Serial-to-WiFi GSLink Examples

Application Note AN022

Supports modules
GS1011M, GS1500M, GS2011M, and GS2100M

Releases
2.4.x, 2.5.x, 3.4.x, 3.5.x, and 5.1.1 or Later

Publication No. GS-AN022
Version: 1.1
Date: September 22, 2014

GainSpan Corporate Headquarters
3590 N. First Street
Suite 300
San Jose, CA 95134
Tel +1 (408) 627-6500
www.GainSpan.com

Release History

Version	Date	Remarks
1.0	13-Dec-11	Initial Release
1.1	14-Sept-22	Added AT command that supports setting up the GainSpan Node Username and Password specific for 5.1.0 firmware. See page 10.

Table of Contents

Figures

Chapter 1 System Overview

Purpose

This document provides examples for using GainSpan's Serial-to-WiFi GSLink. The examples provided in this document will allow you to perform restful HTTP and XML exchanges between either PCs or mobile devices and GainSpan's SoCs.

GSLink

The GSLink capability included within the GainSpan Serial-to-WiFi firmware (2.4.x and later for GS1011M and 3.4.x and later for GS1500M modules and 5.1.x and later for GS2000 modules) provides a mechanism to send and receive raw HTTP data as well as data in XML format.

Data can be sent and received either as a complete data as part of an HTTP message (raw HTTP method) or it can be sent and received as XML data, with each element being sent and received individually.

This is the case when the GainSpan node is acting as an HTTP Server and is sending or receiving data. In case of the GainSpan node being an HTTP Client it would know the type of communication it is doing with the server and can choose the raw HTTP or XML format of communication accordingly, because the communication is initiated by the GainSpan node.

When in raw HTTP communication mode, the complete XML data is sent or received by the Host as one data unit. In the case of the XML communication format, each element of the XML thread can be written individually and could be received individually helping the host parse and process each element easily.

Chapter 2 Sending XML Data to an HTTP Client

In the following example, the GainSpan node's webserver will accept an HTTP Get from an HTTP client and will indicate the host microcontroller connected to it through either UART or SPI that such an HTTP Get request was received from an HTTP client. The microcontroller will then provide data to the GainSpan node which it wishes to send back to the HTTP client as a response to the HTTP Get sent by it through GSLink. GSLink will then format and frame the microcontroller's data with the proper XML and HTTP encapsulation and the response will be sent by the GainSpan node as Wi-Fi payload with the proper TCP/IP and 802.11 framing so that the HTTP client at the other end will receive the requested XML data.

Software and Hardware Setup

To perform sending XML data to an HTTP client, you will need to first setup the following:

1. A GainSpan evaluation board.
2. A PC that complies with the following requirements:
    - Terminal program (Tera Term or equivalent)
    - WiFi network adapter
3. A serial or USB cable to connect to the GainSpan evaluation board.

Screen shots throughout this manual were taken using the GS1011M. However, they also apply to the GS2011M and GS2100M as well

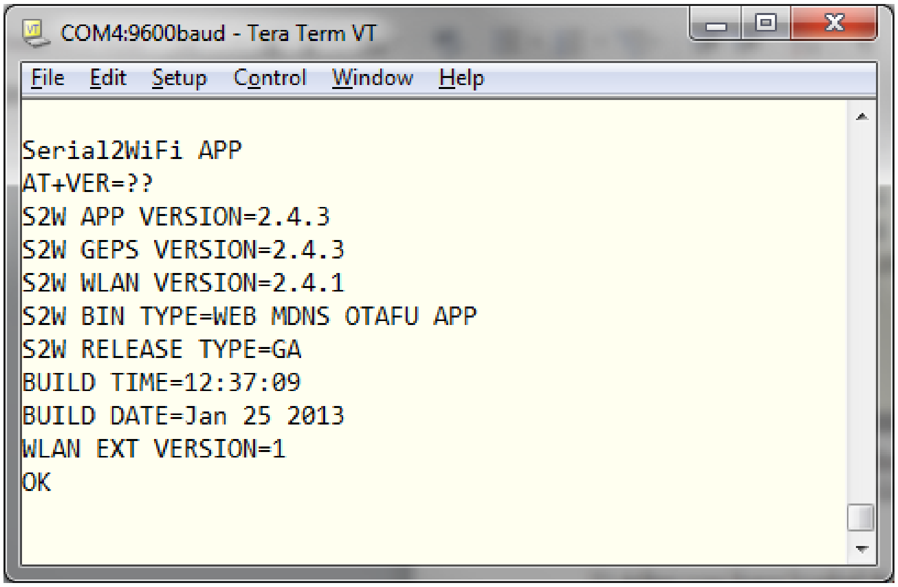
Load the Following Software

1. Load the module on your evaluation board with the following firmware:

   - For GS1011M:
     - 2.4.3 Wireless LAN firmware (WFW-REL-2_4_1.bin)
     - 2.4.3 Serial to Wi-Fi MDNS + OTA FWU App firmware
   - For GS1500M:
     - Use 3.4.x and 3.5.x binaries and firmware
   - For GS2K Module:
     - Use 5.1.x or later binaries and firmware

   The module will display the following firmware within the Tera Term VT window after booting (see Figure 1).

Figure 1: Serial-to-WiFi Firmware Loaded



```
Serial2WiFi APP
AT+VER=??
S2W APP VERSION=2.4.3
S2W GEPS VERSION=2.4.3
S2W WLAN VERSION=2.4.1
S2W BIN TYPE=WEB MDNS OTAFU APP
S2W RELEASE TYPE=GA
BUILD TIME=12:37:09
BUILD DATE=Jan 25 2013
WLAN EXT VERSION=1
OK
```

Make sure that you load the correct GainSpan firmware that has webserver, GSLink, and MDNS capabilities built into it.

2. After you have loaded the proper firmware into your GainSpan module, issue the following commands:

   ```
   at+ndhcp=0
   at+nset=192.168.8.1,255.255.255.0,192.168.8.1
   at+wm=2
   at+wa=example_gslink,,11
   at+dhcpsrvr=1
   at+webserver=1,admin,admin
   at+urirecv=/example
   at+xmlparse=1
   at+mdsstart
   at+mdnshreg=example_gslink,local
   at+mdnssrvreg=example,,_http,_tcp,local,80,0,api=gs_profile_example:
   0.7.0:/example,path=/example
   at+mdnsannounce
   ```

   The above sequence commands will perform the following:

   - Set the GainSpan node to use a fixed IP address (in this example, 192.168.8.1).

     ```
     at+ndhcp=0
     at+nset=192.168.8.1,255.255.255.0,192.168.8.1
     ```

   - Configure the GainSpan node as a limited access point (in this example, setting it up with the SSID=example_gslink and operating in WiFi channel 11).

     ```
     at+wm=2
     at+wa=example_gslink,,11
     ```

   - Enable the GainSpan node's DHCP server so that clients that associate to it can obtain an IP address automatically from it.

     ```
     at+dhcpsrvr=1
     ```

- Enable the GainSpan node's webserver, setting up a user name and password for the clients that want to connect to it.

  User Name: admin
  Password: admin

```
at+webserver=1,admin,admin
```

For the GS2000 firmware release 5.1.0, only change the above command to:

```
at+webserver=1,admin,admin,0
```

If this command isn't issued, an error is returned and the procedure will not work with this version of firmware.

- Configure the GainSpan node so that any request that arrives at its webserver for the URL/example.html will be forwarded to the microcontroller connected to the GainSpan node via its serial interface instead of being handled directly by the GainSpan node's webserver.

```
at+urirecv=/example
```

- Enable XML parsing for data sent and received through the webserver by the GainSpan node.

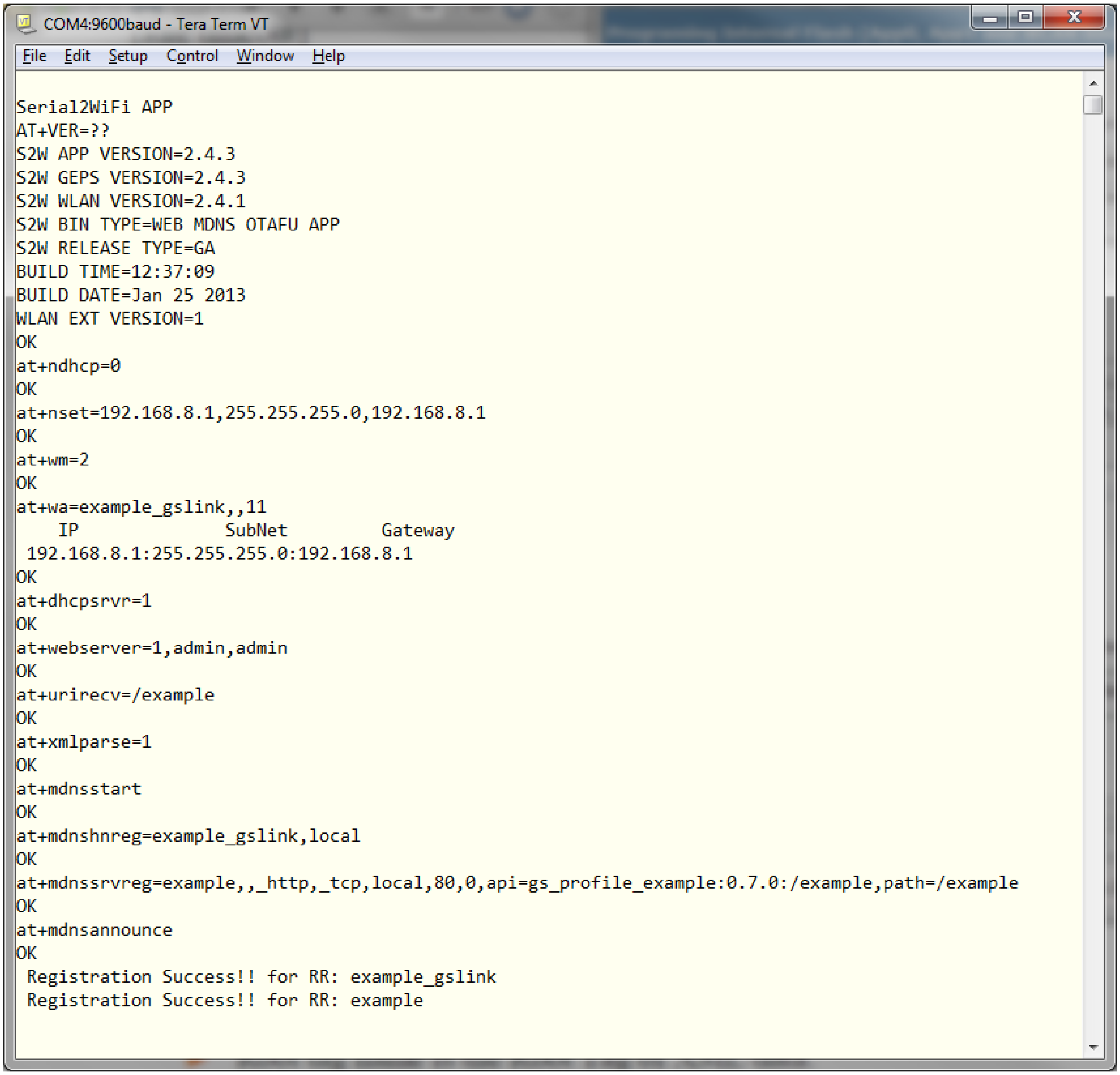
```
at+xmlparse=1
```

- Enable MDNS so that the GainSpan node will provide MDNS advertisement of the service that provides the XML entities that the microcontroller will server through the \example.html URL. This can be used so that HTTP clients do not need to know the exact URL (\example.html) in this case, and get to know the URl used to transmit the microcontroller's data through those MDNS advertisements.

```
at+mdnsstart'
at+mdnshnreg=example_gslink.local
at+mdnssrvreg=example,,_http,_tcp,local,80,0,api=gs_profile_ex
ample:0.7.0:/example,path=/example
at+mdsannouce
```

Figure 2 shows the above command sequence being issued to a GainSpan node.

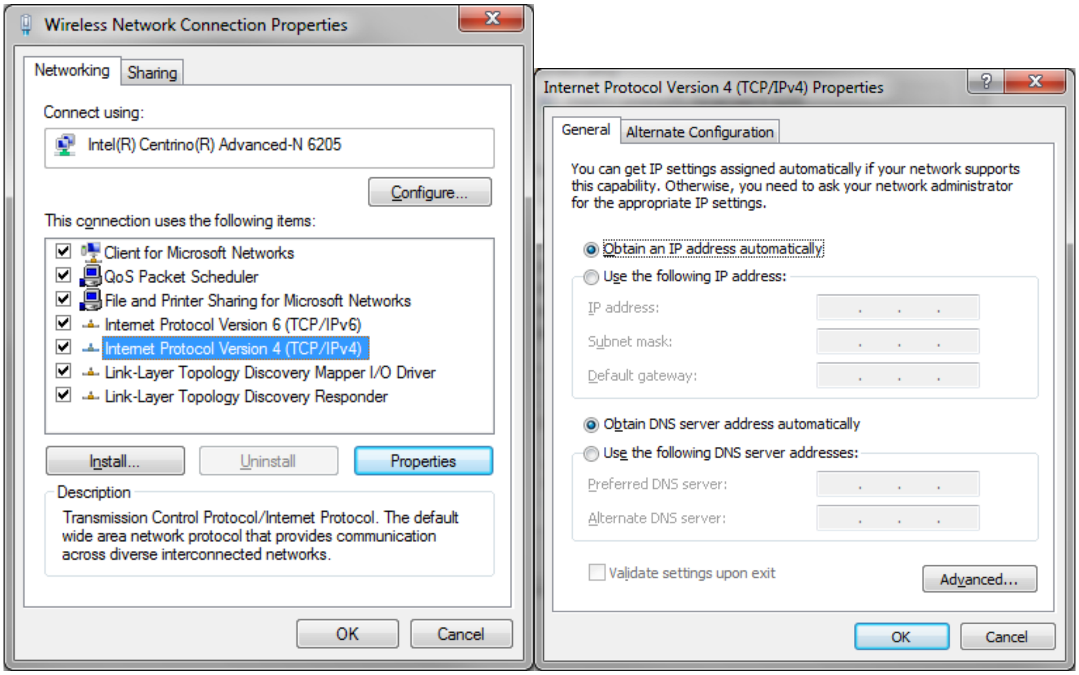
Figure 2: Output of Loading Firmware


COM4:9600baud - Tera Term VT
File Edit Setup Control Window Help

```
Serial2WiFi APP
AT+VER=??
S2W APP VERSION=2.4.3
S2W GEPS VERSION=2.4.3
S2W WLAN VERSION=2.4.1
S2W BIN TYPE=WEB MDNS OTAFU APP
S2W RELEASE TYPE=GA
BUILD TIME=12:37:09
BUILD DATE=Jan 25 2013
WLAN EXT VERSION=1
OK
at+ndhcp=0
OK
at+nset=192.168.8.1,255.255.255.0,192.168.8.1
OK
at+wm=2
OK
at+wa=example_gslink,,11
    IP              SubNet          Gateway
 192.168.8.1:255.255.255.0:192.168.8.1
OK
at+dhcpsrvr=1
OK
at+webserver=1,admin,admin
OK
at+urirecv=/example
OK
at+xmlparse=1
OK
at+mdnsstart
OK
at+mdnshnreg=example_gslink,local
OK
at+mdnssrvreg=example,,_http,_tcp,local,80,0,api=gs_profile_example:0.7.0:/example,path=/example
OK
at+mdnsannounce
OK
 Registration Success!! for RR: example_gslink
 Registration Success!! for RR: example
```

3. Make sure that your PCs wireless adapter is configured to obtain an IP address automatically (see Figure 3).

Figure 3: Obtaining IP Address Automatically



Wireless Network Connection Properties

Networking | Sharing

Connect using:

Intel(R) Centrino(R) Advanced-N 6205

Configure...

This connection uses the following items:

☑ Client for Microsoft Networks
☑ QoS Packet Scheduler
☑ File and Printer Sharing for Microsoft Networks
☑ Internet Protocol Version 6 (TCP/IPv6)
☑ Internet Protocol Version 4 (TCP/IPv4)
☑ Link-Layer Topology Discovery Mapper I/O Driver
☑ Link-Layer Topology Discovery Responder

Install... | Uninstall | Properties

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

OK | Cancel

Internet Protocol Version 4 (TCP/IPv4) Properties

General | Alternate Configuration

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically
○ Use the following IP address:

IP address: . . .
Subnet mask: . . .
Default gateway: . . .

● Obtain DNS server address automatically
○ Use the following DNS server addresses:

Preferred DNS server: . . .
Alternate DNS server: . . .

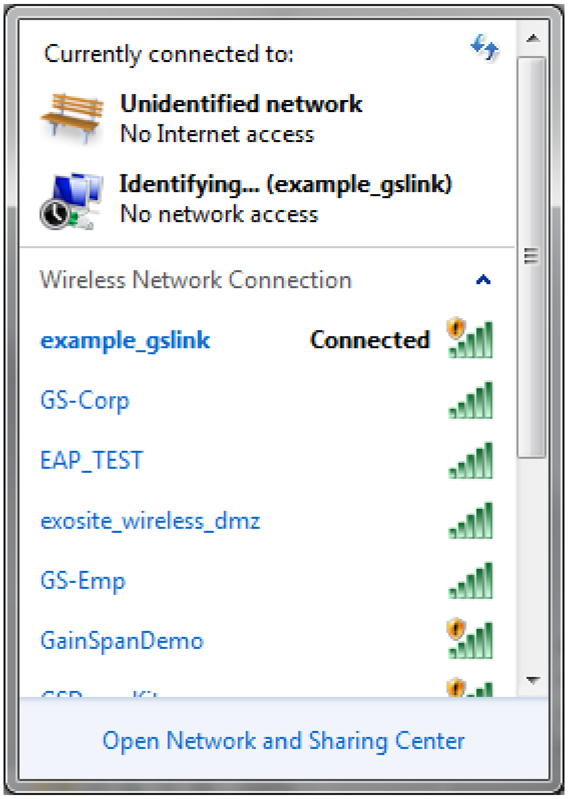
□ Validate settings upon exit

Advanced...

OK | Cancel

4. On your PC, have your PC's wireless adapter associate to the created "example_gslink" network (see Figure 4).

Figure 4: Associating PC with Wireless Adapter



Currently connected to:

Unidentified network
No Internet access

Identifying... (example_gslink)
No network access

Wireless Network Connection

example_gslink **Connected**

GS-Corp

EAP_TEST

exosite_wireless_dmz

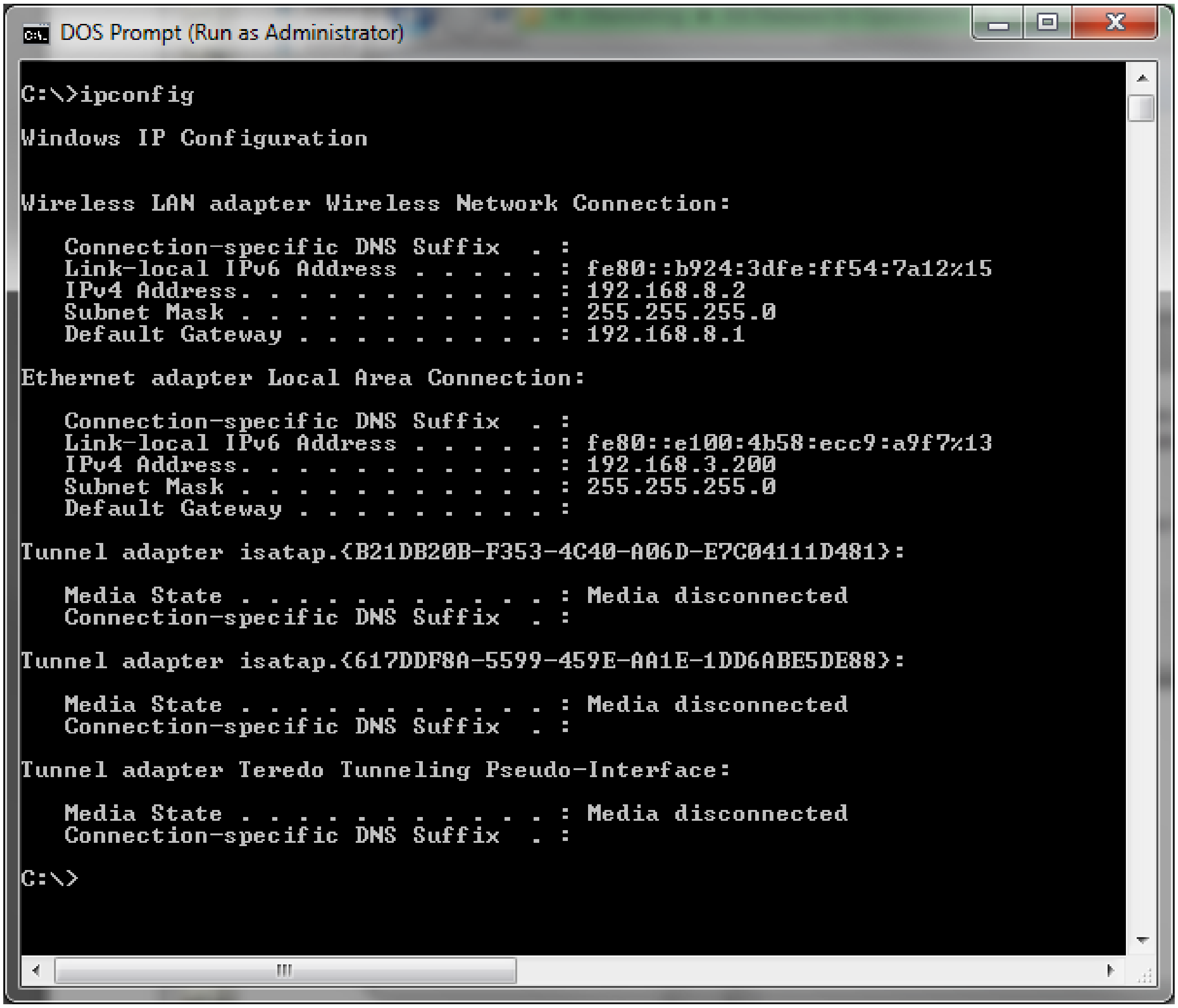
GS-Emp

GainSpanDemo

Open Network and Sharing Center

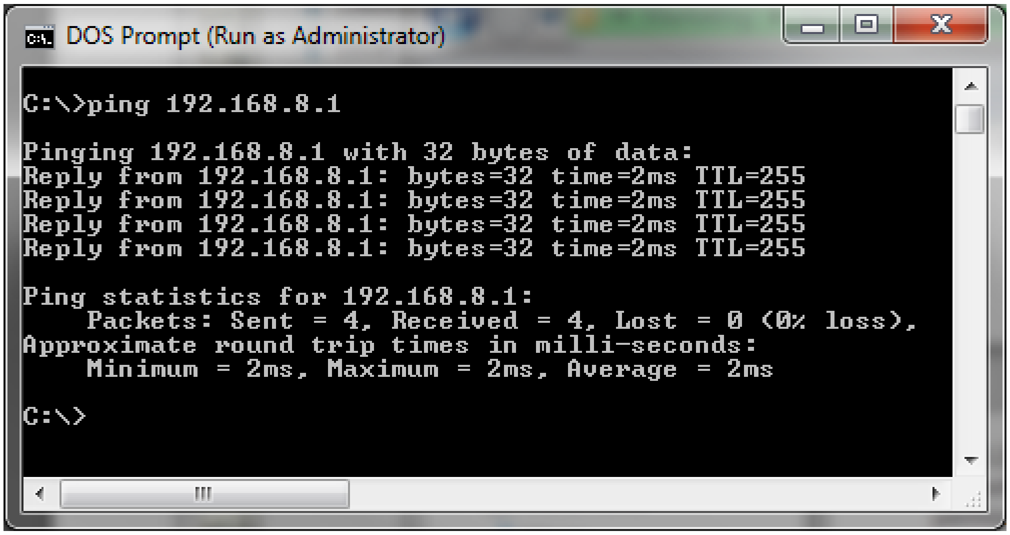
5. Once your PCs wireless adapter has associated to the “example_gslink” network, you will see that it has obtained an IP address in the 192.168.8.x subnet and that it can ping the GainSpan node at 192.168.8.1 as shown in Figure 5 and Figure 6.

Figure 5: Obtaining an IP Address



```
C:\>ipconfig

Windows IP Configuration


Wireless LAN adapter Wireless Network Connection:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::b924:3dfe:ff54:7a12%15
   IPv4 Address. . . . . . . . . . . : 192.168.8.2
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.8.1

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::e100:4b58:ecc9:a9f7%13
   IPv4 Address. . . . . . . . . . . : 192.168.3.200
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Tunnel adapter isatap.{B21DB20B-F353-4C40-A06D-E7C04111D481}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter isatap.{617DDF8A-5599-459E-AA1E-1DD6ABE5DE88}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Teredo Tunneling Pseudo-Interface:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\>
```

Figure 6: Pinging the GainSpan Node After Obtaining an IP Address from It



```
C:\>ping 192.168.8.1

Pinging 192.168.8.1 with 32 bytes of data:
Reply from 192.168.8.1: bytes=32 time=2ms TTL=255
Reply from 192.168.8.1: bytes=32 time=2ms TTL=255
Reply from 192.168.8.1: bytes=32 time=2ms TTL=255
Reply from 192.168.8.1: bytes=32 time=2ms TTL=255

Ping statistics for 192.168.8.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 2ms, Average = 2ms

C:\>
```

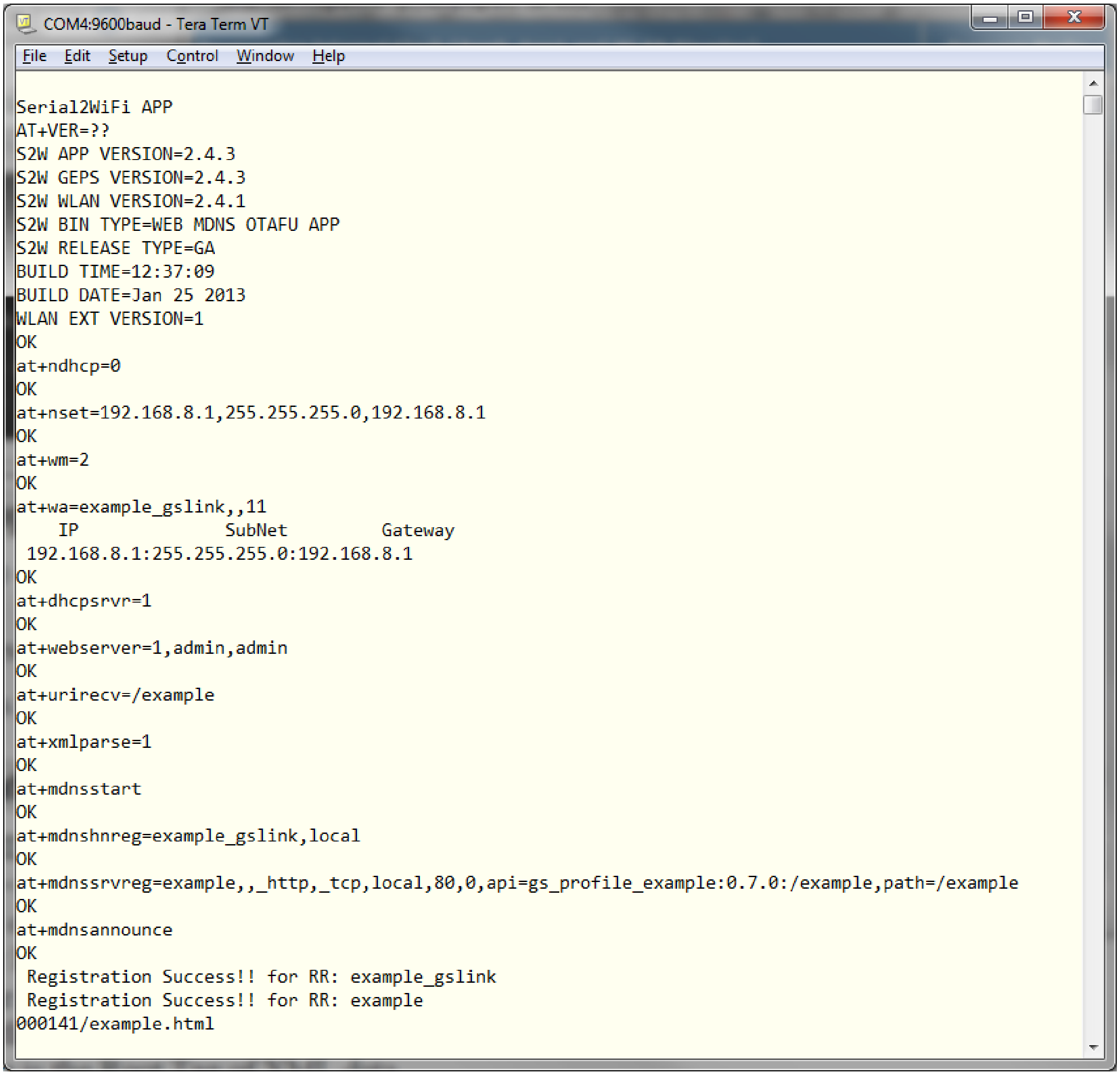
6. Open up the 192.168.1/example.html URL in a browser on your PC. The webserver running in the GainSpan module will request a User Name and Password. This was configured previously in the command sequence issued to the module.

7. Enter as follows:

   User Name: admin
   Password: admin

8. The GainSpan node will notify the microcontroller that this URL has been requested with the following - 000141/example.html (see Figure 7).

Figure 7: URL Requested Example HTML



COM4:9600baud - Tera Term VT
File Edit Setup Control Window Help

```
Serial2WiFi APP
AT+VER=??
S2W APP VERSION=2.4.3
S2W GEPS VERSION=2.4.3
S2W WLAN VERSION=2.4.1
S2W BIN TYPE=WEB MDNS OTAFU APP
S2W RELEASE TYPE=GA
BUILD TIME=12:37:09
BUILD DATE=Jan 25 2013
WLAN EXT VERSION=1
OK
at+ndhcp=0
OK
at+nset=192.168.8.1,255.255.255.0,192.168.8.1
OK
at+wm=2
OK
at+wa=example_gslink,,11
    IP              SubNet         Gateway
 192.168.8.1:255.255.255.0:192.168.8.1
OK
at+dhcpsrvr=1
OK
at+webserver=1,admin,admin
OK
at+urirecv=/example
OK
at+xmlparse=1
OK
at+mdnsstart
OK
at+mdnshnreg=example_gslink,local
OK
at+mdnssrvreg=example,,_http,_tcp,local,80,0,api=gs_profile_example:0.7.0:/example,path=/example
OK
at+mdnsannounce
OK
 Registration Success!! for RR: example_gslink
 Registration Success!! for RR: example
000141/example.html
```

9. The microcontroller can then reply to this request by issuing the following command.

```
at+xmlsend=<CID>,<Type>,<Timeout>,<Page URI>,<Root tag name>[,<N>]
```

Followed by [N] sets of data to be sent via XML formatted by the microcontroller as follows:

```
<ESC>G<CID><len><tagname>:<value>
```

For this example, the microcontroller will send the following XML data:

```
Temp=22(Temperature = 22 degrees)
Light=313 (Light = 313 lumens)
Acc=924,803,228 (Accelerometer coordinates = 924,803,228)
Leds=1 (LEDs are turned on)
```

Formatted as XML data in the following way:

```
<example>
      <temp>22</temp>
      <light>313</light>
      <acc>924,803,228</acc>
      <leds>1</leds>
<example>
```

On the /example.html URL.

For this purpose, the microcontroller will send the following to the GainSpan node:

<CID> = The CID that was given to the connection from the HTTP client (in this case 0)

<Type>= The type of response that the microcontroller wants to send to the HTTP client. In this example, the microcontroller will be replying to an HTTP Get sent by the client, thus will use a GETRESP=6

<Timeout>= The HTTP timeout that should be used for the transmission (in this example, it is set to 100).

<Page URL> = The URL on which the HTTP data will be transmitted. In this case /example.html.

<Root Tag name>= What will be the root tag name used for the XML data. In this example “example”.

<N>= number of XML elements that will be sent out. In this example – 4.

Followed by 4 sets of data formatted in the following way:

```
<ESC>G
00007temp:22
<ESC>G
00009light:31’
<ESC>G
00015acc:924,803,228
<ESC>G
00006leds:1
```

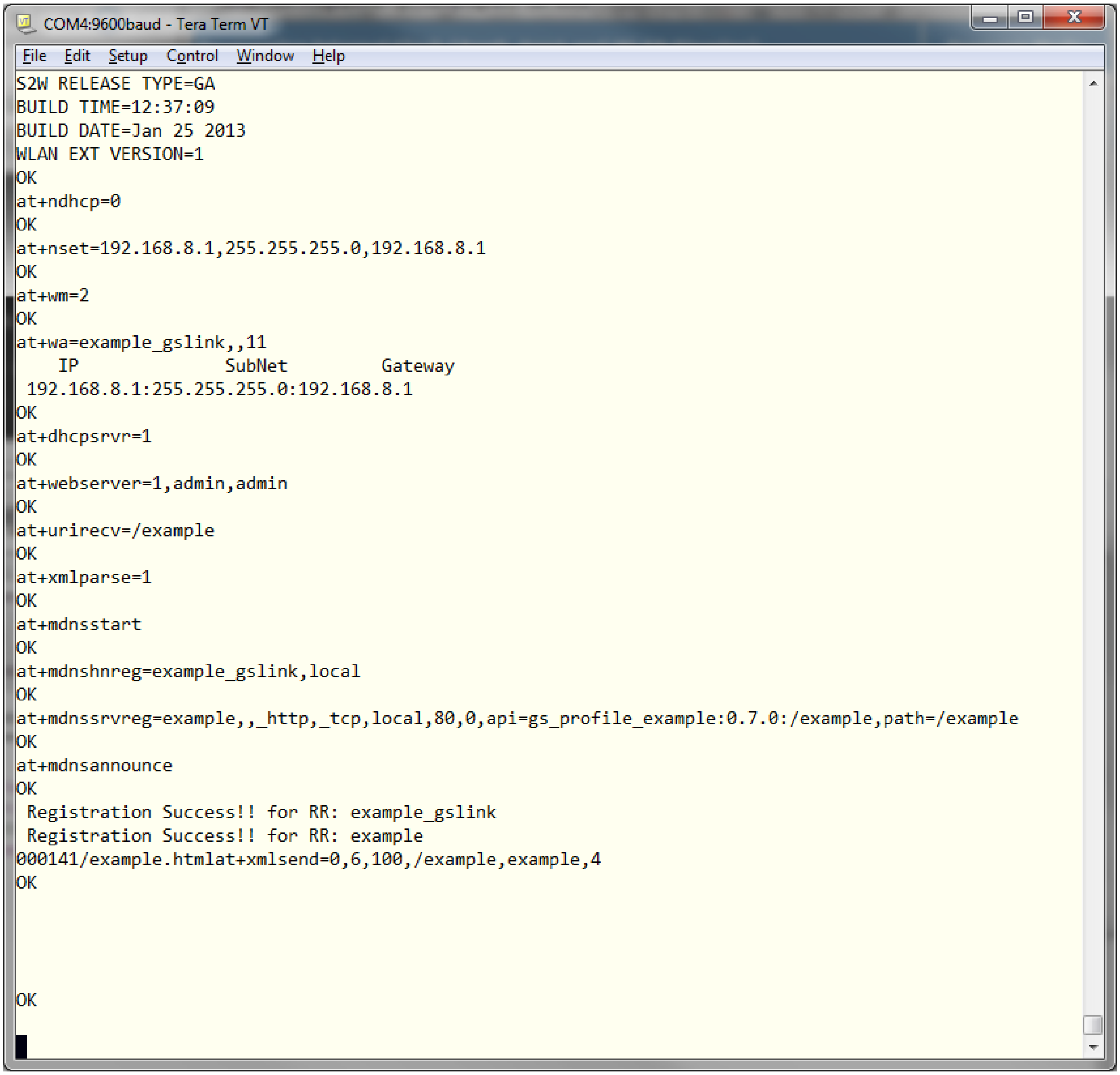
Where <ESC> is the ESCAPE character (ASCII 27dec)

Or, as follows:

```
sendln'at+xmlsend=0,6,100,/example,example,4'
<ESC>G
00007temp:22
<ESC>G
00009light:31'
<ESC>G
00015acc:924,803,228
<ESC>G
sendln '00006leds:1'
```

This is displayed in Figure 8 below.

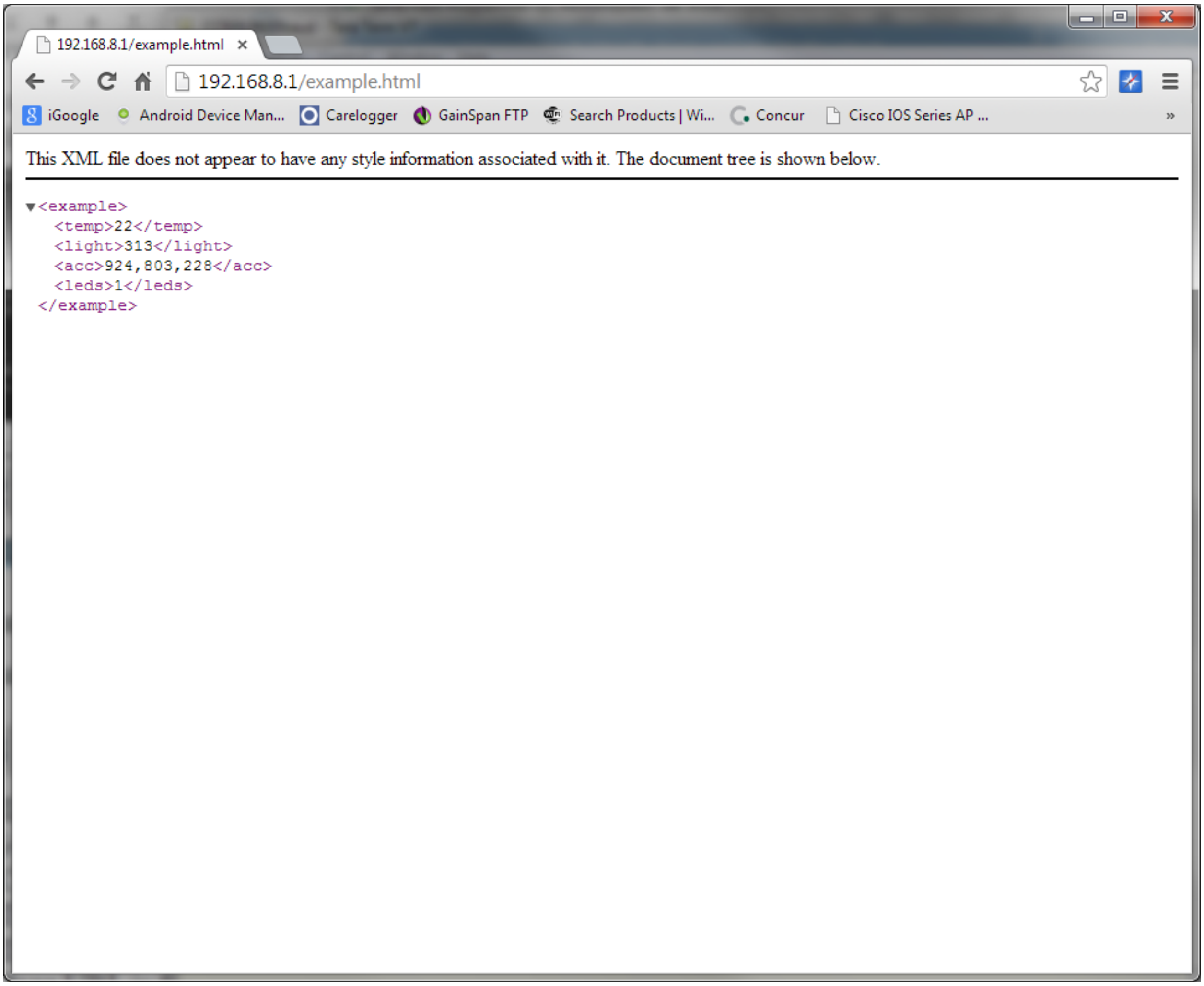
Figure 8: Example Registered Successfully


COM4:9600baud - Tera Term VT
File Edit Setup Control Window Help

```
S2W RELEASE TYPE=GA
BUILD TIME=12:37:09
BUILD DATE=Jan 25 2013
WLAN EXT VERSION=1
OK
at+ndhcp=0
OK
at+nset=192.168.8.1,255.255.255.0,192.168.8.1
OK
at+wm=2
OK
at+wa=example_gslink,,11
    IP              SubNet          Gateway
 192.168.8.1:255.255.255.0:192.168.8.1
OK
at+dhcpsrvr=1
OK
at+webserver=1,admin,admin
OK
at+urirecv=/example
OK
at+xmlparse=1
OK
at+mdnsstart
OK
at+mdnshnreg=example_gslink,local
OK
at+mdnssrvreg=example,,_http,_tcp,local,80,0,api=gs_profile_example:0.7.0:/example,path=/example
OK
at+mdnsannounce
OK
 Registration Success!! for RR: example_gslink
 Registration Success!! for RR: example
000141/example.htmlat+xmlsend=0,6,100,/example,example,4
OK




OK
```

The browser on the PC (The HTTP client in this example) will then display the transmitted XML entities (see Figure 9).

Figure 9: Transmitted XML Entities Displayed in Browser Window



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<example>
   <temp>22</temp>
   <light>313</light>
   <acc>924,803,228</acc>
   <leds>1</leds>
 </example>
```

Tera Term VT Macros

If Tera Term VT is going to be used as the terminal program to replicate the above example, the following Tera Term macros can be used for that purpose.

Setup GainSpan Node

To perform the setup of the GainSpan node.

```
sendln'at+ndhcp=0'
sendln'at+nset=192.168.8.1,255.255.255.0,192.168.8.1'
sendln'at+wm=2'
sendln'at+wa=example_gslink,,11'
sendln'at+dhcpsrvr=1'
sendln'at+webserver=1,admin,admin'
sendln'at+urirecv=/example'
sendln'at+xmlparse=1'
sendln'at+mdnsstart'
sendln'at+mdnshnreg=example_gslink,local'
sendln'at+mdnssrvreg=example,,_http,_tcp,local,80,0,api=gs_profile_example
:0.7.0:/example,path=/example'
sendln'at+mdnsannounce'
```

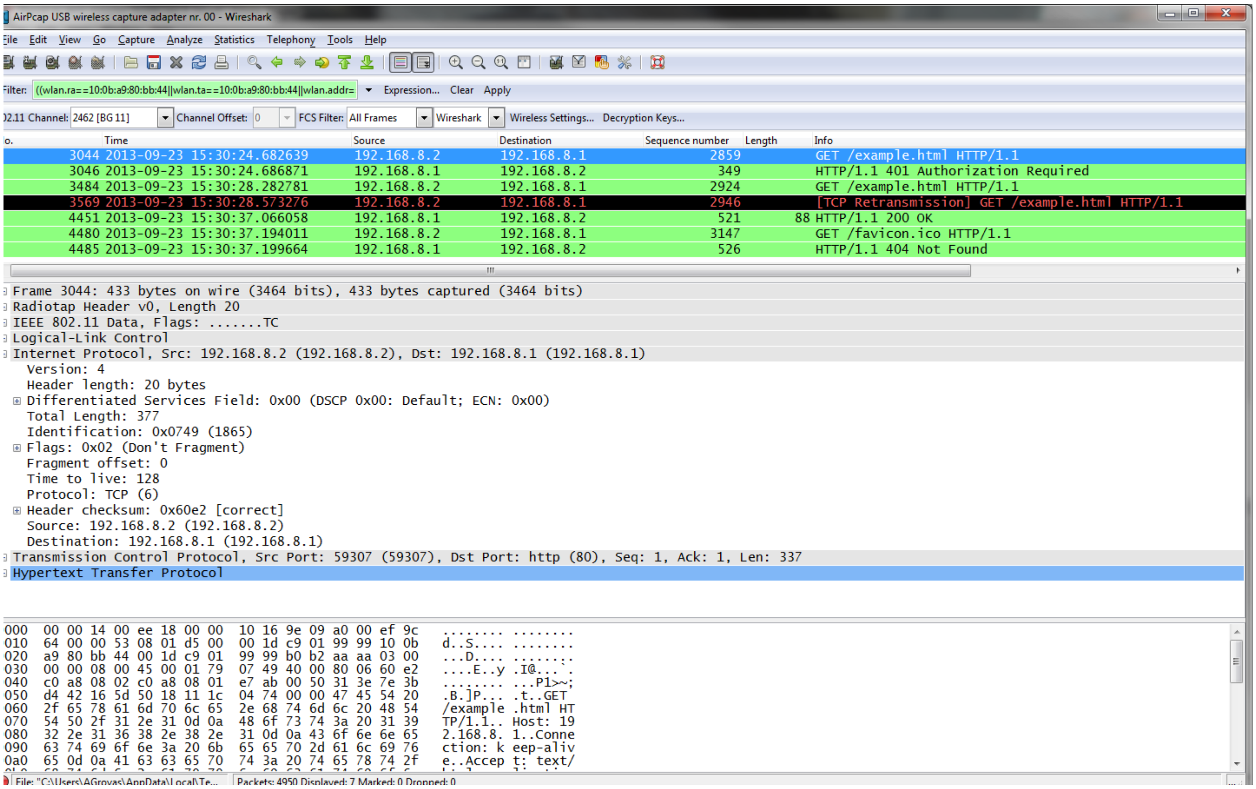
Simulate Microcontroller Response to the HTTP GET

To simulate the microcontroller response to the HTTP Get transmitting the above 4 XML entities

```
sendln'at+xmlsend=0,6,100,/example,example,4'
wait 'OK'

send 27 71 ;ESC + G
sendln '00007temp:22'

send 27 71 ; ESC + G
sendln '00009light:313'

send 27 71 ; ESC + G
sendln '00015acc:924,803,228'

send 27 71 ; ESC + G
sendln '00006leds:1'
```

Figure 10 shows the HTTP traffic exchanged between the GainSpan node's webserver and the PC's browser HTTP client.

Figure 10: HTTP Traffic Exchanged (Screen 1)


AirPcap USB wireless capture adapter nr. 00 - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Expression... Clear Apply
Wireless Settings... Decryption Keys...

Filter: ((wlan.ra==10:0b:a9:80:bb:44||wlan.ta==10:0b:a9:80:bb:44||wlan.addr=

02.11 Channel: 2462 [BG 11] Channel Offset: 0 FCS Filter: All Frames Wireshark

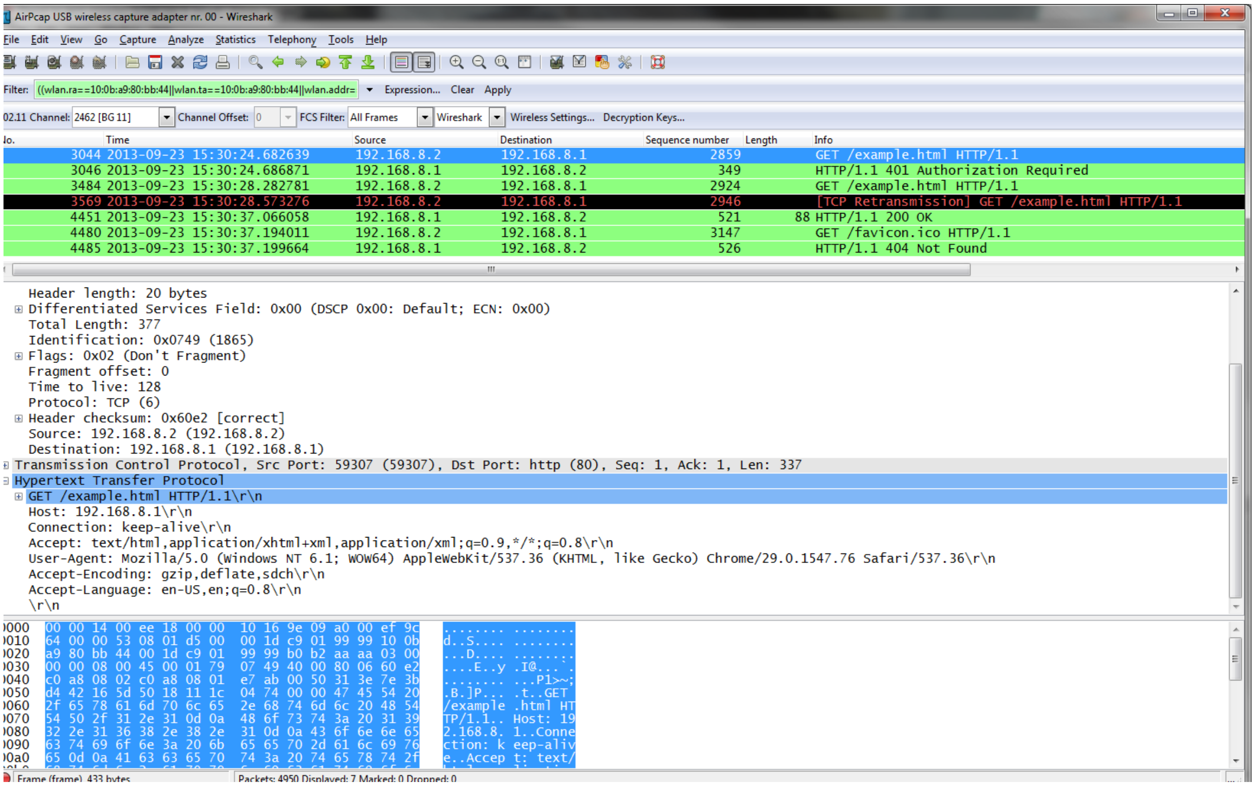
```
o.    Time                          Source          Destination     Sequence number  Length  Info
3044 2013-09-23 15:30:24.682639  192.168.8.2     192.168.8.1               2859            GET /example.html HTTP/1.1
3046 2013-09-23 15:30:24.686871  192.168.8.1     192.168.8.2                349            HTTP/1.1 401 Authorization Required
3484 2013-09-23 15:30:28.282781  192.168.8.2     192.168.8.1               2924            GET /example.html HTTP/1.1
3569 2013-09-23 15:30:28.573276  192.168.8.2     192.168.8.1               2946            [TCP Retransmission] GET /example.html HTTP/1.1
4451 2013-09-23 15:30:37.066058  192.168.8.1     192.168.8.2                521         88 HTTP/1.1 200 OK
4480 2013-09-23 15:30:37.194011  192.168.8.2     192.168.8.1               3147            GET /favicon.ico HTTP/1.1
4485 2013-09-23 15:30:37.199664  192.168.8.1     192.168.8.2                526            HTTP/1.1 404 Not Found
```

```
Frame 3044: 433 bytes on wire (3464 bits), 433 bytes captured (3464 bits)
Radiotap Header v0, Length 20
IEEE 802.11 Data, Flags: .......TC
Logical-Link Control
Internet Protocol, Src: 192.168.8.2 (192.168.8.2), Dst: 192.168.8.1 (192.168.8.1)
  Version: 4
  Header length: 20 bytes
⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
  Total Length: 377
  Identification: 0x0749 (1865)
⊞ Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 128
  Protocol: TCP (6)
⊞ Header checksum: 0x60e2 [correct]
  Source: 192.168.8.2 (192.168.8.2)
  Destination: 192.168.8.1 (192.168.8.1)
Transmission Control Protocol, Src Port: 59307 (59307), Dst Port: http (80), Seq: 1, Ack: 1, Len: 337
Hypertext Transfer Protocol
```

```
000  00 00 14 00 ee 18 00 00  10 16 9e 09 a0 00 ef 9c   ........ ........
010  64 00 00 53 08 01 d5 00  00 1d c9 01 99 99 10 0b   d..S.... ........
020  a9 80 bb 44 00 1d c9 01  99 99 b0 b2 aa aa 03 00   ...D.... ........
030  00 00 08 00 45 00 01 79  07 49 40 00 80 06 60 e2   ....E..y .I@...`.
040  c0 a8 08 02 c0 a8 08 01  e7 ab 00 50 31 3e 7e 3b   ........ ...P1>~;
050  d4 42 16 5d 50 18 11 1c  04 74 00 00 47 45 54 20   .B.]P... .t..GET 
060  2f 65 78 61 6d 70 6c 65  2e 68 74 6d 6c 20 48 54   /example .html HT
070  54 50 2f 31 2e 31 0d 0a  48 6f 73 74 3a 20 31 39   TP/1.1.. Host: 19
080  32 2e 31 36 38 2e 38 2e  31 0d 0a 43 6f 6e 6e 65   2.168.8. 1..Conne
090  63 74 69 6f 6e 3a 20 6b  65 65 70 2d 61 6c 69 76   ction: k eep-aliv
0a0  65 0d 0a 41 63 63 65 70  74 3a 20 74 65 78 74 2f   e..Accep t: text/
```

File: "C:\Users\AGrovas\AppData\Local\Te... Packets: 4950 Displayed: 7 Marked: 0 Dropped: 0

Figure 11: HTTP Traffic Exchanged (Screen 2)


AirPcap USB wireless capture adapter nr. 00 - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Expression... Clear Apply
Wireless Settings... Decryption Keys...

Filter: ((wlan.ra==10:0b:a9:80:bb:44||wlan.ta==10:0b:a9:80:bb:44||wlan.addr=

02.11 Channel: 2462 [BG 11] Channel Offset: 0 FCS Filter: All Frames Wireshark

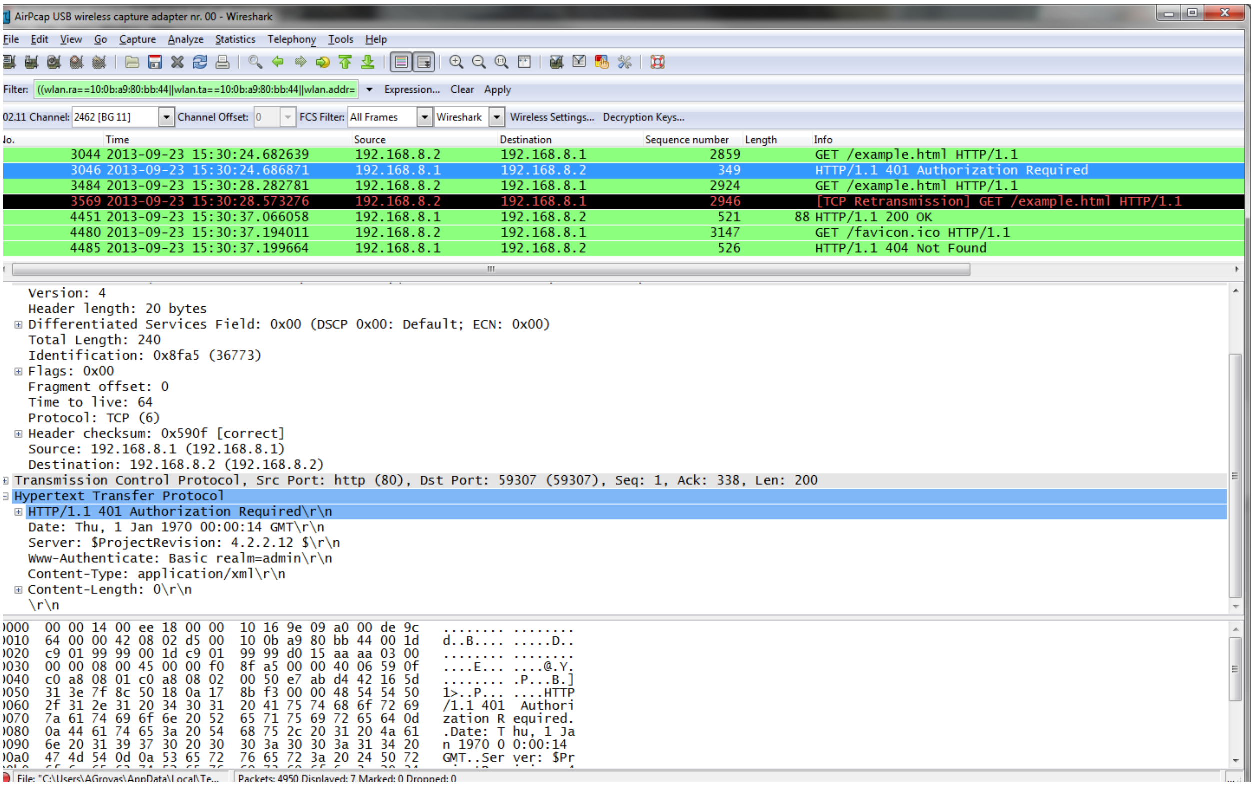
No.	Time	Source	Destination	Sequence number	Length	Info
3044	2013-09-23 15:30:24.682639	192.168.8.2	192.168.8.1	2859		GET /example.html HTTP/1.1
3046	2013-09-23 15:30:24.686871	192.168.8.1	192.168.8.2	349		HTTP/1.1 401 Authorization Required
3484	2013-09-23 15:30:28.282781	192.168.8.2	192.168.8.1	2924		GET /example.html HTTP/1.1
3569	2013-09-23 15:30:28.573276	192.168.8.2	192.168.8.1	2946		[TCP Retransmission] GET /example.html HTTP/1.1
4451	2013-09-23 15:30:37.066058	192.168.8.1	192.168.8.2	521	88	HTTP/1.1 200 OK
4480	2013-09-23 15:30:37.194011	192.168.8.2	192.168.8.1	3147		GET /favicon.ico HTTP/1.1
4485	2013-09-23 15:30:37.199664	192.168.8.1	192.168.8.2	526		HTTP/1.1 404 Not Found

```
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 377
    Identification: 0x0749 (1865)
  ⊞ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (6)
  ⊞ Header checksum: 0x60e2 [correct]
    Source: 192.168.8.2 (192.168.8.2)
    Destination: 192.168.8.1 (192.168.8.1)
⊞ Transmission Control Protocol, Src Port: 59307 (59307), Dst Port: http (80), Seq: 1, Ack: 1, Len: 337
⊟ Hypertext Transfer Protocol
  ⊞ GET /example.html HTTP/1.1\r\n
    Host: 192.168.8.1\r\n
    Connection: keep-alive\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/29.0.1547.76 Safari/537.36\r\n
    Accept-Encoding: gzip,deflate,sdch\r\n
    Accept-Language: en-US,en;q=0.8\r\n
    \r\n
```

```
0000  00 00 14 00 ee 18 00 00  10 16 9e 09 a0 00 ef 9c   ........ ........
0010  64 00 00 53 08 01 d5 00  00 1d c9 01 99 99 10 0b   d..S.... ........
0020  a9 80 bb 44 00 1d c9 01  99 99 b0 b2 aa aa 03 00   ...D.... ........
0030  00 00 08 00 45 00 01 79  07 49 40 00 80 06 60 e2   ....E..y .I@...`.
0040  c0 a8 08 02 c0 a8 08 01  e7 ab 00 50 31 3e 7e 3b   ........ ...P1>~;
0050  d4 42 16 5d 50 18 11 1c  04 74 00 00 47 45 54 20   .B.]P... .t..GET 
0060  2f 65 78 61 6d 70 6c 65  2e 68 74 6d 6c 20 48 54   /example .html HT
0070  54 50 2f 31 2e 31 0d 0a  48 6f 73 74 3a 20 31 39   TP/1.1.. Host: 19
0080  32 2e 31 36 38 2e 38 2e  31 0d 0a 43 6f 6e 6e 65   2.168.8. 1..Conne
0090  63 74 69 6f 6e 3a 20 6b  65 65 70 2d 61 6c 69 76   ction: k eep-aliv
00a0  65 0d 0a 41 63 63 65 70  74 3a 20 74 65 78 74 2f   e..Accep t: text/
```

Frame (frame) 433 bytes | Packets: 4950 Displayed: 7 Marked: 0 Dropped: 0

Figure 12: HTTP Traffic Exchanged (Screen 3)


AirPcap USB wireless capture adapter nr. 00 - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Expression... Clear Apply
Wireless Settings... Decryption Keys...

Filter: ((wlan.ra==10:0b:a9:80:bb:44||wlan.ta==10:0b:a9:80:bb:44||wlan.addr=

02.11 Channel: 2462 [BG 11] Channel Offset: 0 FCS Filter: All Frames Wireshark

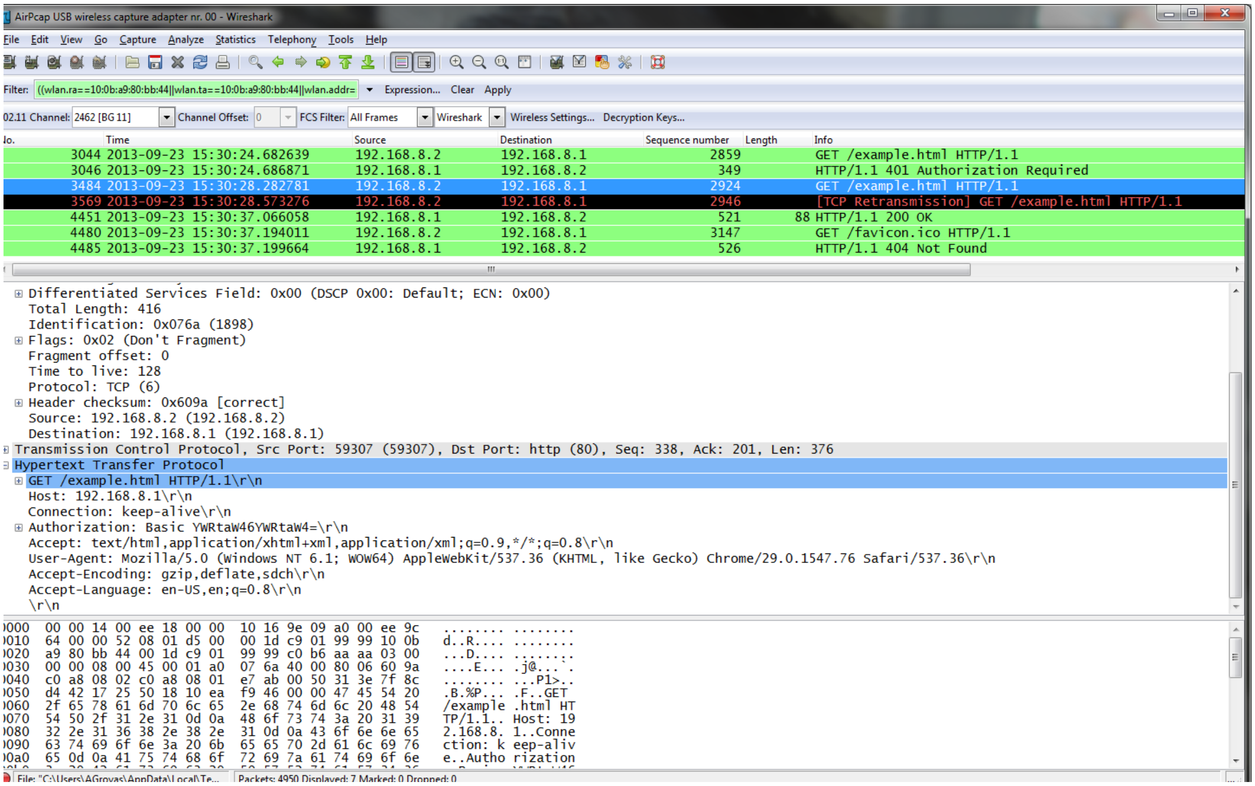
```
lo.      Time                        Source        Destination   Sequence number Length Info
    3044 2013-09-23 15:30:24.682639  192.168.8.2   192.168.8.1        2859              GET /example.html HTTP/1.1
    3046 2013-09-23 15:30:24.686871  192.168.8.1   192.168.8.2         349              HTTP/1.1 401 Authorization Required
    3484 2013-09-23 15:30:28.282781  192.168.8.2   192.168.8.1        2924              GET /example.html HTTP/1.1
    3569 2013-09-23 15:30:28.573276  192.168.8.2   192.168.8.1        2946              [TCP Retransmission] GET /example.html HTTP/1.1
    4451 2013-09-23 15:30:37.066058  192.168.8.1   192.168.8.2         521      88 HTTP/1.1 200 OK
    4480 2013-09-23 15:30:37.194011  192.168.8.2   192.168.8.1        3147              GET /favicon.ico HTTP/1.1
    4485 2013-09-23 15:30:37.199664  192.168.8.1   192.168.8.2         526              HTTP/1.1 404 Not Found
```

```
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 240
    Identification: 0x8fa5 (36773)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (6)
  ⊞ Header checksum: 0x590f [correct]
    Source: 192.168.8.1 (192.168.8.1)
    Destination: 192.168.8.2 (192.168.8.2)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 59307 (59307), Seq: 1, Ack: 338, Len: 200
⊟ Hypertext Transfer Protocol
  ⊞ HTTP/1.1 401 Authorization Required\r\n
    Date: Thu, 1 Jan 1970 00:00:14 GMT\r\n
    Server: $ProjectRevision: 4.2.2.12 $\r\n
    Www-Authenticate: Basic realm=admin\r\n
    Content-Type: application/xml\r\n
  ⊞ Content-Length: 0\r\n
    \r\n
```

```
000  00 00 14 00 ee 18 00 00  10 16 9e 09 a0 00 de 9c   ........ ........
010  64 00 00 42 08 02 d5 00  10 0b a9 80 bb 44 00 1d   d..B.... .....D..
020  c9 01 99 99 00 1d c9 01  99 99 d0 15 aa aa 03 00   ........ ........
030  00 00 08 00 45 00 00 f0  8f a5 00 00 40 06 59 0f   ....E... ....@.Y.
040  c0 a8 08 01 c0 a8 08 02  00 50 e7 ab d4 42 16 5d   ........ .P...B.]
050  31 3e 7f 8c 50 18 0a 17  8b f3 00 00 48 54 54 50   1>..P... ....HTTP
060  2f 31 2e 31 20 34 30 31  20 41 75 74 68 6f 72 69   /1.1 401  Authori
070  7a 61 74 69 6f 6e 20 52  65 71 75 69 72 65 64 0d   zation R equired.
080  0a 44 61 74 65 3a 20 54  68 75 2c 20 31 20 4a 61   .Date: T hu, 1 Ja
090  6e 20 31 39 37 30 20 30  30 3a 30 30 3a 31 34 20   n 1970 0 0:00:14
0a0  47 4d 54 0d 0a 53 65 72  76 65 72 3a 20 24 50 72   GMT..Ser ver: $Pr
```

File: "C:\Users\AGrovas\AppData\Local\Te... Packets: 4950 Displayed: 7 Marked: 0 Dropped: 0

Figure 13: HTTP Traffic Exchanged (Screen 4)


AirPcap USB wireless capture adapter nr. 00 - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Expression... Clear Apply
Wireless Settings... Decryption Keys...

Filter: ((wlan.ra==10:0b:a9:80:bb:44||wlan.ta==10:0b:a9:80:bb:44||wlan.addr=

02.11 Channel: 2462 [BG 11] Channel Offset: 0 FCS Filter: All Frames Wireshark

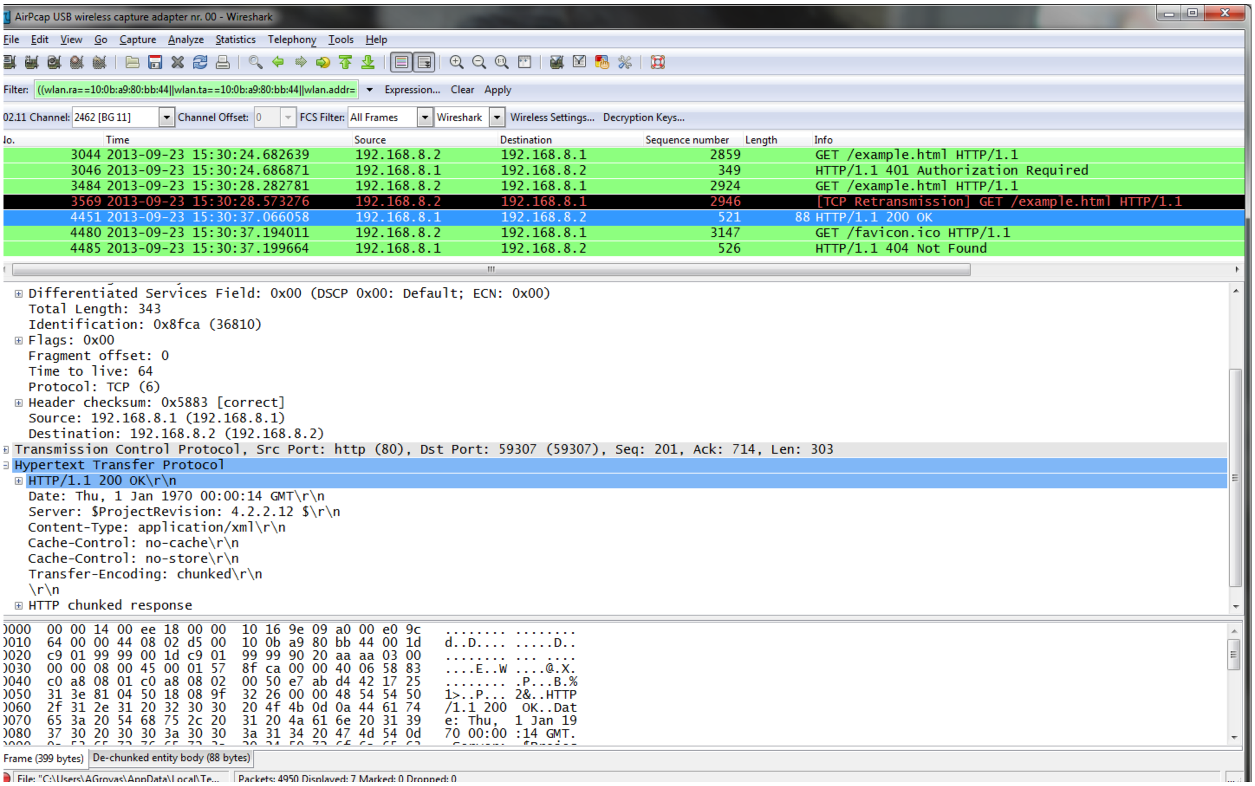
```
lo.         Time                        Source          Destination     Sequence number  Length  Info
      3044 2013-09-23 15:30:24.682639   192.168.8.2     192.168.8.1                2859          GET /example.html HTTP/1.1
      3046 2013-09-23 15:30:24.686871   192.168.8.1     192.168.8.2                 349          HTTP/1.1 401 Authorization Required
      3484 2013-09-23 15:30:28.282781   192.168.8.2     192.168.8.1                2924          GET /example.html HTTP/1.1
      3569 2013-09-23 15:30:28.573276   192.168.8.2     192.168.8.1                2946          [TCP Retransmission] GET /example.html HTTP/1.1
      4451 2013-09-23 15:30:37.066058   192.168.8.1     192.168.8.2                 521       88 HTTP/1.1 200 OK
      4480 2013-09-23 15:30:37.194011   192.168.8.2     192.168.8.1                3147          GET /favicon.ico HTTP/1.1
      4485 2013-09-23 15:30:37.199664   192.168.8.1     192.168.8.2                 526          HTTP/1.1 404 Not Found
```

```
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 416
    Identification: 0x076a (1898)
  ⊞ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (6)
  ⊞ Header checksum: 0x609a [correct]
    Source: 192.168.8.2 (192.168.8.2)
    Destination: 192.168.8.1 (192.168.8.1)
⊞ Transmission Control Protocol, Src Port: 59307 (59307), Dst Port: http (80), Seq: 338, Ack: 201, Len: 376
⊟ Hypertext Transfer Protocol
  ⊞ GET /example.html HTTP/1.1\r\n
    Host: 192.168.8.1\r\n
    Connection: keep-alive\r\n
  ⊞ Authorization: Basic YWRtaW46YWRtaW4=\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/29.0.1547.76 Safari/537.36\r\n
    Accept-Encoding: gzip,deflate,sdch\r\n
    Accept-Language: en-US,en;q=0.8\r\n
    \r\n
```

```
)000  00 00 14 00 ee 18 00 00  10 16 9e 09 a0 00 ee 9c   ........ ........
)010  64 00 00 52 08 01 d5 00  00 1d c9 01 99 99 10 0b   d..R.... ........
)020  a9 80 bb 44 00 1d c9 01  99 99 c0 b6 aa aa 03 00   ...D.... ........
)030  00 00 08 00 45 00 01 a0  07 6a 40 00 80 06 60 9a   ....E... .j@...`.
)040  c0 a8 08 02 c0 a8 08 01  e7 ab 00 50 31 3e 7f 8c   ........ ...P1>..
)050  d4 42 17 25 50 18 10 ea  f9 46 00 00 47 45 54 20   .B.%P... .F..GET 
)060  2f 65 78 61 6d 70 6c 65  2e 68 74 6d 6c 20 48 54   /example .html HT
)070  54 50 2f 31 2e 31 0d 0a  48 6f 73 74 3a 20 31 39   TP/1.1.. Host: 19
)080  32 2e 31 36 38 2e 38 2e  31 0d 0a 43 6f 6e 6e 65   2.168.8. 1..Conne
)090  63 74 69 6f 6e 3a 20 6b  65 65 70 2d 61 6c 69 76   ction: k eep-aliv
)0a0  65 0d 0a 41 75 74 68 6f  72 69 7a 61 74 69 6f 6e   e..Autho rization
```

File: "C:\Users\AGrovas\AppData\Local\Te... Packets: 4950 Displayed: 7 Marked: 0 Dropped: 0

Figure 14: HTTP Traffic Exchanged (Screen 5)


AirPcap USB wireless capture adapter nr. 00 - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Expression... Clear Apply
Wireless Settings... Decryption Keys...

Filter: ((wlan.ra==10:0b:a9:80:bb:44||wlan.ta==10:0b:a9:80:bb:44||wlan.addr=

02.11 Channel: 2462 [BG 11] Channel Offset: 0 FCS Filter: All Frames Wireshark

```
Io.         Time                          Source          Destination       Sequence number  Length   Info
        3044 2013-09-23 15:30:24.682639   192.168.8.2     192.168.8.1                  2859           GET /example.html HTTP/1.1
        3046 2013-09-23 15:30:24.686871   192.168.8.1     192.168.8.2                   349           HTTP/1.1 401 Authorization Required
        3484 2013-09-23 15:30:28.282781   192.168.8.2     192.168.8.1                  2924           GET /example.html HTTP/1.1
        3569 2013-09-23 15:30:28.573276   192.168.8.2     192.168.8.1                  2946           [TCP Retransmission] GET /example.html HTTP/1.1
        4451 2013-09-23 15:30:37.066058   192.168.8.1     192.168.8.2                   521        88 HTTP/1.1 200 OK
        4480 2013-09-23 15:30:37.194011   192.168.8.2     192.168.8.1                  3147           GET /favicon.ico HTTP/1.1
        4485 2013-09-23 15:30:37.199664   192.168.8.1     192.168.8.2                   526           HTTP/1.1 404 Not Found
```

```
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 343
    Identification: 0x8fca (36810)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (6)
  ⊞ Header checksum: 0x5883 [correct]
    Source: 192.168.8.1 (192.168.8.1)
    Destination: 192.168.8.2 (192.168.8.2)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 59307 (59307), Seq: 201, Ack: 714, Len: 303
⊟ Hypertext Transfer Protocol
  ⊞ HTTP/1.1 200 OK\r\n
    Date: Thu, 1 Jan 1970 00:00:14 GMT\r\n
    Server: $ProjectRevision: 4.2.2.12 $\r\n
    Content-Type: application/xml\r\n
    Cache-Control: no-cache\r\n
    Cache-Control: no-store\r\n
    Transfer-Encoding: chunked\r\n
    \r\n
  ⊞ HTTP chunked response
```

```
)000  00 00 14 00 ee 18 00 00  10 16 9e 09 a0 00 e0 9c   ........ ........
)010  64 00 00 44 08 02 d5 00  10 0b a9 80 bb 44 00 1d   d..D.... .....D..
)020  c9 01 99 99 00 1d c9 01  99 99 90 20 aa aa 03 00   ........ ... ....
)030  00 00 08 00 45 00 01 57  8f ca 00 00 40 06 58 83   ....E..W ....@.X.
)040  c0 a8 08 01 c0 a8 08 02  00 50 e7 ab d4 42 17 25   ........ .P...B.%
)050  31 3e 81 04 50 18 08 9f  32 26 00 00 48 54 54 50   1>..P... 2&..HTTP
)060  2f 31 2e 31 20 32 30 30  20 4f 4b 0d 0a 44 61 74   /1.1 200  OK..Dat
)070  65 3a 20 54 68 75 2c 20  31 20 4a 61 6e 20 31 39   e: Thu,  1 Jan 19
)080  37 30 20 30 30 3a 30 30  3a 31 34 20 47 4d 54 0d   70 00:00 :14 GMT.
```

Frame (399 bytes) | De-chunked entity body (88 bytes)

File: "C:\Users\AGrovas\AppData\Local\Te... | Packets: 4950 Displayed: 7 Marked: 0 Dropped: 0