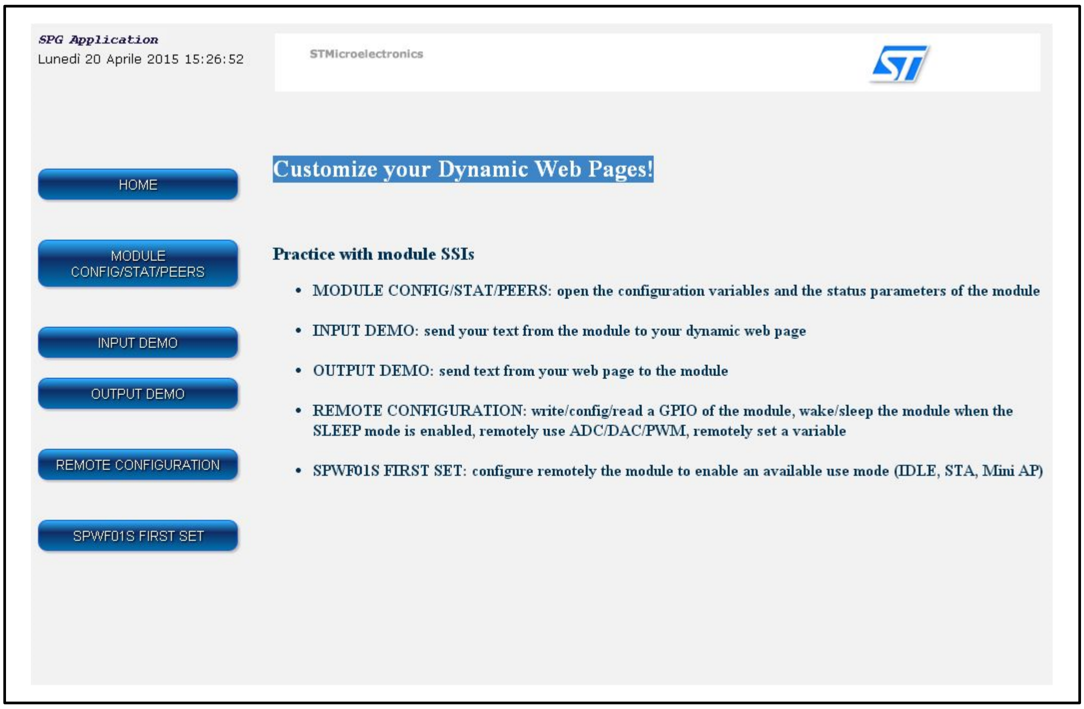
2 Dynamic website

The subsections that follow explain, in detail, how to use the SSIs and CGIs.

An outfile.img file (containing the HTML files and images) is provided and can be uploaded to the module’s external Flash memory using the AT+S.HTTPDFSUPDATE command.

2.1 Index page – index.html

Figure 1: Index page



Customize your Dynamic Web Pages!

Practice with module SSIs

- MODULE CONFIG/STAT/PEERS: open the configuration variables and the status parameters of the module
- INPUT DEMO: send your text from the module to your dynamic web page
- OUTPUT DEMO: send text from your web page to the module
- REMOTE CONFIGURATION: write/config/read a GPIO of the module, wake/sleep the module when the SLEEP mode is enabled, remotely use ADC/DAC/PWM, remotely set a variable
- SPWF01S FIRST SET: configure remotely the module to enable an available use mode (IDLE, STA, Mini AP)

2.1.1 HTML code

```
<html>
<head>
<title>SPWF01S WIFI MODULE</title>
</head>
<frameset cols="23%,*%" frameborder="0" framespacing="0" border="0">
         <frame name="menu" src="menu.html" scrolling="auto" framespacing="0">
     <frameset rows="150, 650">
          <frame name="ST" src="your_banner.html" scrolling="auto" framespacing="0">
          <frame name="homepage" src="home.html" scrolling="auto" framespacing="0">
     </frameset>
<body>
<noframes>Your browser does not support frames.</noframes>
</body>
</html>
```

2.1.2 HTML code description

The Index page is composed of 3 frames. The HTML frames allows users to present documents in multiple views, which may be independent windows or sub-windows. Multiple views offer users a way to keep certain information visible while other views are scrolled or replaced.

On the DEMO page, within the same window, one frame displays a static banner; a second window displays a navigation menu and a third displays the main document, which can be scrolled through or replaced by navigating in the second frame.

In detail:

- the 1st frame is named “menu” and displays menu.html page

```
<frame name="menu" src="menu.html" scrolling="auto" framespacing="0">
```

- the 2nd frame is named “ST” and displays your_banner.html page

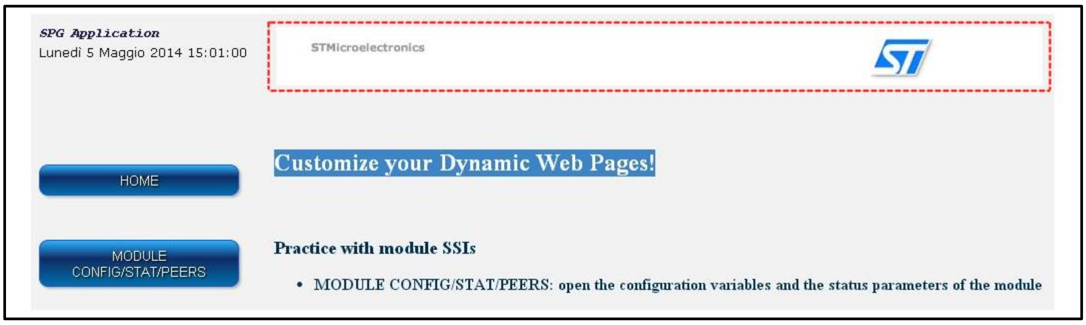
```
<frame name="ST" src="your_banner.html" scrolling="auto" framespacing="0">
```

- 3rd frame is named “homepage” and displays home.html page

```
<frame name="homepage" src="home.html" scrolling="auto" framespacing="0">
```

2.2 Your banner frame – your_banner.html

Figure 2: Dynamic web page customization page



SPG Application
Lunedi 5 Maggio 2014 15:01:00

Customize your Dynamic Web Pages!

Practice with module SSIs

- MODULE CONFIG/STAT/PEERS: open the configuration variables and the status parameters of the module

2.2.1 HTML code

```
<html>

<body bgcolor="#F2F2F2">

<a href="http://www.st.com/wifimodules" target="_blank"><img src="st766.gif"
ALIGN=right BORDER=0 TITLE="SPWF01Sx official website" ATL=TITLE="SPWF01Sx official
website" VSPACE=5 HSPACE=5></a

</body>

</html>
```

2.2.2 HTML code description

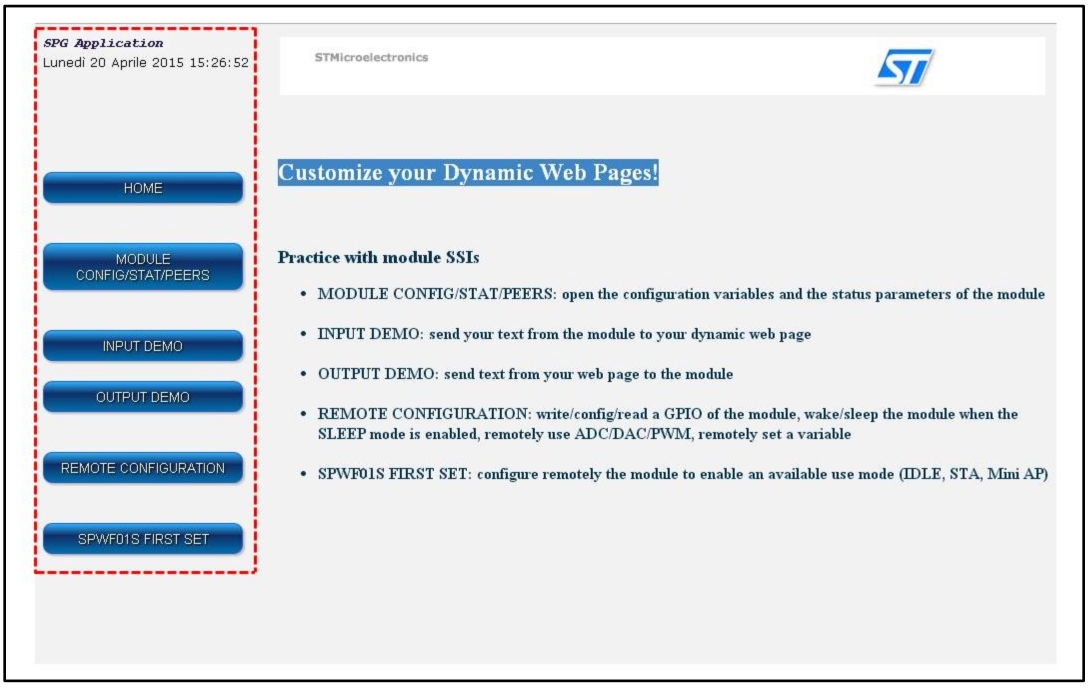
In the banner frame, the user can include a custom image and a target web site. The image can be stored in the external Flash of the module as the HTML pages.

In detail:

- The code `<a href="http://www.st.com/wifimodules"` is used to link a web site.
- The code `<img src="st766.gif"` is used to link a custom image.

2.3 Menu frame – menu.html

Figure 3: Menu frame



Customize your Dynamic Web Pages!

Practice with module SSIs

- MODULE CONFIG/STAT/PEERS: open the configuration variables and the status parameters of the module
- INPUT DEMO: send your text from the module to your dynamic web page
- OUTPUT DEMO: send text from your web page to the module
- REMOTE CONFIGURATION: write/config/read a GPIO of the module, wake/sleep the module when the SLEEP mode is enabled, remotely use ADC/DAC/PWM, remotely set a variable
- SPWF01S FIRST SET: configure remotely the module to enable an available use mode (IDLE, STA, Mini AP)

2.3.1 HTML code

```
<!DOCTYPE html>
<html>
<head>
</head>


<body bgcolor="#F2F2F2">


<span style="background-color: #f3f3f3;"><span style="color: #20124d;"><i><b><span
style="font-size: small;"><span style="font-family: &quot;Courier
New&quot;,Courier,monospace;">SPG Application
</span></span></b></i></span></span>
<br>
<script language="JavaScript1.2">


function digitalclock(){
var data =new Date()
mese = data.getMonth();
giorno =data.getDay();
anno = data.getFullYear();
date = data.getDate();
ora = data.getHours();
minuti=data.getMinutes();

secondi=data.getSeconds();

if(giorno == 0)giorno = "Domenica";

if(giorno == 1)giorno = "Luned\ì";

if(giorno == 2)giorno = "Marted\ì";

if(giorno == 3)giorno = "Mercoled\ì";

if(giorno == 4)giorno = "Gioved\ì";

if(giorno == 5)giorno = "Venerd\ì";

if(giorno == 6)giorno = "Sabato";

if(mese == 0)mese = "Gennaio";

if(mese ==1)mese = "Febbraio";

if(mese ==2)mese = "Marzo";

if(mese ==3)mese = "Aprile";

if(mese ==4)mese = "Maggio";

if(mese ==5)mese = "Giugno";

if(mese ==6)mese = "Luglio";

if(mese ==7)mese = "Agosto";

if(mese ==8)mese = "Settembre";

if(mese ==9)mese = "Ottobre";

if(mese ==10)mese = "Novembre";

if(mese ==11)mese = "Dicembre";

if (ora <=9)ora="0"+ora;

if (minuti<=9)minuti="0"+minuti;

if (secondi<=9)secondi="0"+secondi;

data="<font size='2' face='Verdana'>"+giorno+" "+date+" "+mese+" "+anno+"
"+ora+":"+minuti+":"+secondi+"</font>"

if (document.all)

document.all.clock.innerHTML=data;

else if (document.getElementById)

document.getElementById("clock").innerHTML=data;

else

document.write(data)

}

if (!document.all&&!document.getElementById)

digitalclock()

function loadtime(){

if (document.all||document.getElementById)

setInterval("digitalclock()",1000)

}

onload=loadtime

// -->

</script>

</head>
```

```
<body>
<span id="clock"></span>


<br><br><br><br><br><br>
<body>
<a href="/home.html" target="homepage"><button style="
width: 15em;
padding: .5em;
color: #ffffff;
text-shadow: 1px 1px 1px #000000;
border: solid thin #1c58ce;
-webkit-border-radius: .7em;
-moz-border-radius: .7em;
border-radius: .7em;
-webkit-box-shadow: 2px 2px 3px #999999;
box-shadow: 2px 2px 2px #bbbbbb;
background-color: #1c58ce;
background-image: -webkit-gradient(linear, left top, left bottom,
from(#2bb4ff), to(#0070ad),color-stop(0.4, #102d65));">HOME</button></a>
</body>
<br><br><br>

<body>
<a href="/configuration.html" target="homepage"><button style="
width: 15em;
padding: .5em;
color: #ffffff;
text-shadow: 1px 1px 1px #000000;
border: solid thin #1c58ce;
-webkit-border-radius: .7em;
-moz-border-radius: .7em;
border-radius: .7em;
-webkit-box-shadow: 2px 2px 3px #999999;
box-shadow: 2px 2px 2px #bbbbbb;
background-color: #1c58ce;
background-image: -webkit-gradient(linear, left top, left bottom,
from(#2bb4ff), to(#0070ad),color-stop(0.4, #102d65));">MODULE
CONFIG/STAT/PEERS</button></a>
</body>
<br><br><br>


<body>
```

```
<a href="/input_demo.shtml" target="homepage"><button style="
width: 15em;
padding: .5em;
color: #ffffff;
text-shadow: 1px 1px 1px #000000;
border: solid thin #1c58ce;
-webkit-border-radius: .7em;
-moz-border-radius: .7em;
border-radius: .7em;
-webkit-box-shadow: 2px 2px 3px #999999;
box-shadow: 2px 2px 2px #bbbbbb;
background-color: #1c58ce;
background-image: -webkit-gradient(linear, left top, left bottom,
from(#2bb4ff), to(#0070ad),color-stop(0.4, #102d65));">INPUT DEMO</button></a>
</body>
<br><br>


<body>
<a href="/output_demo.html" target="homepage"><button style="
width: 15em;
padding: .5em;
color: #ffffff;
text-shadow: 1px 1px 1px #000000;
border: solid thin #1c58ce;
-webkit-border-radius: .7em;
-moz-border-radius: .7em;
border-radius: .7em;
-webkit-box-shadow: 2px 2px 3px #999999;
box-shadow: 2px 2px 2px #bbbbbb;
background-color: #1c58ce;
background-image: -webkit-gradient(linear, left top, left bottom,
from(#2bb4ff), to(#0070ad),color-stop(0.4, #102d65));">OUTPUT DEMO</button></a>
</body>
<br><br><br>


<body>
<a href="/remote.html" target="homepage"><button style="
width: 15em;
padding: .5em;
color: #ffffff;
text-shadow: 1px 1px 1px #000000;
```

```
border: solid thin #1c58ce;
-webkit-border-radius: .7em;
-moz-border-radius: .7em;
border-radius: .7em;
-webkit-box-shadow: 2px 2px 3px #999999;
box-shadow: 2px 2px 2px #bbbbbb;
background-color: #1c58ce;
background-image: -webkit-gradient(linear, left top, left bottom,
from(#2bb4ff), to(#0070ad),color-stop(0.4, #102d65));"> REMOTE
CONFIGURATION</button></a>
</body>
<br><br><br>
<body>
<a href="/firstset.html" target="homepage"><button style="
width: 15em;
padding: .5em;
color: #ffffff;
text-shadow: 1px 1px 1px #000000;
border: solid thin #1c58ce;
-webkit-border-radius: .7em;
-moz-border-radius: .7em;
border-radius: .7em;
-webkit-box-shadow: 2px 2px 3px #999999;
box-shadow: 2px 2px 2px #bbbbbb;
background-color: #1c58ce;
background-image: -webkit-gradient(linear, left top, left bottom,
from(#2bb4ff), to(#0070ad),color-stop(0.4, #102d65));">SPWF01S FIRST
SET</button></a>
</body>
</html>
```

2.3.2 HTML code description

The Menu frame is divided into two sections. The first displays a sample digital clock and date (marked in blue). A second displays six HTML5 buttons that cover all the HTML sub-pages.

To correctly perform the DEMO, it is recommended to use a browser that is compatible with the HTML5 standard (i.e. Google Chrome). The HTML5 buttons can be replaced with basic buttons if full HTML compatibility is required.

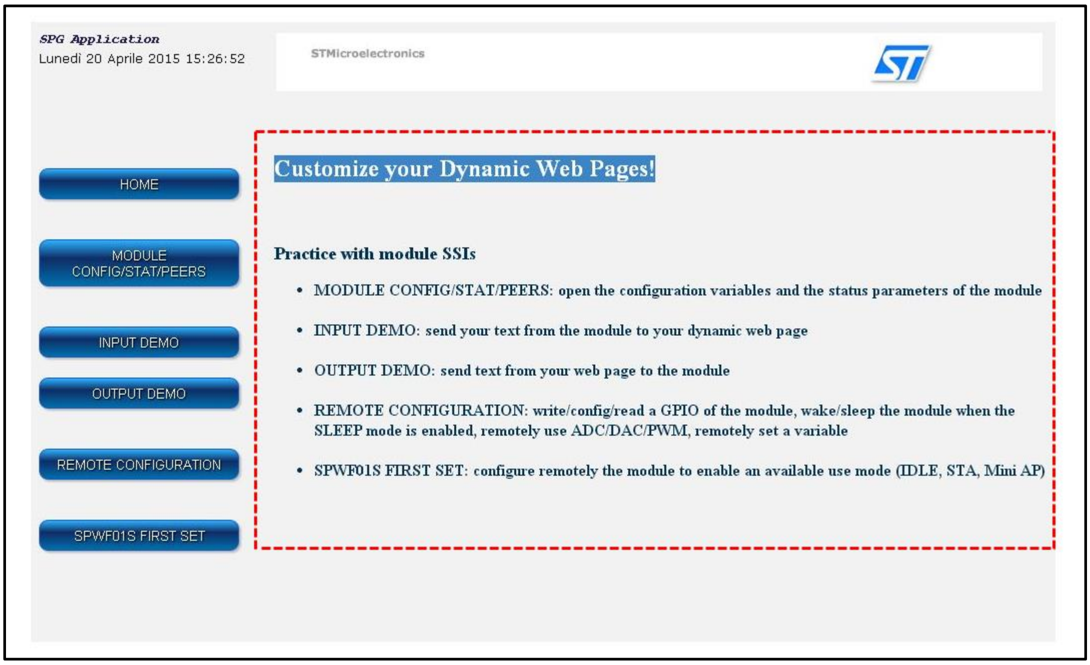
One of the HTML5 buttons is shown below, in detail:

```
<body>
<a href="/home.html" target="homepage"><button style="
width: 15em;
padding: .5em;
color: #ffffff;
text-shadow: 1px 1px 1px #000000;
border: solid thin #1c58ce;
-webkit-border-radius: .7em;
-moz-border-radius: .7em;
border-radius: .7em;
-webkit-box-shadow: 2px 2px 3px #999999;
box-shadow: 2px 2px 2px #bbbbbb;
background-color: #1c58ce;
background-image: -webkit-gradient(linear, left top, left bottom,
from(#2bb4ff), to(#0070ad),color-stop(0.4, #102d65));">HOME</button></a>
</body>
```

The code `<a href="/home.html" target="homepage">` can be used to open a web page in a target frame. This code allows users to open the home.html page in the main frame.

2.4 HOME page (main frame) – home.html

Figure 4: Home page - main frame



SPG Application

Lunedì 20 Aprile 2015 15:26:52

Customize your Dynamic Web Pages!

Practice with module SSIs

- MODULE CONFIG/STAT/PEERS: open the configuration variables and the status parameters of the module
- INPUT DEMO: send your text from the module to your dynamic web page
- OUTPUT DEMO: send text from your web page to the module
- REMOTE CONFIGURATION: write/config/read a GPIO of the module, wake/sleep the module when the SLEEP mode is enabled, remotely use ADC/DAC/PWM, remotely set a variable
- SPWF01S FIRST SET: configure remotely the module to enable an available use mode (IDLE, STA, Mini AP)

2.4.1 HTML code

```
<html>

<head>

<title>HOME</title>

</head>

<body bgcolor="#F2F2F2">

<span style="color: white; background-color: #3d85c6; font-size: x-large;"text-
align: center;"><b>Customize your Dynamic Web Pages!</b></span><br>

<br><br><br>

<span style="font-size: large;"><b><span style="color: #003153;"><span style="font-
size: large;">Practice with module SSIs</span></span></b></span><br>

<span style="font-size: large;"><b><span style="color: #003153;"><span style="font-
size: medium;"><ul><li>MODULE CONFIG/STAT/PEERS: open the configuration variables
and the status parameters of the module<br><br><li>INPUT DEMO: send your text from
the module to your dynamic web page<br><br><li>OUTPUT DEMO: send text from your web
page to the module<br><br><li>REMOTE CONFIGURATION: write/config/read a GPIO of the
module, wake/sleep the module when the SLEEP mode is enabled, remotely use
ADC/DAC/PWM, remotely set a variable<br><br><li>SPWF01S FIRST SET: configure
remotely the module to enable an available use mode (IDLE, STA, Mini
AP)</ul></span></span></b></span>

</html>
```

2.4.2 HTML code description

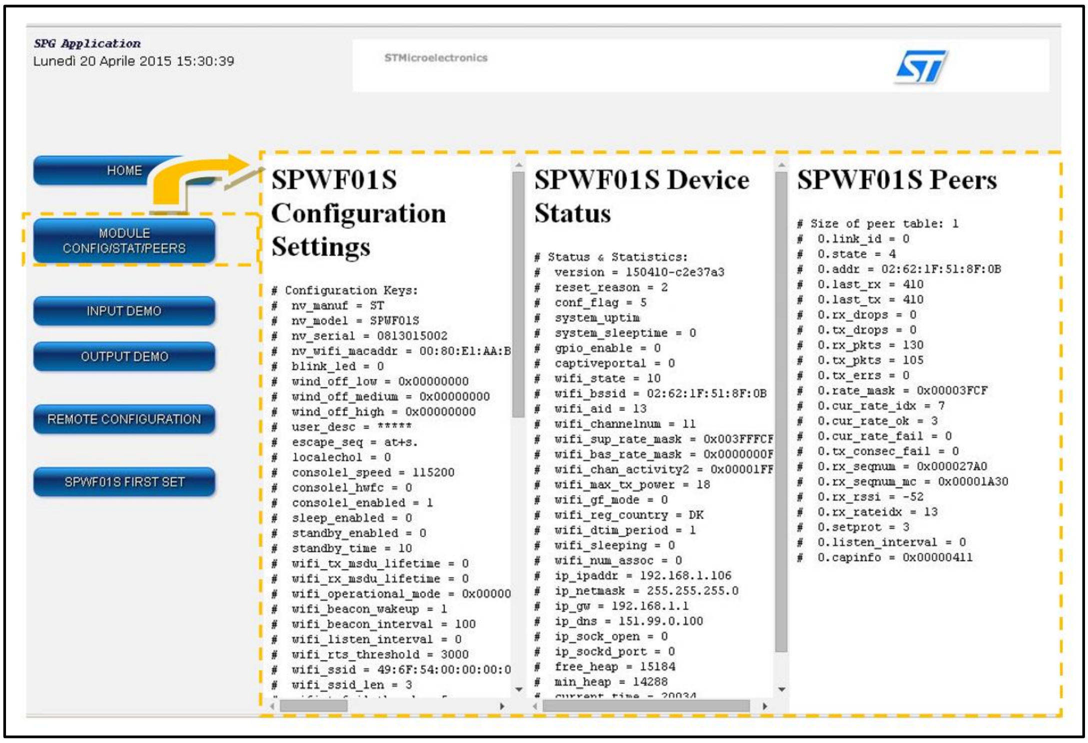
The HOME frame performs two functions:

- displays the home.html page when index.html is requested
- displays the HTML page when a button is clicked

The home.html page (included in the package) describes the content of this DEMO and can be freely modified.

2.5 MODULE CONFIG/STAT/PEERS PAGE – configuration.html

Figure 5: Configuration page



SPG Application
Lunedì 20 Aprile 2015 15:30:39

SPWF01S Configuration Settings

```
# Configuration Keys:
#  nv_manuf = ST
#  nv_model = SPWF01S
#  nv_serial = 0813015002
#  nv_wifi_macaddr = 00:80:E1:AA:B
#  blink_led = 0
#  wind_off_low = 0x00000000
#  wind_off_medium = 0x00000000
#  wind_off_high = 0x00000000
#  user_desc = *****
#  escape_seq = at+s.
#  localechol = 0
#  consolel_speed = 115200
#  consolel_hwfc = 0
#  consolel_enabled = 1
#  sleep_enabled = 0
#  standby_enabled = 0
#  standby_time = 10
#  wifi_tx_msdu_lifetime = 0
#  wifi_rx_msdu_lifetime = 0
#  wifi_operational_mode = 0x00000
#  wifi_beacon_wakeup = 1
#  wifi_beacon_interval = 100
#  wifi_listen_interval = 0
#  wifi_rts_threshold = 3000
#  wifi_ssid = 49:6F:54:00:00:00:0
#  wifi_ssid_len = 3
```

SPWF01S Device Status

```
# Status & Statistics:
#  version = 150410-c2e37a3
#  reset_reason = 2
#  conf_flag = 5
#  system_uptim
#  system_sleeptime = 0
#  gpio_enable = 0
#  captiveportal = 0
#  wifi_state = 10
#  wifi_bssid = 02:62:1F:51:8F:0B
#  wifi_aid = 13
#  wifi_channelnum = 11
#  wifi_sup_rate_mask = 0x003FFFCF
#  wifi_bas_rate_mask = 0x0000000F
#  wifi_chan_activity2 = 0x00001FF
#  wifi_max_tx_power = 18
#  wifi_gf_mode = 0
#  wifi_reg_country = DK
#  wifi_dtim_period = 1
#  wifi_sleeping = 0
#  wifi_num_assoc = 0
#  ip_ipaddr = 192.168.1.106
#  ip_netmask = 255.255.255.0
#  ip_gw = 192.168.1.1
#  ip_dns = 151.99.0.100
#  ip_sock_open = 0
#  ip_sockd_port = 0
#  free_heap = 15184
#  min_heap = 14288
#  current_time = 20034
```

SPWF01S Peers

```
# Size of peer table: 1
#  0.link_id = 0
#  0.state = 4
#  0.addr = 02:62:1F:51:8F:0B
#  0.last_rx = 410
#  0.last_tx = 410
#  0.rx_drops = 0
#  0.tx_drops = 0
#  0.rx_pkts = 130
#  0.tx_pkts = 105
#  0.tx_errs = 0
#  0.rate_mask = 0x00003FCF
#  0.cur_rate_idx = 7
#  0.cur_rate_ok = 3
#  0.cur_rate_fail = 0
#  0.tx_consec_fail = 0
#  0.rx_seqnum = 0x000027A0
#  0.rx_seqnum_mc = 0x00001A30
#  0.rx_rssi = -52
#  0.rx_rateidx = 13
#  0.setprot = 3
#  0.listen_interval = 0
#  0.capinfo = 0x00000411
```

2.5.1 HTML code – configuration.html

```
<html>
<head>
<title>MODULE CONFIGURATION</title>
</head>
<frameset cols="33%, 33%,*" frameborder="0" framespacing="0" border="0">
         <frame name="config" src="config.shtml" scrolling="auto" framespacing="0">
                    <frame name="stat" src="status.shtml" scrolling="auto"
framespacing="0">
         <frame name="peers" src="peers.shtml" scrolling="auto" framespacing="0">
     </frameset>
<body>
<noframes>Your browser does not support frames.</noframes>
</body>
</html>
```

2.5.2 HTML code description – configuration.html

The configuration.html page allows user to create two frames that link to the “config.shtml” page and “status.shtml” page.

These web pages are stored in the internal Flash and can be customized and overwritten.

2.5.3 HTML code – config.shtml

```
<html>
<head><title>ST SPWF01S Configuration Settings</title></head>
<body bgcolor="white" text="black">
<h1>SPWF01S Configuration Settings</h1>
<pre>
<!--#devconf-->
</pre>
</body>
</html>
```

2.5.4 HTML code description – config.shtml

The config.shtml page contains the “***devconf***” SSI tag.

Using this tag in a **custom.shtml** page, the user can obtain the module's configuration variables in real-time.

2.5.5 HTML code – status.shtml

```
<html>
<head><title>ST SPWF01S Device Status</title></head>
<body bgcolor="white" text="black">
<h1>SPWF01S Device Status</h1>
<pre>
<!--#devsts-->
</pre>
</body>
</html>
```

2.5.6 HTML code description – status.shtml

The status.shtml page contains the “***devsts***” SSI tag. Using this tag in a **custom.shtml** page, the user can obtain the module's status/statistics in real-time.

2.5.7 HTML code – peers.shtml

```
<html>
<head><title>ST SPWF01S Peers</title></head>
<body bgcolor="white" text="black">
<h1>SPWF01S Peers</h1>
<pre>
<!--#peers-->
</pre>
</body>
</html>
```

2.5.8 HTML code description – peers.shtml

The peers.shtml page contains the "***peers***" SSI tag. Using this tag in a **custom.shtml** page, the user can obtain the module's peers info in real-time.

Note: a new SSI devsts/devconf/peers optimization has been added in the FW 3.4.

```
<!--#x:y--></p> - "x" can be "devsts" or "devconf", "y" is status/configuration
variable name

<!--#x:n:y--></p> - "x" is "peers", "n" is a number (0 to 5), and "y" is peer status
variable name
```

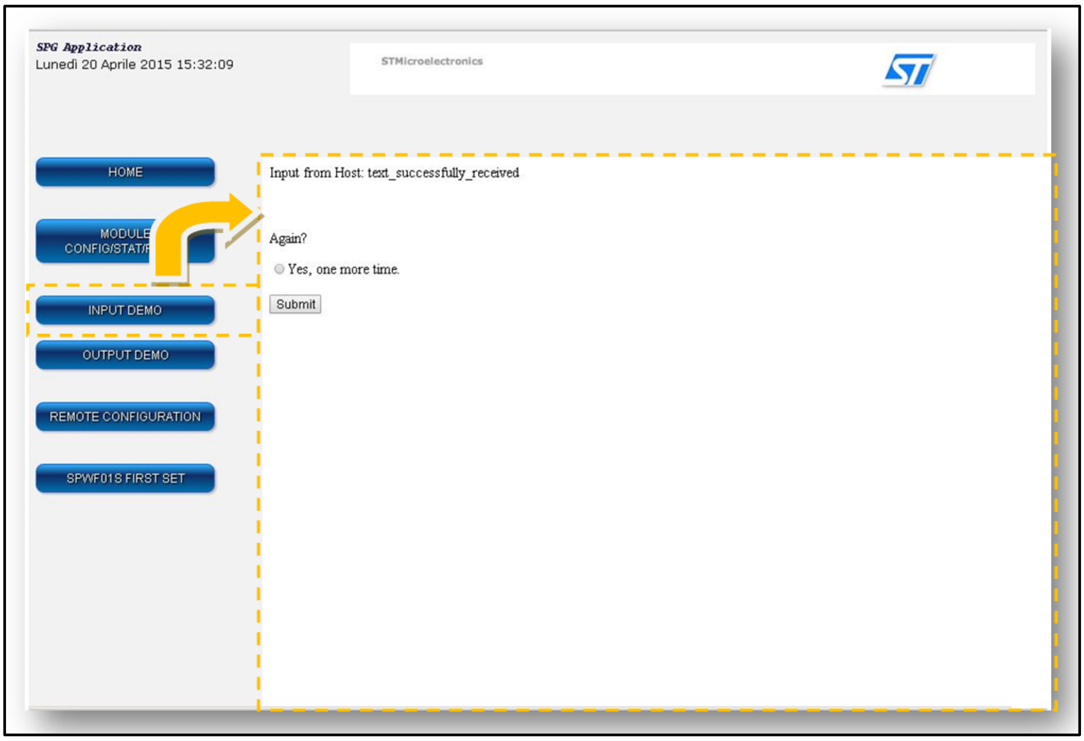
For example:

```
<!--#devsts:wifi_state--></p>

<!--#devconf:user_desc--></p>

<!--#peers:1:link_id--></p>
```

It provides the status wifi_state variable, the config user_desc variable and the link_id of the peers No.1.

2.6 INPUT DEMO – input_demo.shtml

Figure 6: INPUT DEMO page



Input from Host: text_successfully_received

Again?

○ Yes, one more time.

2.6.1 HTML code

```
<html>
<head><title>Input Demo</title></head>
<body bgcolor="white" text="black">
       <p>Input from Host: <!--#input--> </p>

       <p>
       <br/><br/>Again?<br/>
       <form name="CGI Uart" method="GET" action="/input.cgi">
       <input type="radio" name="input" value="on"/>Yes, one more time.<br/>
         <br>
               <input type="submit" name="submit" value="Submit" >
</form>
</p>
</body>
</html>
```

2.6.2 HTML code description

The input_demo.shtml page included in the package is a replacement of the input_demo.shtml stored in the internal Flash memory of the module.

This page contains both the SSI tag and CGI.

The first asks the HOST to provide a message for client, while the second replaces the HTML comment (SSI) with this HOST's message.

Note: a new input_demo optimization has been added to the FW ver. 3.4.

- the SSI `<!--#input-->` provides an anonymous "wind:56:..." indication
- the SSI `<!--#input:xxxx-->` provides a "wind:56:xxxx" indication
- If additional colons are inserted in the SSI string (i.e. `<!--#input:xxxx:demo:-->`), they will be skipped and inserted in the "wind:56:..." indication (i.e. "wind:56:xxxx:demo:")

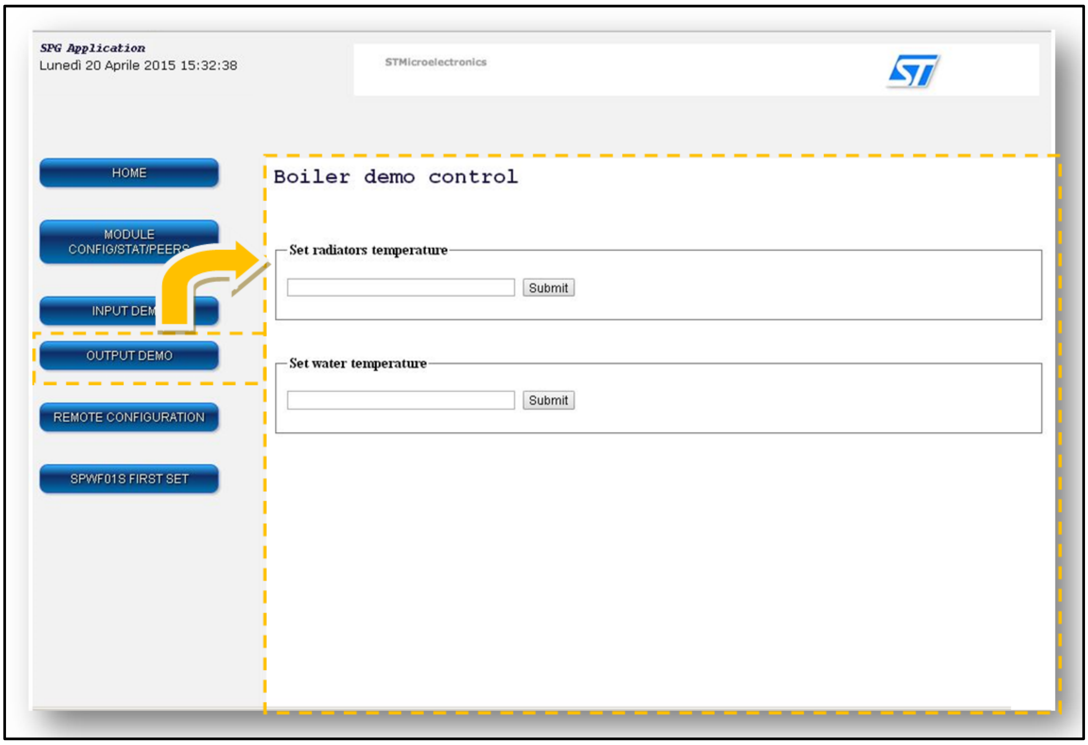
This feature allows the external host to discriminate between different clients performing the request.

VERY IMPORTANT

- Both client and server are pending (2 sec) after the page request. If no message is sent from the module, a timeout mechanism will be triggered.
- The SSI tag is triggered on the carriage return; therefore the message terminator must be `<CR>`.

2.7 OUTPUT DEMO – output_demo.html

Figure 7: OUTPUT DEMO page



2.7.1 HTML code

```
<html>
<head><title>Boiler demo control</title></head>
<body bgcolor="white" text="black">
<span style="background-color: ;"><span style="color: #20124d;"><b><span
style="font-size: x-large;"><span style="font-family: &quot;Courier
New&quot;,Courier,monospace;">Boiler demo
control</span></span></b></i></span></span>
<br><br><br><br>
<fieldset>
<legend><h4>Set radiators temperature</h4></legend>
        <p><form name="CGI Example" method="GET" action="output.cgi" >
        <input type="text" name="text" size="40" maxlength="40">&nbsp;
           <input type="submit" name="submit" value="Submit" >
          </form>
        </p>
</fieldset>
<br><br>
<fieldset>
<legend><h4>Set water temperature</h4></legend>
        <p><form name="CGI Example" method="GET" action="output.cgi" >
        <input type="text" name="text" size="40" maxlength="40">&nbsp;
```

```
            <input type="submit" name="submit" value="Submit" >
          </form>
        </p>
</fieldset>
</body>
</html>
```

2.7.2 HTML code description

The output_demo.html included in the package is a replacement of the output_demo.html stored in the internal Flash memory.

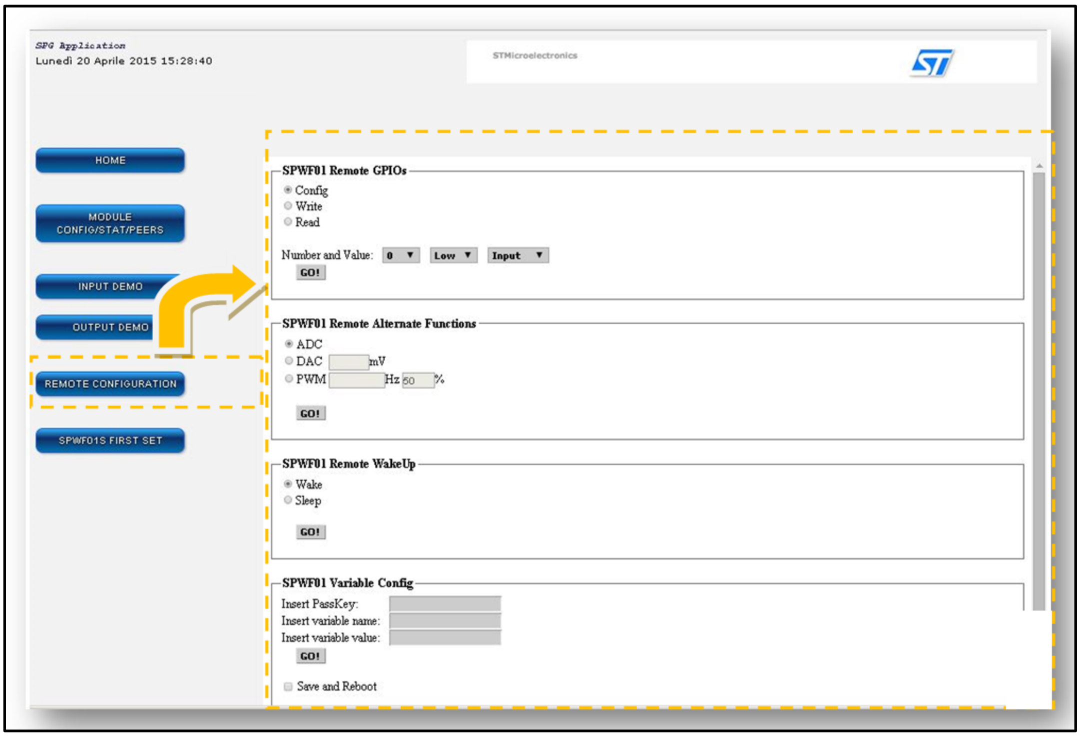
The user can customize the output_demo.html in order to create a dynamic page to send data to the module.

The output_demo.html contains the *output.cgi* link that allows the user to push data over the module's UART. So, all of the text put into the box will be sent to the UART after the submit action.

Please note that *output.cgi* always redirects to the “output_demo.html” as a last action. See the Appendix for a set of examples to build different output_demo pages.

2.8 REMOTE CONFIGURATION – remote.html

Figure 8: REMOTE CONFIGURATION page



SPG Application
Lunedì 20 Aprile 2015 15:28:40

SPWF01 Remote GPIOs

● Config
○ Write
○ Read

Number and Value: 0 ▼ Low ▼ Input ▼

SPWF01 Remote Alternate Functions

● ADC
○ DAC mV
○ PWM Hz 50 %

SPWF01 Remote WakeUp

● Wake
○ Sleep

SPWF01 Variable Config

Insert PassKey:
Insert variable name:
Insert variable value:

□ Save and Reboot

2.8.1 HTML code

```
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
  <title>remote</title>
</head>
<body>
<script>
function setVar()
{
if(document.getElementById("cbox").checked)
  {alert("Press OK to send, then wait a few seconds.")}
}
function setC()
{
document.getElementById("LoI").disabled=true;
document.getElementById("IoO").disabled=false;
}
function setW()
{
document.getElementById("LoI").disabled=false;
document.getElementById("IoO").disabled=true;
}
function setR()
{
document.getElementById("LoI").disabled=true;
document.getElementById("IoO").disabled=true;
}
function setA()
{
document.getElementById("vlt").disabled=true;
document.getElementById("frq").disabled=true;
document.getElementById("dc").disabled=true;
}
function setD()
{
document.getElementById("vlt").disabled=false;
document.getElementById("frq").disabled=true;
document.getElementById("dc").disabled=true;
}
function setP()
{
```

```
document.getElementById("vlt").disabled=true;
document.getElementById("frq").disabled=false;
document.getElementById("dc").disabled=false;
}
</script>
<fieldset>
<legend><h4>SPWF01 Remote GPIOs</h4></legend>
<form enctype="text/plain" method="get" action="remote.cgi" name="confiGPIOs">
<input type="radio" name="OpName" value="CONFIG" onclick="setC()"
checked/>Config<br/>
<input type="radio" name="OpName" value="WRITE" onclick="setW()"/>Write<br/>
<input type="radio" name="OpName" value="READ" onclick="setR()"/>Read<br/><br/>
<table>
<tr>
<td>Number and Value: </td>
<td> <select name="Number" style="margin-left: 10px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;">
  <option value="0">0</option>
  <option value="1">1</option>
  <option value="2">2</option>
  <option value="3">3</option>
  <option value="4">4</option>
  <option value="5">5</option>
  <option value="6">6</option>
  <option value="7">7</option>
  <option value="8">8</option>
  <option value="9">9</option>
  <option value="10">10</option>
  <option value="11">11</option>
  <option value="12">12</option>
  <option value="13">13</option>
  <option value="14">14</option>
  <option value="15">15</option>
  </select></td>
<td> <select disabled id="LoI" name="Value" style="margin-left: 10px; color:
#000000; font-family: Verdana; font-weight: bold; font-size: 12px; background-color:
#cccccc;">
  <option value="0">Low</option>
  <option value="1">High</option>
  </select></td>
<td> <select id="IoO" name="Direction" style="margin-left: 10px; color: #000000;
font-family: Verdana; font-weight: bold; font-size: 12px; background-color:
#cccccc;">
  <option value="0">Input</option>
  <option value="1">Output</option>
  </select></td>
</tr>
</table>
  <input type="submit" name="set" value="GO!" style="margin-left: 20px; color:
#000000; font-family: Verdana; font-weight: bold; font-size: 12px; background-color:
#cccccc;">
</fieldset>
</form>
<br/>
<fieldset>
<legend><h4>SPWF01 Remote Alternate Functions</h4></legend>
<form enctype="text/plain" method="get" action="remote.cgi" name="confiAFs">
<table>
<tr><td><input type="radio" name="OpName" value="ADC" onclick="setA()"
checked/>ADC<br/></td></tr>
<tr><td><input type="radio" name="OpName" value="DAC"
onclick="setD()"/>DAC<br/></td>
<td><input type="text" name="voltage" id="vlt" size="4" disabled="true">mV</td></tr>
<tr><td><input type="radio" name="OpName" value="PWM"
onclick="setP()"/>PWM<br/></td>
<td><input type="text" name="frequency" id="frq" size="8" disabled="true">Hz</td>
<td><input type="text" name="dcycle" id="dc" size="2" value="50"
disabled="true">%</td></tr>
</table>
  <br/>
  <input type="submit" name="AFset" value="GO!" style="margin-left: 20px; color:
#000000; font-family: Verdana; font-weight: bold; font-size: 12px; background-color:
#cccccc;">
</fieldset>
</form>
<br/>
<fieldset>
<legend><h4>SPWF01 Remote WakeUp</h4></legend>
<form enctype="text/plain" method="get" action="remote.cgi" name="configPMs">
<input type="radio" name="OpName" value="WAKE" checked/>Wake<br/>
<input type="radio" name="OpName" value="SLEEP"/>Sleep<br/><br/>
<input type="submit" name="change" value="GO!" style="margin-left: 20px; color:
#000000; font-family: Verdana; font-weight: bold; font-size: 12px; background-color:
#cccccc;">
</fieldset>
</form>
<br/>
<fieldset>
<legend><h4>SPWF01 Variable Config</h4></legend>
<form enctype="text/plain" method="get" action="remote.cgi" name="variables">
<table>
<tr>
```

```
<td>Insert PassKey:</td>

<td><input type="password" style="margin-left: 10px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;"
name="KEY"></td>

</tr>

<tr>

<td>Insert variable name:</td>

<td><input type="text" style="margin-left: 10px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;"
name="VAR"></td>

</tr>

<tr>

<td>Insert variable value:</td>

<td><input type="text" style="margin-left: 10px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;"
name="VAL"></td>

</tr>

</table>

  <input type="submit" name="SCFG" value="GO!" onclick="setVar()" style="margin-
left: 20px; color: #000000; font-family: Verdana; font-weight: bold; font-size:
12px; background-color: #cccccc;">

  <br/><br/>

  <input type="checkbox" name="box" id="cbox"> Save and Reboot

</fieldset>

</form>

</body>

</html>
```

2.8.2 HTML code description

The remote.html page reported in this AN is stored in the internal Flash of the module.

In any case, the user can build a new remote.html page using the code provided in the previous section.

In this page, the remote.cgi is used which allows users to perform GPIO control (write, config and read) as described in the user manual

In detail:

The script section (in blue) allows management of the multiple-choice boxes to conveniently enable/disable their functioning.

The main section contains the HTML code to create a whole form.

- The remote.cgi page must be included in the HTML form:

```
      <form enctype="text/plain" method="get" action="remote.cgi" name="configure">
```

- The following HTML code allows management of the available actions (config, write, read):

```
      <input type="radio" name="OpName" value="CONFIG" onclick="setC()"
checked/>Config<br/>
```

```
      <input type="radio" name="OpName" value="WRITE" onclick="setW()"/>Write<br/>


      <input type="radio" name="OpName" value="READ"
onclick="setR()"/>Read<br/><br/>
```

The user can customize the “input type” field (radio) and the “name” field (OpName). The “**value**” field (**CONFIG, WRITE and READ**) cannot be modified to correctly run the CGI script.

- The following code allows creating the multiple choice boxes (GPIO number to control, value or direction)

```
<table>
<tr>
<td>Number and Value: </td>
<td> <select name="Number" style="margin-left: 10px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;">
  <option value="0">0</option>
  <option value="1">1</option>
  <option value="2">2</option>
  <option value="3">3</option>
  <option value="4">4</option>
  <option value="5">5</option>
  <option value="6">6</option>
  <option value="7">7</option>
  <option value="8">8</option>
  <option value="9">9</option>
  <option value="10">10</option>
  <option value="11">11</option>
  <option value="12">12</option>
  <option value="13">13</option>
  <option value="14">14</option>
  <option value="15">15</option>
  </select></td>
<td> <select disabled id="LoI" name="Value" style="margin-left: 10px; color:
#000000; font-family: Verdana; font-weight: bold; font-size: 12px; background-color:
#cccccc;">
  <option value="0">Low</option>
  <option value="1">High</option>
  </select></td>
<td> <select id="IoO" name="Direction" style="margin-left: 10px; color: #000000;
font-family: Verdana; font-weight: bold; font-size: 12px; background-color:
#cccccc;">
  <option value="0">Input</option>
  <option value="1">Output</option>
  </select></td>
</tr>
```

```
</table>
```

- The following code allows the user to correctly send his choice to the module

```
<input type="submit" name="set" value="GO!" style="margin-left: 20px; color:
#000000; font-family: Verdana; font-weight: bold; font-size: 12px; background-color:
#cccccc;">
```

The "**name**" field (**set**) cannot be modified to correctly run the CGI script.

Note: the CGI script checks the number of parameters added in the HTML page (READ: 2 parameters, WRITE/CONFIG: 3 parameters); therefore it is very important respect the workflow described above.

The same considerations can be applied for the remote ADC/DAC/PWM, Wake/Sleep and Variable config HTML code.

GPIO CONTROL in-a-click

The remote.cgi page can also be used by a single HTTP GET from the HOST.

READ a GPIO

[Module IP Address]/remote.cgi?OpName=READ&Number=[GPIO to be read]&set=GO

i.e. ***http://192.168.1.100/remote.cgi?OpName=READ&Number=5&set=GO***

WRITE a GPIO

[Module IP Address]/remote.cgi?OpName=WRITE&Number=[GPIO to be wrote]&Value=[0 for Low, 1 for High]&set=GO

i.e. ***http://192.168.1.100/remote.cgi?OpName=WRITE&Number=8&Value=1&set=GO***

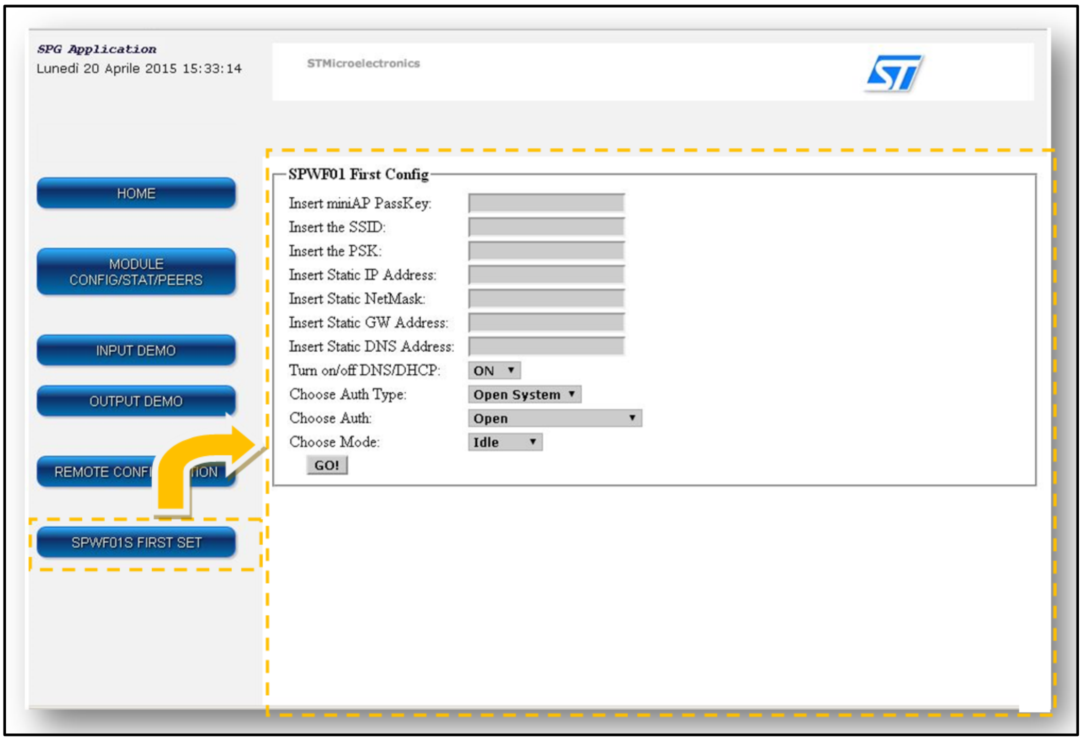
CONFIG a GPIO

[Module IP Address]/remote.cgi?OpName=CONFIG&Number=[GPIO to be configured]&Direction=[0 for Input, 1 for Output]&set=GO

i.e.
http://192.168.1.100/remote.cgi?OpName=CONFIG&Number=7&Direction=1&set=GO

2.9 SPWF01S FIRST SET – firstset.html

Figure 9: SPWF01S FIRST SET page



SPWF01 First Config

Insert miniAP PassKey:
Insert the SSID:
Insert the PSK:
Insert Static IP Address:
Insert Static NetMask:
Insert Static GW Address:
Insert Static DNS Address:
Turn on/off DNS/DHCP: ON
Choose Auth Type: Open System
Choose Auth: Open
Choose Mode: Idle

2.9.1 HTML code

```
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
  <title>firstset</title>
</head>
<body>
<form enctype="text/plain" method="get" action="firstset.cgi" name="configure">
<fieldset>
<legend><h4>SPWF01 First Config</h4></legend>
<table>
<tr>
<td>Insert miniAP PassKey:</td>
<td><input type="password" style="margin-left: 10px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;"
name="KEY"></td>
</tr>
<tr>
<td>Insert the SSID:</td>
<td><input type="text" style="margin-left: 10px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;"
name="SSID"></td>
</tr>
```

```
<tr>
<td>Insert the PSK:</td>
<td><input type="password" style="margin-left: 10px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;"
name="PWD"></td>
</tr>
<tr>
<td>Insert Static IP Address:</td>
<td><input type="text" style="margin-left: 10px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;"
name="IP"></td>
</tr>
<tr>
<td>Insert Static NetMask:</td>
<td><input type="text" style="margin-left: 10px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;"
name="MASK"></td>
</tr>
<tr>
<td>Insert Static GW Address:</td>
<td><input type="text" style="margin-left: 10px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;"
name="GW"></td>
</tr>
<tr>
<td>Insert Static DNS Address:</td>
<td><input type="text" style="margin-left: 10px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;"
name="DNS"></td>
</tr>
<tr>
<td>Turn on/off DNS/DHCP:</td>
<td> <select name="Dhcp" style="margin-left: 10px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;">
  <option value="0">OFF</option>
  <option value="1" selected>ON</option>
  </select></td>
</tr>
<tr>
<td>Choose Auth Type:</td>
<td> <select name="IbssAuth" style="margin-left: 10px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;">
  <option value="0">Open System</option>
  <option value="1">Shared Key</option>
  </select></td>
</tr>
<tr>
```

```
<td>Choose Auth:</td>
<td> <select name="Auth" style="margin-left: 10px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;">
  <option value="0">Open</option>
  <option value="1">Wep</option>
  <option value="2">Wpa & Wpa2 Personal</option>
  <!--<option value="3">Wpa Enterprise</option>-->
  </select></td>
</tr>
<tr>
<td>Choose Mode:</td>
<td> <select name="Mode" style="margin-left: 10px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;">
  <option value="0">Idle</option>
  <option value="1">Station</option>
  <option value="2">IBSS</option>
  <option value="3">Mini AP</option>
  </select></td>
</tr>
</table>
  <input type="submit" name="confirm" value="GO!" onclick="alert('Press OK to send,
then wait a few seconds.')" style="margin-left: 20px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;">
</fieldset>
</form>
</body>
</html>
```

2.9.2 HTML code description

The *firstset.cgi* allows also performing the first configuration of the SPWF01S Module.

Following the HTML code reported above, it is necessary to observe these steps:

- The firstset.cgi page must be included in the HTML form

```
<form enctype="text/plain" method="get" action=" firstset.cgi" name="configure">
```

- Insert the 1st parameter (KEY)

```
<td>Insert miniAP PassKey:</td>
<td><input type="password" style="margin-left: 10px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;"
name="KEY"></td>
```

The “**name**” field (**KEY**) cannot be modified to correctly run the CGI script.

- Insert the 2nd parameter (SSID)

```
<td>Insert the SSID:</td>
<td><input type="text" style="margin-left: 10px; color: #000000; font-family:
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;"
name="SSID"></td>
```

The “**name**” field (**SSID**) cannot be modified to correctly run the CGI script.

- Insert the 3rd parameter (PWD)

```
<td>Insert the PSK:</td>

<td><input type="password" style="margin-left: 10px; color: #000000; font-family: 
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;" 
name="PWD"></td>
```

The “**name**” field (**PWD**) cannot be modified to correctly run the CGI script.

- Insert the 4th parameter (IP)

```
<td>Insert Static IP Address:</td>

<td><input type="text" style="margin-left: 10px; color: #000000; font-family: 
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;" 
name="IP"></td>
```

The “**name**” field (**IP**) cannot be modified to correctly run the CGI script.

- Insert the 5th parameter (MASK)

```
<td>Insert Static NetMask:</td>

<td><input type="text" style="margin-left: 10px; color: #000000; font-family: 
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;" 
name="MASK"></td>
```

The “**name**” field (**MASK**) cannot be modified to correctly run the CGI script.

- Insert the 6th parameter (GW)

```
<td>Insert Static GW Address:</td>

<td><input type="text" style="margin-left: 10px; color: #000000; font-family: 
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;" 
name="GW"></td>
```

The “**name**” field (**GW**) cannot be modified to correctly run the CGI script.

- Insert the 7th parameter (DNS)

```
<td>Insert Static DNS Address:</td>

<td><input type="text" style="margin-left: 10px; color: #000000; font-family: 
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;" 
name="DNS"></td>
```

The “name” field (DNS) cannot be modified to correctly run the CGI script.

- Insert the 8th parameter (IbssAuth)

```
<td>Choose Auth Type:</td>

<td> <select name="IbssAuth" style="margin-left: 10px; color: #000000; font-family: 
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;">

  <option value="0">Open System</option>

  <option value="1">Shared Key</option>

  </select></td>
```

The “**name**” field (**IbssAuth**) cannot be modified to correctly run the CGI script.

- Insert the 9th parameter (Auth)

```
<td>Choose Auth:</td>

<td> <select name="Auth" style="margin-left: 10px; color: #000000; font-family: 
Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;">

  <option value="0">Open</option>

  <option value="1">Wep</option>

  <option value="2">Wpa & Wpa2 Personal</option>
  </select></td>
```

The “**name**” field (**Auth**) cannot be modified to correctly run the CGI script.

- Insert the 10th parameter (Mode)

```
<td>Choose Mode:</td>

<td> <select name="Mode" style="margin-left: 10px; color: #000000; font-family: Verdana; font-weight: bold; font-size: 12px; background-color: #cccccc;">

  <option value="0">Idle</option>

  <option value="1">Station</option>

  <option value="3">Mini AP</option>

  </select></td>
```

The “**name**” field (**Mode**) cannot be modified to correctly run the CGI script.

- Insert the last parameter (confirm)

```
<input type="submit" name="confirm" value="GO!" onclick="alert('Press OK to send, then wait a few seconds.')"
```

The “**name**” field (**confirm**) cannot be modified to correctly run the CGI script.

FIRST SET in-a-click

The first configuration of the module can be also performed by a single HTTP GET from the HOST.

FIRST SET

[module IP address]/firstset.cgi?KEY=anonymous&SSID=[insert the network SSID]&PWD=[insert the password]&IP=[only if DHCP is OFF]&MASK=[only if DHCP is OFF]&GW=[only if DHCP is OFF]&DNS=[only if DHCP is OFF]&Dhcp=[insert 1 for DHCP on, 0 for DHCP off]&IbssAuth=[only for WEP, 0 for OPEN System/1 for Shared Key]&Auth=[0 for open, 1 for WEP, 2 for WPA/WPA2]&Mode=[0 for idle, 1 for station, 2 for IBSS, 3 for Mini AP]&confirm=GO

i.e. how to configure a WPA/WPA2 network:

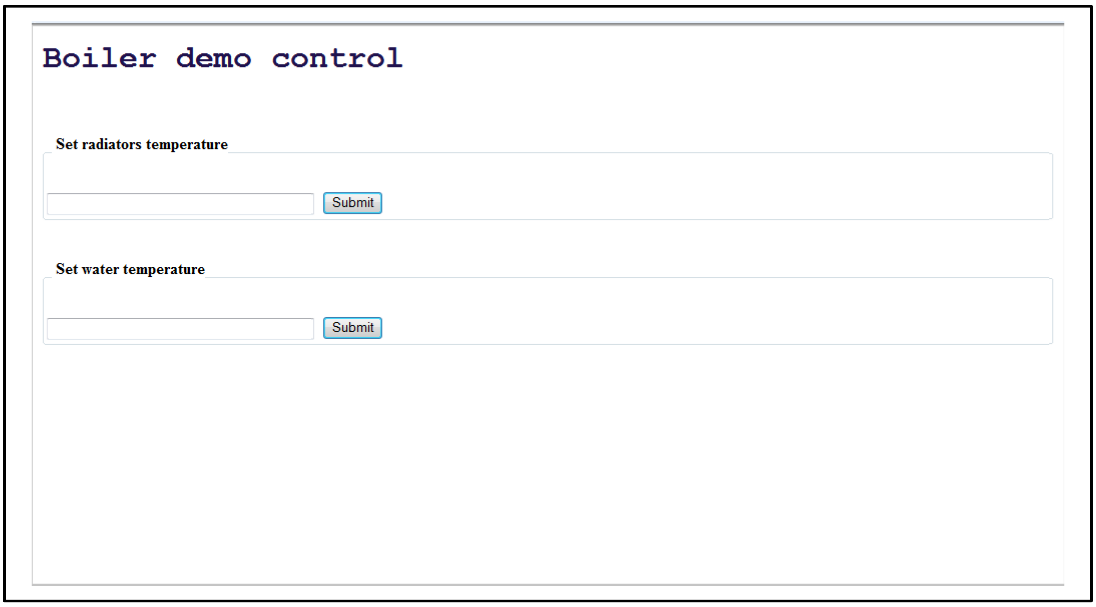
http://192.168.1.100/firstset.cgi?KEY=anonymous&SSID=AndroidAP&PWD=mypassword&Auth=2&Mode=1&confirm=GO

Appendix A

Output demo – multiple text boxes

The following example illustrates how to create a custom web page (i.e. output_boxes.html) containing multiple text boxes.

Figure 10: Boiler demo control page



Boiler demo control

Set radiators temperature

Set water temperature

HTML code

```
<html>
<head><title>Boiler demo control</title></head>
<body bgcolor="white" text="black">

<script>
   var allowSubmit = true;

   function testSubmit() {
      if (!allowSubmit)
         return false;

      allowSubmit = false;
      document.getElementById("submitButton").disabled = true;

      setTimeout(function() {
                    allowSubmit = true;
                    document.getElementById("submitButton").disabled = false;
                 }, 1000);
      return true;


   }
</script>
<span style="background-color: ;"><span style="color: #20124d;"><b><span
style="font-size: x-large;"><span style="font-family: &quot;Courier
New&quot;,Courier,monospace;">Boiler demo
control</span></span></b></i></span></span>
<br><br><br><br>
<fieldset>
<legend><h4>Set radiators temperature</h4></legend>
          <p><form name="CGI Example" method="GET" action="output.cgi"
onSubmit="return testSubmit();">
          <input type="text" name="text" size="40" maxlength="40">&nbsp;
                 <input id="submitButton" type="submit" name="submit" value="Submit"
>
              </form>
          </p>
</fieldset>
<br><br>
<fieldset>
<legend><h4>Set water temperature</h4></legend>
          <p><form name="CGI Example" method="GET" action=" output.cgi"
OnSubmit="setTimeout('redirect()', 10)">
          <input type="text" name="text" size="40" maxlength="40">&nbsp;
                 <input id="submitButton" type="submit" name="submit" value="Submit"
>
              </form>
          </p>
</fieldset>
</body>
</html>
```

HTML code description

Because the output.cgi automatically redirects to the output_demo.html, this HTML code allows the user to skip the automatic redirect, if it is not required.

In detail:

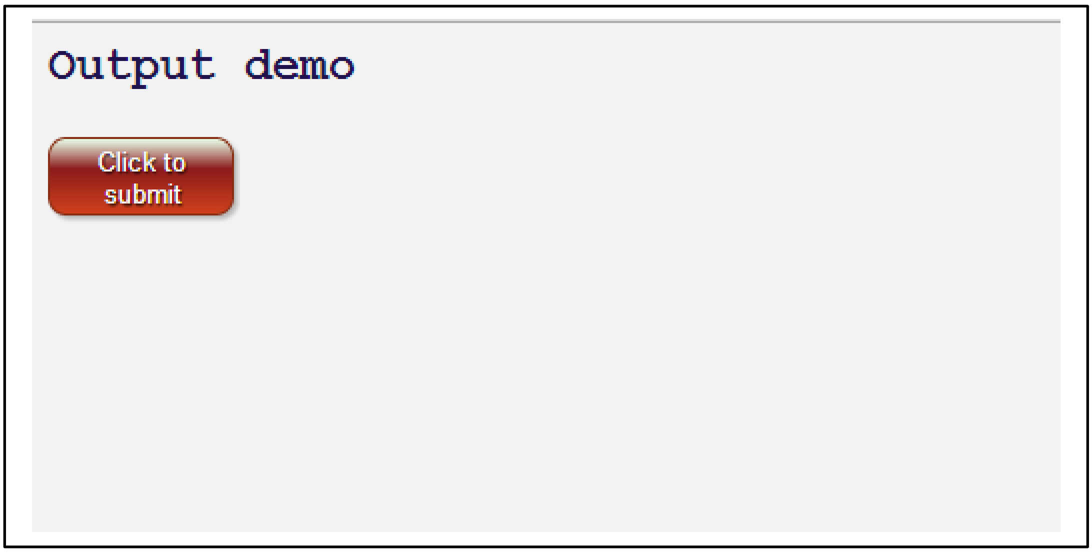
- The script section of the code enables skipping of the automatic redirect
- Insert the “form” including the “Onsubmit” tag:

```
<p><form name="CGI Example" method="GET" action=" output.cgi"
OnSubmit="setTimeout('redirect()',
```

Output demo using jQuery library – button

The following example shows how to create a web page (output_button.html) containing a button to send data (encapsulated in the button) from the HOST to the module’s UART. The jQuery library must be linked in the HTML code or added in the module’s web server.

Figure 11: Output demo page with submit button



Output demo

HTML code

```
<html>
<head>
</head>
<body bgcolor="#F2F2F2">
<script src="/jQuery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
               $.get("output.cgi?text=Data&submit=Submit");
  });
});
</script>
<span style="background-color: ;"><span style="color: #20124d;"><b><span
style="font-size: x-large;"><span style="font-family: &quot;Courier
New&quot;,Courier,monospace;">Output demo</span></span></b></i></span></span>
<br><br>
<DIV align="left">
<button style="
width: 7em;
padding: .2em;
color: #ffffff;
text-shadow: 1px 1px 1px #000000;
border: solid thin #882c13;
-webkit-border-radius: .7em;
-moz-border-radius: .7em;
border-radius: .7em;
-webkit-box-shadow: 2px 2px 3px #999999;
```

```
box-shadow: 2px 2px 2px #bbbbbb;

background-color: #ce411c;

background-image: -webkit-gradient(linear, left top, left bottom,

from(#e9fde8), to(#ce411c),color-stop(0.4, #8c1b1b));">Click to
submit</button></a></DIV>

</body>

</html>
```

HTML code description

The jQuery library (jQuery.min.js, ~70 KBytes) is included in the package and is already included in the outfile.img.

About the code:

- The first script allows data to be customized for sending to the module’s UART

```
$.get("output.cgi?text=Data&submit=Submit");
```

- The second part of the code allows the building of the HTML5 button.

```
<button style="

width: 7em;

padding: .2em;

color: #ffffff;

text-shadow: 1px 1px 1px #000000;

border: solid thin #882c13;

-webkit-border-radius: .7em;

-moz-border-radius: .7em;

border-radius: .7em;

-webkit-box-shadow: 2px 2px 3px #999999;

box-shadow: 2px 2px 2px #bbbbbb;

background-color: #ce411c;

background-image: -webkit-gradient(linear, left top, left bottom,

from(#e9fde8), to(#ce411c),color-stop(0.4, #8c1b1b));">Click to
submit</button></a></DIV>
```

3 Revision history

Table 1: Document revision history

Date	Version	Changes
14-Oct-2016	1	Initial release.

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.