

# SPWF Series of WiFi Modules

Subsystem Products Group



# Benefits of Wireless OEM Modules

- **Off-the-shelf RF plug&play solutions**
  - Enable short **time-to-market** product development
  - **No RF specific knowledge** is required for the integration of the modules in the target application
- **Cost-effective** for design of multiple platforms or multiple versions of the same platform
  - Save 8-12 months in design cycle
  - Significantly reduce engineering and production costs
- **Pre-certified** RF modules
  - Reduce the effort and certification cost on the customer side



# ST Wireless OEM Modules Portfolio

## BlueTooth



### SPBT Series

HW	SW Protocol
SPBT2532	BT 2.1
SPBT2632	BT3.0



## 802.15.4/ZigBee



### SPZB32W Series

HW	SW Protocol
SPZB32W1	SimpleMAC
SPZB32W2	ZigBeeIP



## WiFi



### SPWF Series

HW	SW Protocol
SPWF01S	TCP/IP

NEW Development



**ST core leading industry technology inside**

# Drivers for an Explosive WiFi Growth

## ■ Momentum of Wi-Fi technology

- 10% of the world's population uses Wi-Fi
- Double-digit growth year over year
- Ubiquitous in home, enterprise, industry, education and government environments

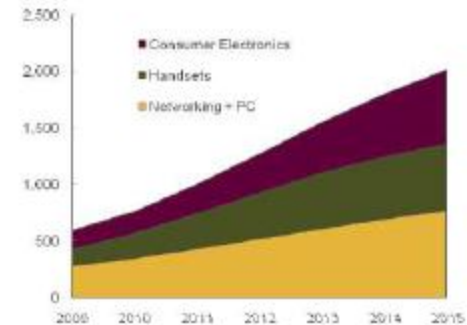
## ■ Prevalence of Wi-Fi based products

- Home market deployment, installed in 200M homes Worldwide
- Consumer electronic deployment, 100% in Smart Phones, major growth
- PC deployment, ~80% laptops, 100% tablets

## ■ Characteristics of the technology

- Easy path to the internet (TCP/IP based)
- Secured for machine-to-machine
- Flexible over multiple applications

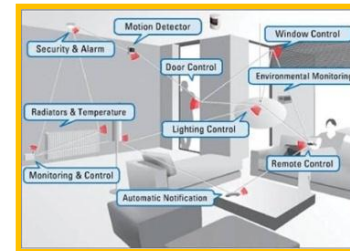
Wi-Fi IC Shipments, by Device Type  
in Millions - ABI Research, 4Q 2010



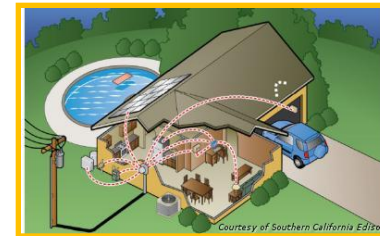
# Target Applications

- Smart Appliances
- Industrial Control and Data Acquisition
- Home Automation & Home Energy
- Home Security Systems
- Wireless Sensors
- Cable Replacement
- Medical Equipments

## Home/Building Automation



## Smart Energy/ Smart Grid



## Industrial

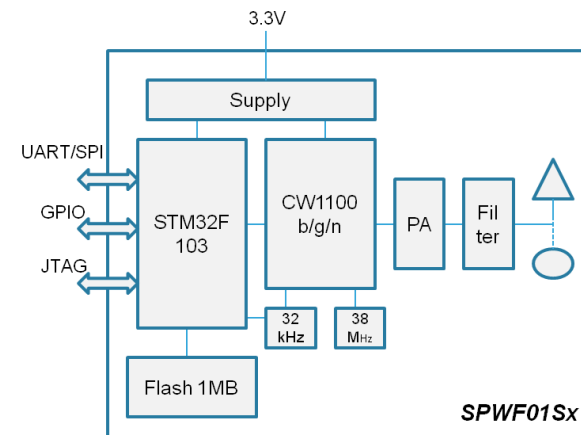


# SPWF01Sx Modules

## Features

- **Radio.** 2.4 GHz IEEE 802.11b/g/n
- **Micro.** STM32 ARM Cortex-M3
- **Memory.** 64KB RAM, 1.5 MB Flash
- **Size (mm).** 26.92 x 15.24 x 2.35
- **Interfaces.**
  - Serial (UART, I2C, SPI)
  - 16 GPIOs
  - JTAG
- **XTAL.** Integrated 32kHz to support low power modes
- Side pads **SMD**
- **Temperature.** Industrial temperature range
- **Antenna Options.** Integrated Antenna/U.fl. Connector
- **Certifications:**
  - FCC, IC and CE certified
  - ROHS Compliant
- **Software.** Multiple Stacks Available
  - Full Stack
    - AT
    - SDK

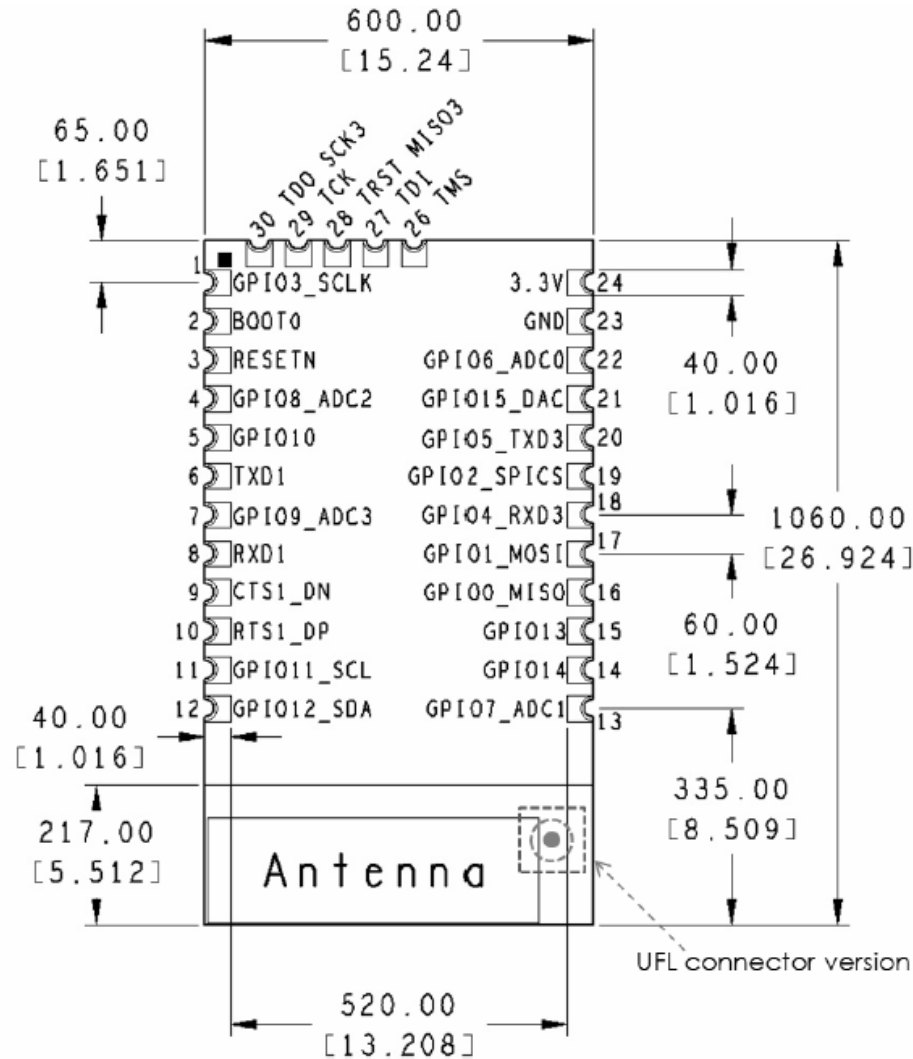
## Serial To WiFi Module



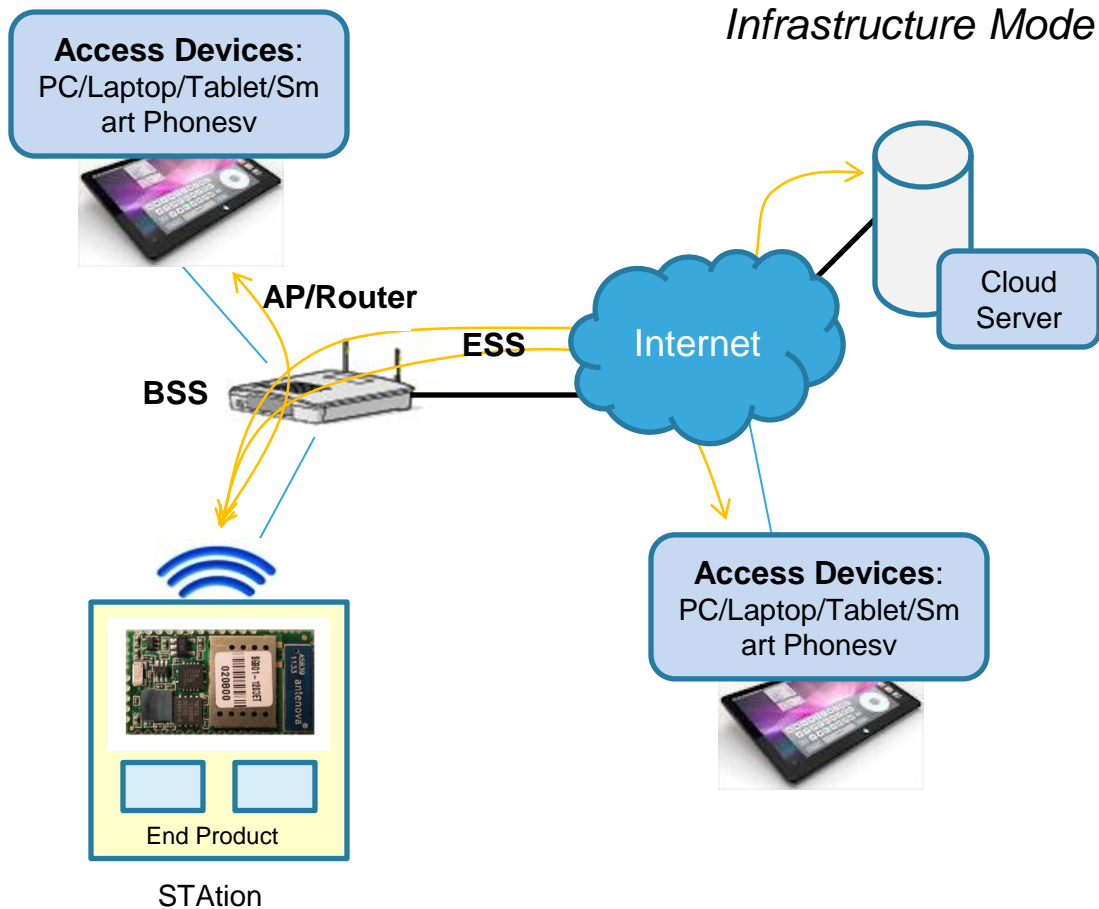
Part Number	Antenna Options	SW Library
SPWF01SA.11	Chip Antenna	Full Stack (AT)
SPWF01SC.11	U.FL	Full Stack (AT)

# SPWF01Sx Footprint

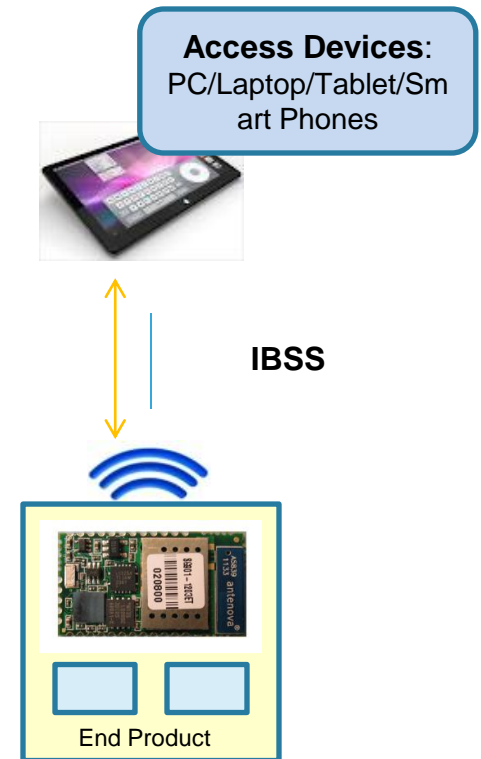
Dimensions: L: 26.92mm W: 15.24mm H: 2.35mm



# Supported System Configurations

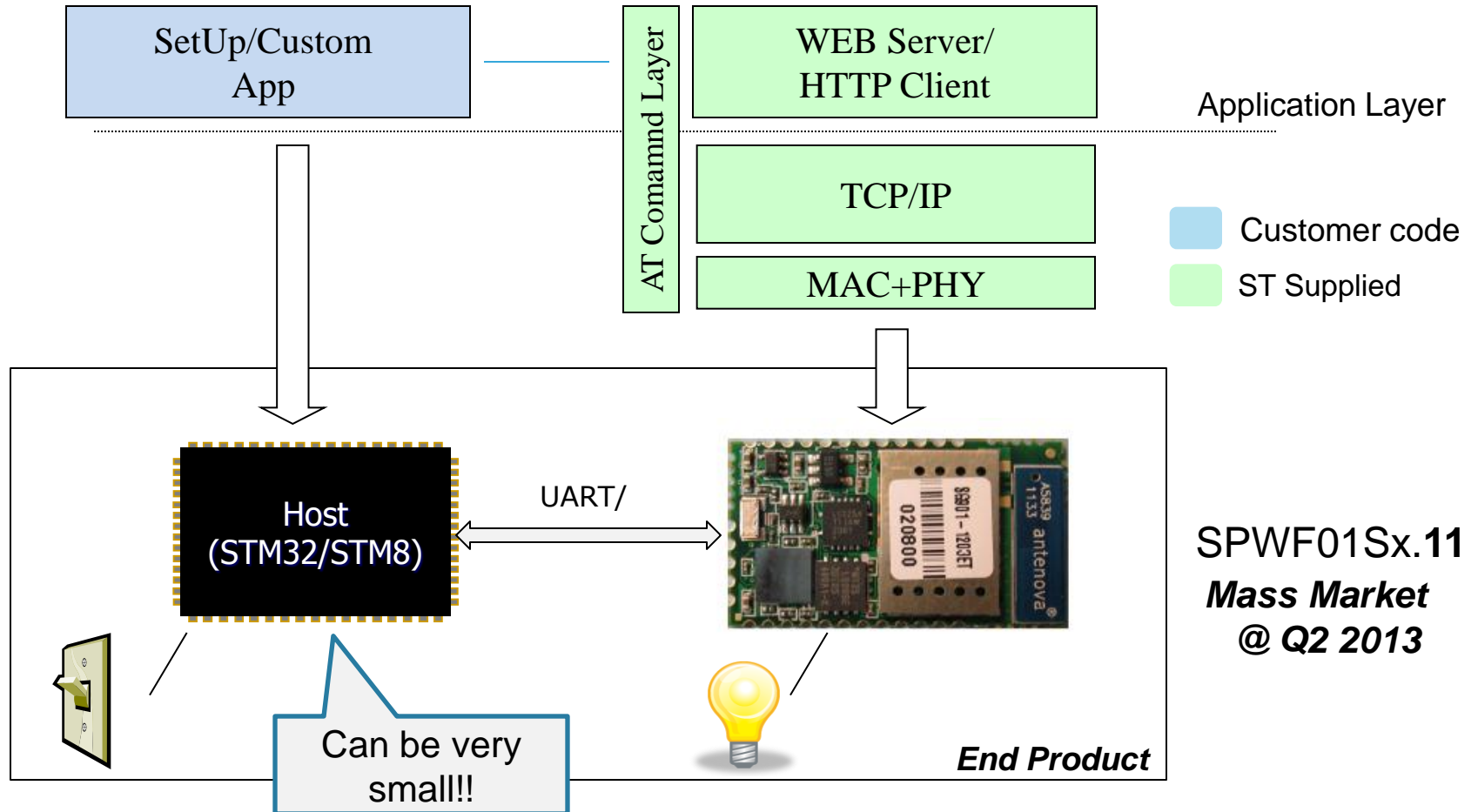


*Ad-Hoc/Wi-Fi Direct Mode*





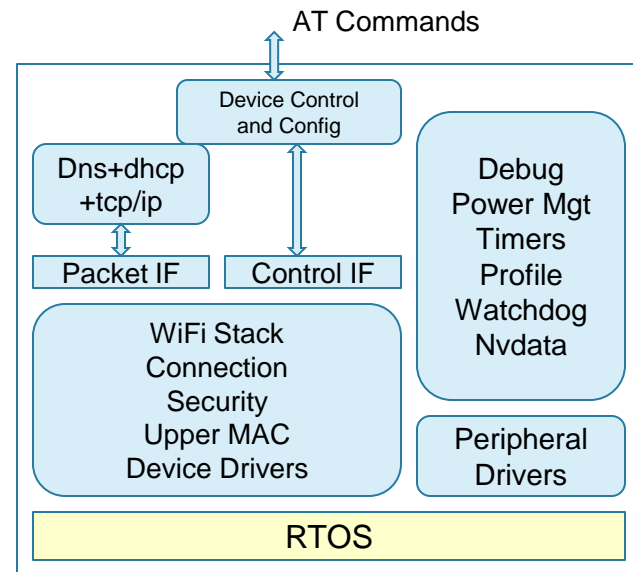
Enable the use of the module as a Network Coprocessor



# AT Full Stack Features

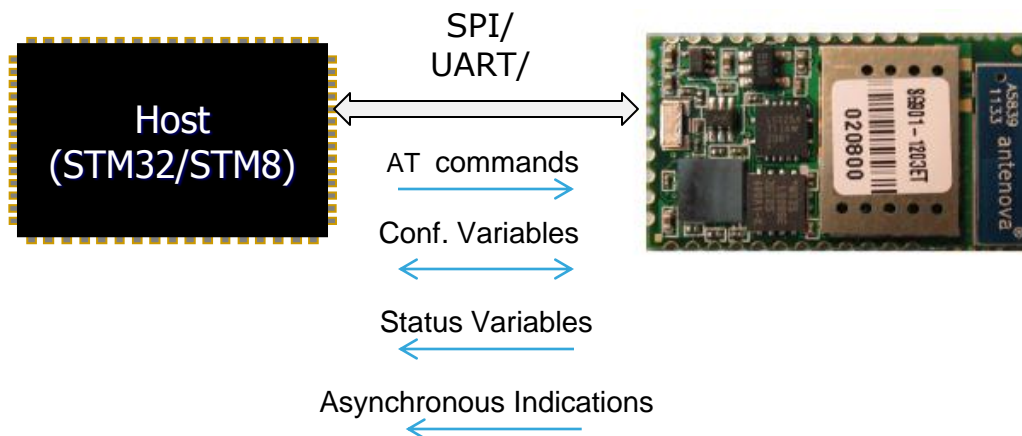
## Available

- Rich AT command set for RS-232
- Standards-compliant 802.11b/g/n operation
- IBSS and BSS Station operation modes
- Advanced Power Saving Modes
- Wireless security (WEP, WPA/WPA2-PSK)
- Full IPv4 stack + TCP + UDP
- DHCP client and DNS Client
- Field update via WiFi and RS-232
- Rich AT command set for RS-232 control
- TCP/UDP Socket Client
- Built-in application utilities:
  - web server
  - http client (http get) (pull data mode)
  - “http post via get” (push data mode)



# AT Full Stack Command Interface

Utilities	Notes
<b>AT-style commands</b>	Multiple Categories: i.e. Utilities, Configuration, Network, GPIO, Files Management
<b>Configuration Variables</b>	Multiple SetUp categories: i.e security, network, applications.
<b>Status Variables</b>	Radio, channels ...
<b>Asynchronous Indications</b>	Radio/Protocol/Status Indication Run-Time Messages that are echoed on the serial port



# AT Full Stack Commands

## Utilities

AT	Attention
AT+S.HELP	Display Help Text
AT+S.FWUPDATE	Perform a firmware update
AT+S.WIFI	Enable/Disable WiFi device
AT+CFUN	Comm Function (Reset)
AT+S.MFGTEST	Perform manufacturing tests
AT+S.PEMDATA	Configure certificate store
AT+S.ECHO	Send data out serial port
AT+S.HTTPPDFSUPDATE	Update static HTTPD Filesystem

## Configuration

AT+S.GCFG	Get configuration value
AT+S.SCFG	Set configuration value
AT+S.SSIDTXT	Set a textual SSID
AT+S.STS	Report current status/statistic
AT&V	Display all configuration values
AT&F	Restore factory default settings
AT&W	Save current settings
AT+S.NVW	Write production settings

## Network

AT+S.PING	Send a ping to a specified host
AT+S.SCAN	Channels Scan
AT+S.HTTPGET	Issue an HTTP GET
AT+S.ROAM	Trigger WiFi reassociation sequence

## File Management

AT+S.FSC	Create a file
AT+S.FSA	Append to an existing file
AT+S.FSD	Delete an existing file
AT+S.FSL	List existing filename(s)
AT+S.FSP	Print the contents of an existing file

## Socket

AT+S.SOCKON	Open a network socket
AT+S.SOCKOS	Open Serial Port
AT+S.SOCKW	Write len bytes of data to socket
AT+S.SOCKR	Return len bytes of Data from socket
AT+S.SOCKQ	Query pending data
AT+S.SOCKC	Close socket

## GPIO

AT+S.GPIOC	Configure General Purpose I/O
AT+S.GPIOR	Query General Purpose Input
AT+S.GPIOW	Set General Purpose Output

# AT Full Stack Variables

## Configuration Variables

Production Data

UART Configuration

Power save configuration

Security Configuration

Radio Setup

Protocol Setup

Mode Setup

## Status Variables

Protocol Statistics

Radio Statistics

Module Status

Security Setup

Modules Configuration

*Remotely accessible  
In the “`config.shtml`” page*

*Remotely accessible  
In the “`status.shtml`” page*

# «Web Server» Use Mode



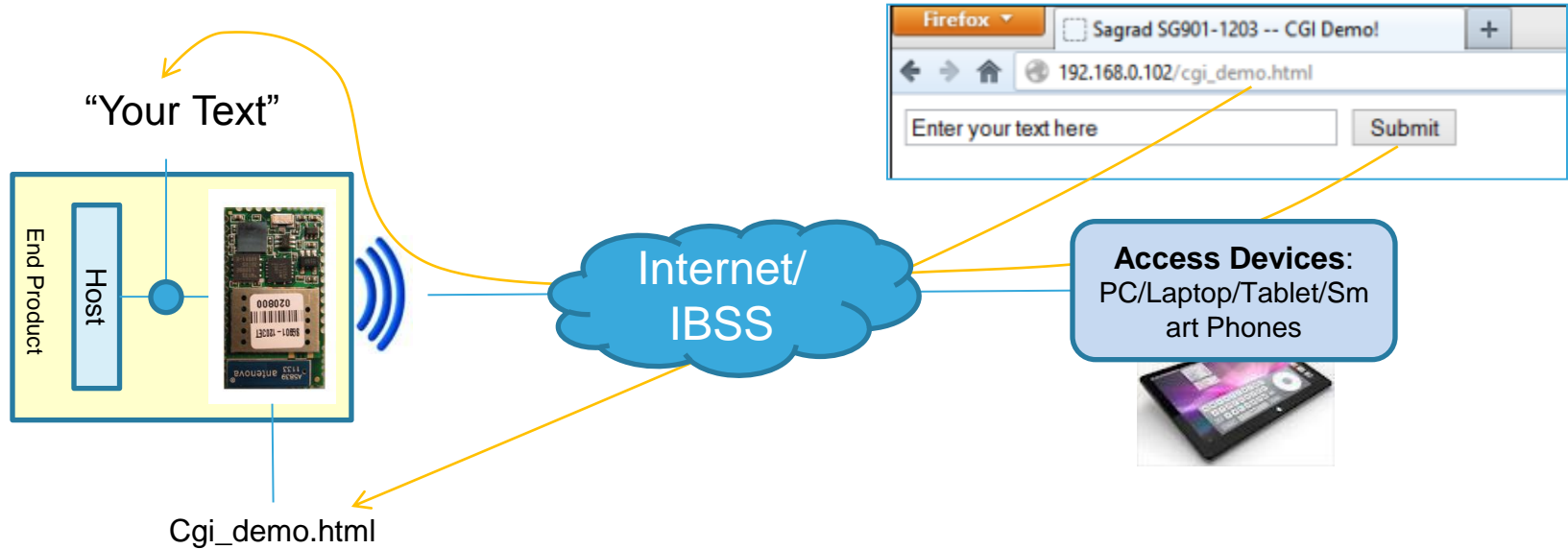
HTML Pages/Ascii files resident in the memory of the module



<b>Built-in HTML files (Resident in the Flash)</b>	status.shtml	Remote Access to Status parameters (i.e. radio, channels)
	index.html	Home page
	config.shtml	Remote Access to the config variables values
	404.html	Error page
<b>Custom html/ascii files (resident in the RAM)</b>	Any name	Can be loaded via the serial port
<b>Custom files (resident in the Flash)</b>	Any name	Can be loaded remotely

# «Post Data» Use Mode

15



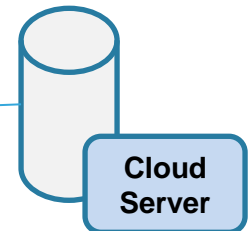
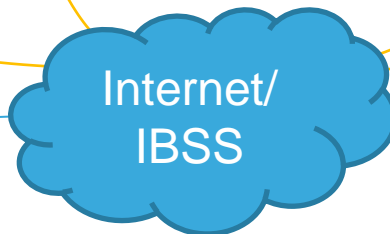
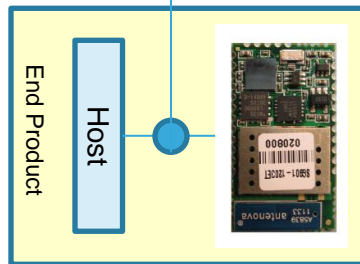
A built-in html page “cgi\_demo.html” allows to remotely push characters on the serial port from a remote browser

# «HTTP Client» Use Mode

- The module implements the **HTTP get** method by means of an AT command
- The HTTP GET feature performs a single HTTP request to the specified host and path. The server response is printed on the UART enabled.

At+s.httpget = www.st.com, /aaaa.jsp

```
<!DOCTYPE html PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html>  
<head>  
<!--[if !IE]-->  
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>  
<meta name="description" content="STMicroelectronics"/>  
<meta name="keywords" content="STMicroelectronics"/>  
<meta name="url" content=""/>  
<meta name="google-site-verification" content="N14Ud4L7AFNMAj2tVkl1xb0210_6d511q1r2a21"/>  
<meta name="WI_S_country" content="Global"/>  
<meta name="WI_S1" content="ST Home"/>  
</!--[endif]-->  
<!--[if IE]-->  
<link type="text/css" rel="stylesheet" href="http://www.st.com/Internet/com/css/style.css"/>  
<link type="text/css" rel="stylesheet" href="http://www.st.com/Internet/com/css/Ptexpol/styles.css"/>  
</!--[endif]-->  
<!--[if !IE]-->  
<link type="text/css" rel="stylesheet" href="http://www.st.com/Internet/com/css/186.css"/>  
</!--[endif]-->  
<!--[if gte IE 7]-->  
<link type="text/css" rel="stylesheet" href="http://www.st.com/Internet/com/css/186.css"/>  
</!--[endif]-->  
</html>
```





- The Socket interface allows communication via TCP, UDP and UART by means of AT commands.
- The module is a client socket only and it can open up to 8 TCP/UDP/UART connections.

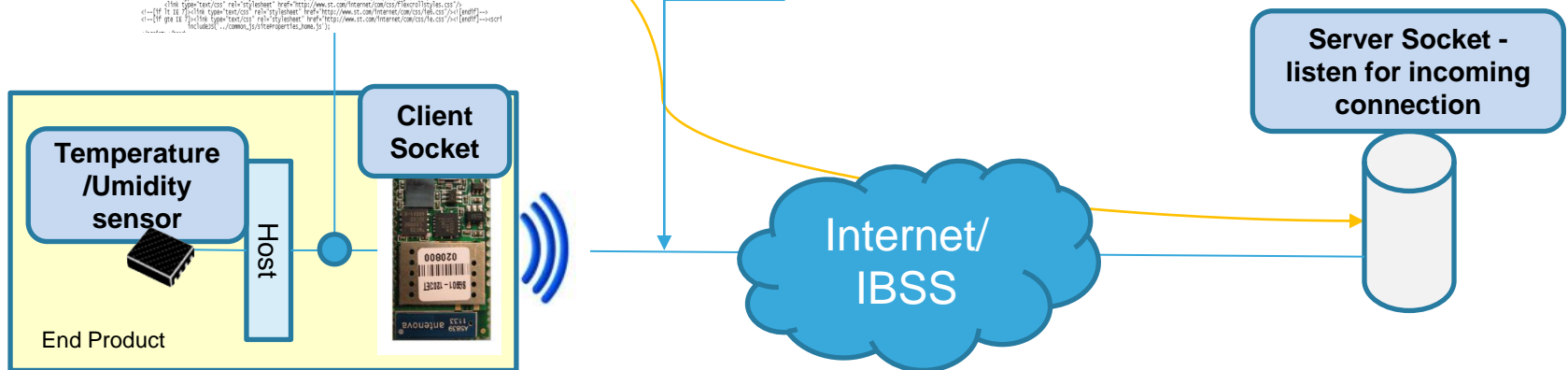
## 1) Open a Socket (returns a SocketID)

AT+S.SockOn (Port, IPserver, TCP/UDP)

## 2) Read/Write on the Socket

AT+S.SockW (SocketID, SensorData)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<!--#include virtual="/ssi/include/ssiutils3" -->
<!--#include virtual="/ssi/include/ssiutils" -->
<!--#include virtual="/ssi/include/ssiutils2" -->
<!--#include virtual="/ssi/include/ssiutils4" -->
<!--#include virtual="/ssi/include/ssiutils5" -->
<!--#include virtual="/ssi/include/ssiutils6" -->
<!--#include virtual="/ssi/include/ssiutils7" -->
<!--#include virtual="/ssi/include/ssiutils8" -->
<!--#include virtual="/ssi/include/ssiutils9" -->
<!--#include virtual="/ssi/include/ssiutils10" -->
<!--#include virtual="/ssi/include/ssiutils11" -->
<!--#include virtual="/ssi/include/ssiutils12" -->
<!--#include virtual="/ssi/include/ssiutils13" -->
<!--#include virtual="/ssi/include/ssiutils14" -->
<!--#include virtual="/ssi/include/ssiutils15" -->
<!--#include virtual="/ssi/include/ssiutils16" -->
<!--#include virtual="/ssi/include/ssiutils17" -->
<!--#include virtual="/ssi/include/ssiutils18" -->
<!--#include virtual="/ssi/include/ssiutils19" -->
<!--#include virtual="/ssi/include/ssiutils20" -->
<!--#include virtual="/ssi/include/ssiutils21" -->
<!--#include virtual="/ssi/include/ssiutils22" -->
<!--#include virtual="/ssi/include/ssiutils23" -->
<!--#include virtual="/ssi/include/ssiutils24" -->
<!--#include virtual="/ssi/include/ssiutils25" -->
<!--#include virtual="/ssi/include/ssiutils26" -->
<!--#include virtual="/ssi/include/ssiutils27" -->
<!--#include virtual="/ssi/include/ssiutils28" -->
<!--#include virtual="/ssi/include/ssiutils29" -->
<!--#include virtual="/ssi/include/ssiutils30" -->
<!--#include virtual="/ssi/include/ssiutils31" -->
<!--#include virtual="/ssi/include/ssiutils32" -->
<!--#include virtual="/ssi/include/ssiutils33" -->
<!--#include virtual="/ssi/include/ssiutils34" -->
<!--#include virtual="/ssi/include/ssiutils35" -->
<!--#include virtual="/ssi/include/ssiutils36" -->
<!--#include virtual="/ssi/include/ssiutils37" -->
<!--#include virtual="/ssi/include/ssiutils38" -->
<!--#include virtual="/ssi/include/ssiutils39" -->
<!--#include virtual="/ssi/include/ssiutils40" -->
<!--#include virtual="/ssi/include/ssiutils41" -->
<!--#include virtual="/ssi/include/ssiutils42" -->
<!--#include virtual="/ssi/include/ssiutils43" -->
<!--#include virtual="/ssi/include/ssiutils44" -->
<!--#include virtual="/ssi/include/ssiutils45" -->
<!--#include virtual="/ssi/include/ssiutils46" -->
<!--#include virtual="/ssi/include/ssiutils47" -->
<!--#include virtual="/ssi/include/ssiutils48" -->
<!--#include virtual="/ssi/include/ssiutils49" -->
<!--#include virtual="/ssi/include/ssiutils50" -->
<!--#include virtual="/ssi/include/ssiutils51" -->
<!--#include virtual="/ssi/include/ssiutils52" -->
<!--#include virtual="/ssi/include/ssiutils53" -->
<!--#include virtual="/ssi/include/ssiutils54" -->
<!--#include virtual="/ssi/include/ssiutils55" -->
<!--#include virtual="/ssi/include/ssiutils56" -->
<!--#include virtual="/ssi/include/ssiutils57" -->
<!--#include virtual="/ssi/include/ssiutils58" -->
<!--#include virtual="/ssi/include/ssiutils59" -->
<!--#include virtual="/ssi/include/ssiutils60" -->
<!--#include virtual="/ssi/include/ssiutils61" -->
<!--#include virtual="/ssi/include/ssiutils62" -->
<!--#include virtual="/ssi/include/ssiutils63" -->
<!--#include virtual="/ssi/include/ssiutils64" -->
<!--#include virtual="/ssi/include/ssiutils65" -->
<!--#include virtual="/ssi/include/ssiutils66" -->
<!--#include virtual="/ssi/include/ssiutils67" -->
<!--#include virtual="/ssi/include/ssiutils68" -->
<!--#include virtual="/ssi/include/ssiutils69" -->
<!--#include virtual="/ssi/include/ssiutils70" -->
<!--#include virtual="/ssi/include/ssiutils71" -->
<!--#include virtual="/ssi/include/ssiutils72" -->
<!--#include virtual="/ssi/include/ssiutils73" -->
<!--#include virtual="/ssi/include/ssiutils74" -->
<!--#include virtual="/ssi/include/ssiutils75" -->
<!--#include virtual="/ssi/include/ssiutils76" -->
<!--#include virtual="/ssi/include/ssiutils77" -->
<!--#include virtual="/ssi/include/ssiutils78" -->
<!--#include virtual="/ssi/include/ssiutils79" -->
<!--#include virtual="/ssi/include/ssiutils80" -->
<!--#include virtual="/ssi/include/ssiutils81" -->
<!--#include virtual="/ssi/include/ssiutils82" -->
<!--#include virtual="/ssi/include/ssiutils83" -->
<!--#include virtual="/ssi/include/ssiutils84" -->
<!--#include virtual="/ssi/include/ssiutils85" -->
<!--#include virtual="/ssi/include/ssiutils86" -->
<!--#include virtual="/ssi/include/ssiutils87" -->
<!--#include virtual="/ssi/include/ssiutils88" -->
<!--#include virtual="/ssi/include/ssiutils89" -->
<!--#include virtual="/ssi/include/ssiutils90" -->
<!--#include virtual="/ssi/include/ssiutils91" -->
<!--#include virtual="/ssi/include/ssiutils92" -->
<!--#include virtual="/ssi/include/ssiutils93" -->
<!--#include virtual="/ssi/include/ssiutils94" -->
<!--#include virtual="/ssi/include/ssiutils95" -->
<!--#include virtual="/ssi/include/ssiutils96" -->
<!--#include virtual="/ssi/include/ssiutils97" -->
<!--#include virtual="/ssi/include/ssiutils98" -->
<!--#include virtual="/ssi/include/ssiutils99" -->
<!--#include virtual="/ssi/include/ssiutils100" -->
</html>
```



- **Serial To WiFi** 802.11b/g/n OEM Module
- **Plug&Play** Solution
- Very **Small** Form Size Factor
- FCC/IC/CE **certified**
- Multiple **Antenna** Options
- **Low Power** Use Modes Available
- **Industrial** Operating Temperature Range
- **Infrastructure** Mode
- **AhHoc/WiFi Direct** Mode
- **“Full TCP/IP Stack”** SW Library with
  - Built-in **Wi-Fi** security
  - Built-in **TCP/IP** stack
  - Built-in **DHCP, DNS**
  - Built-in **HTTP** server/client
- Rich **AT-like** commands for host usage