IP-TO-WIFI

APPLICATION PROGRAMMING GUIDE

Reference:	GS-IP2WF-APG
Version:	SP-1.0
Date:	*19-Jul-11*

Version	Date	Remarks
1.0	24 Nov 2010	Initial release.
1.1	May 2011	Updated for release 2.3.1

GainSpan Corporation
125 South Market Street, Suite 400
San Jose, CA 95113
U.S.A.

+1 (408) 673-2900

info@GainSpan.com
www.GainSpan.com

Specifications, features, and availability are subject to change without notice.

Table of Contents

Figures

Tables

1 System Overview

1.1 Purpose

This document describes the operation and serial command interface for the GainSpan System-On-Chip (SOC) ***IP2WiFi Adapter.*** The IP2WiFi Adapter enables embedded devices with a UART/SPI interface to gain access an 802.11-compliant (Wi-Fi®) wireless network connection using only serial commands.

1.2 Scope

This document reviews the architecture of the IP2WiFi software and provides the programmer with necessary command syntax required to manage the Wi-Fi interface and to send and receive network messages. This document assumes that the reader is generally familiar with GainSpanproducts and the operation and management of 802.11 wireless devices.

1.3 Overview

The IP2WiFi stack is used to provide Wi-Fi capability to any devices having a serial interface. This approach offloads WLAN to the Wi-Fi chip, allowing host microcontrollers to communicate with other hosts on the network using a Wi-Fi wireless link. The host processor can use serial commands to configure the IP2WiFi Adapter and to create wireless and network connections.

1.4 Terminology

Table 1: Glossary of Terms

Term	*Explanation*
AP	Access Point
API	Application Programmer's Interface
BSSID	Basic Service Set Identifier
DHCP	Dynamic Host Configuration Protocol
IP	Internet Protocol
MTU	Maximum Transfer Unit
PSK	Pre-shared key
RSSI	Received Signal Strength Indication
SSID	Service Set Identifier
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver/Transmitter
UDP	User Datagram Protocol
WEP	Wired Equivalent Privacy
WPA	Wi-Fi Protected Access

1.5 Standards

The following standards and conventions are considered in this design:

- IEEE 802.11 b/g
- ITU V.25ter AT Command Set **Error! Reference source not found.**

2 Interface Architecture

The overall architecture of the IP2WiFi interface is depicted in Figure 1. Tx and Rx Data Handlers pass messages to, and from, the Wi-Fi interface. Commands related to management of the IP2WiFi interface and the network connections are intercepted by a Command Processor. A Serial Data Handler translates data to and from a UART/SPI-compatible format.

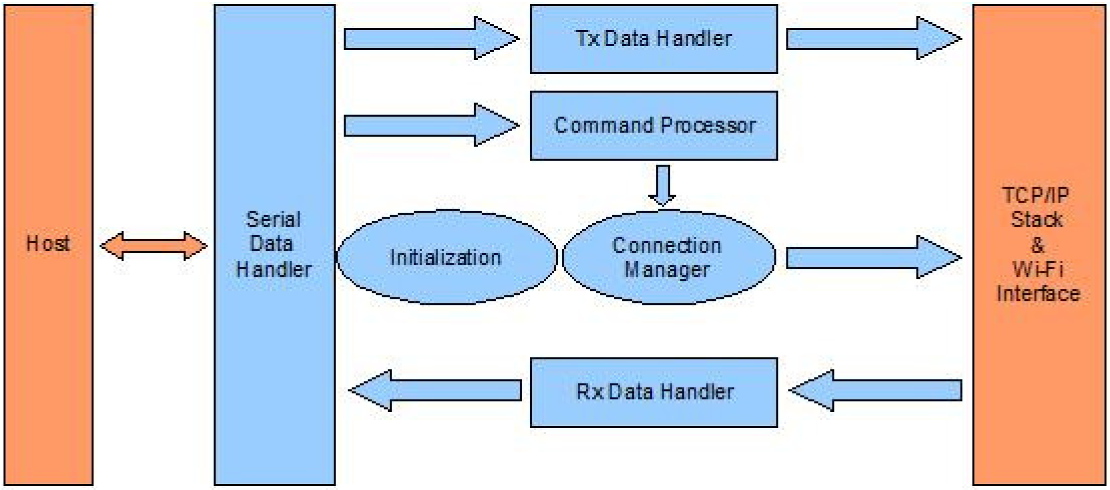

Figure 1: Overall Architecture of the Adapter

The system is composed of the following modules:

- System Initialization (section 3.2)
- Command Processor (section 3.3)
- Data Handlers (section 3.4)
- Serial Data Handler (section 3.5)
- Wireless Connection Management (section 3.6)

The software for the IP2WiFi Adapter is driven using a state machine. Upon powering on, the required initialization of all the modules is performed and then the state machine is entered. This state machine is event-driven and processes the events received from either the serial port or from the Wi-Fi/interface as well as internal events from its own modules. The state machine calls the appropriate handler for a given event per the current state.

The IP2WiFi Adapter has two distinct operating modes. In the default ***command processing operating*** mode, commands to configure and manage the interface are sent over the serial interface. In this default mode, the node accepts commands entered by the Host CPU and processes each of the commands. All commands are available in this mode. The User may establish a data connection here and send data.

In ***data processing*** mode, data can be sent to, or received from, the WiFi interface.

For each mode, configuration parameters are stored in non-volatile memory. In addition to factory-default parameter values, two user-defined profiles (0 and 1) are available. The parameter set to be used is determined by a user command (section 4.5.3).

3 Adapter Description

3.1 Serial Interface Detection

Upon startup, the IP2WiFi adaptor performs an auto detection of serial interface protocol. This detection is done through the GPIO pin 26. If this GPIO is “high” during startup, meaning the serial interface is SPI, otherwise the adaptor enables the UART interface as the IP2WiFi interface to the host.

3.2 System Initialization

Upon startup, the IP2WiFi interface performs the following actions.

- During the initialization process, the module will search for a saved configuration file. If a saved configuration file is available, it is loaded from non-volatile memory. If no saved configuration file is discovered, the default settings will be applied. If there are no saved parameters, the factory-default configuration is loaded.
- The IP2WiFi application is initialized based on the profile settings.
- The interface enters the command processing state.

Upon power-up, the UART interface defaults to 115200 baud, using 8 bit characters with no parity bits and one stop bit. Similarly SPI interface defaults to Mode#0 (CPL=0, CPH=0). Any changes to this configuration that were made in a previous session using the `ATB` command (section 4.2.1) will be lost when power is lost. To make changes in the UART/SPI parameters that will persist across power cycling, the relevant changes must be saved into the power-on profile using `AT&W` (section 4.5.1) and `AT&Y` (section 4.5.3).

3.2.1 External PA Auto Detection

Upon startup, the IP2WiFi interface performs an auto detection of External PA. This detection is done through the GPIO pin 12. If this GPIO is “high” during startup, meaning the external PA is present, the adaptor enables the external PA and forces the adaptor to go to standby for a moment and comes back just to make any changes effective for the external PA configuration.

3.2.2 Profile Definition

The configuration parameter values that define the behavior of the Adapter are grouped into Profiles. These profiles are stored in non-volatile memory when not in use. The default configuration supports two Profiles. The contents of a profile are listed in Table 2.

Table 2: Profile Parameters

Parameter	Values	Reference
General Wireless Parameters		
802.11 Operating Mode	BSS, IBSS	4.6.6
Transmit Power Configuration		4.7.1313
802.11 Transmit Retry Count		4.6.13
Power Save Mode	Enabled, Disabled	4.7.1111
802.11 Radio Mode	Enabled, Disabled	4.7.100
Wireless Interface Security Configuration		
Authentication Mode	Open, Shared	4.7.1
PSK Valid	Valid, Invalid	4.7.5
PSK-SSID	Any valid SSID; used for PSK key computation.	4.7.5
WEP Key Configuration		3
WPA Passphrase		4
UART Configuration		
Echo Mode	Enabled, Disabled	4.1.2
Verbose Mode	Enabled, Disabled	4.1.3
Bits Per Character	5,6,7,8	4.2.1
Number of Stop Bits	1,2	4.2.1
Parity Type	No, Odd, Even	4.2.1
Software Flow Control Mode	Enabled, Disabled	4.2.2
Hardware Flow Control Mode	Enabled, Disabled	4.2.3
Baud Rate		4.2.1

3.3 Command Processing Mode

In command mode, the application receives commands over the serial port. Commands are processed line by line. "Verbose Mode", when referring to commands being executing, displays of status of any command executed in ASCII (human readable) format. When the verbose mode is disabled, the output will simply be in numeric digits, with each digit indicating a particular status. Verbose Mode is enabled by default.

- If "echo" is enabled then each character is echoed back on the serial port
- Each command is terminated with a ***carriage return*** <CR> or ***line feed*** <LF>
- Each response is started with a carriage return <CR> and line feed<LF>, with the exception of the responses to the following commands:

  a. The response to the following group of commands starts with a line feed <LF> only:

     AT+WPAPSK=<SSID>,<Passphrase>

  a) The response to the following group of commands starts with a line feed and carriage return: <LF><CR>.

  AT+SETTIME=<dd/mm/yyyy>,<hh:mm:ss>

- Each response is terminated with a carriage return <CR> and line feed <LF>
- If the characters "A" and "/" are entered at the beginning of a line (after <CRLF>), then the previous command is executed
- Once a complete line (ending with <CR or LF>) is entered, then the command contained therein is processed and an appropriate response returned

Unless otherwise specified, if verbose mode is enabled, then the response to a successful command is the characters "OK". The response to an unsuccessful command is the word "ERROR", followed by a detailed error message, if available. If verbose mode is disabled, command responses is numerical with OK having a value of 0 and error codes represented by positive integers.

The commands are described in Section 4. Possible response codes are described in 3.6.3.

3.4 Data Handling

In Data Processing Mode, data transfers are managed using ***escape sequences***. Each escape sequence starts with the ASCII character 27 (0x1B); this is equivalent to the ESC key. The encoding of data and related commands are described in the following pages. This encoding is used for both transmitted and received data. The encoding of data is described below:

```
<Esc>:R:<Length>:<Ethernet packet>
```

```
Where the Ethernet Packet is:
<DstAddr><SrcAddr><EtherType><Payload>
```

The contents of `< >` are byte or byte stream.

- `Length` is the size of `ethernet packet`
- `DstAddr` is the destination MAC address
- `SrcAddr` is the source MAC address
- `EtherType` is the type of the Ethernet packet. For example, for BACNET-over-Ethernet, `EtherType` is 0x0000.
- `Payload` is the raw data

3.4.1 Unsolicited Data Handling

In Unsolicited Data Mode (data transmission without association), data transfer is managed using ***escape sequences***. Each escape sequence starts with the ASCII character 27 (0x1B), equivalent to the ESC key. The encoding of data is described below. This encoding is used for transmitted data only. The unsolicited data transmission Enable command must be issued before sending unsolicited data through the Adapter.

The format of an unsolicited data frame is:

<ESC>D/d<PayLoad>

The PayLoad contentis byte or byte stream.

3.4.2 Software Flow Control

Software flow control works only with ASCII data transfers and cannot be used for binary data.

If software flow control is enabled, and the interface receives an XOFF character from the serial host, it stops sending to the host until it receives an XON character. If the Adapter is receiving data over the wireless connection during the time that XOFF is enabled, it is possible for the wireless buffer to become full before XON is received. In such a case, data from the network will be lost.

If software flow control is enabled, then the interface sends an XOFF character to the host when it will be unable to service the serial port. The XON character is sent when the interface is once again able to accept data over the serial port.

> Note: *With initialization, the Adapter treats the serial channel as clear with no restrictions on data transmission or reception; no explicit XON is transmitted by the Adapter or required from the Host, even if flow control is enabled.*

3.4.3 Hardware Flow Control

The Hardware Flow control is a handshake mechanisum between the Serial host and S2W adapter on UART interface, which use two additional CTS and RTS connections. This feature prevent the UART hardware FIFO overflow on S2W adapter due to high speed data transmission from/to the S2W adapter. If hardware flow control is enabled, an RTS/CTS handshake will occur between the serial host and the Adapter. This is a hardware feature and available only for UART interface.

The S2W adapter uses both CTS and RTS signals as low for ready to send or receive data from serial host.

3.5 Serial Data Handling

The Serial Data Handler receives and transmits data to and from the hardware serial controller. Data read from the serial port is passed to:

- The command processor in command mode
- The Tx data handler in data mode

Then Data is transferred on the serial port from:

- The command processor in order to output responses to commands
- The Rx data handler in order to output incoming packets
- The auto connection handler in order to output incoming data
- The connection manager in order to output status indications
- The wireless connection manager in order to output status indications

3.6 Wireless Network Management

3.6.1 Scanning

The IP2WiFi interface instructs the Wi-Fi radio to scan for access points and *ad hoc* networks with a specified SSID and/or channel for a specified scan period. Scanning can be performed to find networks with a specific SSID networks operating on a specific radio channel or a combination of these constraints.

3.6.2 Association

The IP2WiFi interface performs all the actions required to join an infrastructure IP network:

- Scan for a specific AP (AT+WS, section 0)
- Authenticate the specified network using the configured authentication mode (AT+WAUTH, section 4.7.1)
- Associate to the AP (AT+WA, section 4.6.7)
- Perform security negotiation if required
- Change state to Wireless Connected

In *ad hoc* mode, the interface can:

- Scan for a specified *ad hoc* Network
- Join the *ad hoc* network, if it exists
- If the *ad hoc* network does not exist, create a new *ad hoc* network to join
- Perform security negotiation, if required
- Change state to Wireless Connected

3.6.3 Response Codes

The possible responses sent by the Adapter to the serial host are enumerated in Table 3.

Table 3: Response Codes.

No	ASCII CHAR	Response	ASCII STRING	Meaning
1	0	S2W_SUCCESS	"OK"	Command Request Success.
2	1	S2W_FAILURE	"ERROR"	Command Request Failed.
3	2	S2W_EINVAL	"ERROR: INVALID INPUT"	Invalid Command or Option or Parameter.
7	6	S2W_ENOTSUP	"ERROR: NOT SUPPORTED"	Operation or Feature not supported.
10	9	S2W_LINK_LOST	"DISASSOCIATED"	Not associated to a wireless network.
11	10	S2W_DISASSO_EVT	“\r\nDisassociation Event\r\n”	Wireless network association lost.
12	11	S2W_STBY_TMR_EVT	"\n\rOut of StandBy-Timer\r\n"	Wake up from Standby due to RTC timer expiration.
13	12	S2W_STBY_ALM_EVT	"\n\rOut of StandBy-Alarm\r\n"	Wake up from Standby due to receipt of an Alarm signal.
14	13	S2W_DPSLEEP_EVT	"\r\nOut of Deep Sleep\r\n"	Wake from Deep Sleep
15	14	S2W_BOOT_UNEXPECTED_EVT	"\r\nUnExpected Warm Boot(Possibly Low Battery)\r\n"	Unexpected reset. Possible reasons: external reset or low battery

3.6.4 Enhanced Asynchronous Messages

NO	Message	SubType	Meaning
1	Disassociation Event	3	Wireless network association

			lost.
2	Out of StandBy-Timer	4	Wake up from Standby due to RTC timer expiration.
3	Out of StandBy-Alarm	5	Wake up from Standby due to receipt of an Alarm signal.
4	Out of Deep Sleep	6	Wake from Deep Sleep.
5	UnExpected Warm Boot(Possibly Low Battery)	7	Unexpected reset. Possible reasons: external reset or low battery.

3.6.5 Exception Messages

The possible exception messages sent by the Adapter to the serial host are enumerated in Table 4:

Table 4: Exception Messages.

No	*ASCII STRING*	*Meaning*
1	\n\rAPP Reset-Wlan SW Reset\r\n	Adapter reset due to WLAN processor software reset.
2	"\n\rAPP Reset-APP SW Reset\r\n"	Adapter reset due to app processor software reset...
3	\n\rAPP Reset-Wlan-Wd\r\n	Adapter reset due to WLAN processor watchdog.
4	\n\rAPP Reset-App-Wd\r\n	Adapter reset due to app processor watchdog
5	\n\rAPP Reset-Wlan Except\r\n	Adapter reset due to WLAN processor software abort or assert.
6	\n\rAPP Reset-FW-UP-FAILURE\r\n	Adapter reset due to firmware upgrade failure.

7	\n\rAPP Reset-FW-UP-SUCCESS\r\n	Adapter reset due to firmware upgrade success.
8	\n\rAPP Reset-FW-UP-RECOVERY\r\n	Adapter reset due to firmware upgrade failure with one of the flash image updated successfully.

If the exception is due to one of the WLAN wd/SW Reset/Except, then the adapter send memory dump information of its WLAN registers to the serial host starts with the message \r\n---MEM-DUMP-START:\r\n and end with the message \n\r---MEM-DUMP-END:\r\n.

3.6.6 Boot Messages

The possible boot messages sent by the Adapter to the serial host are enumerated in Table 6.

Table 5: Boot Messages.

NO	ASCII STRING	Meaning
1	\r\n IP2WiFi APP\r\n	Normal IP2WiFi adapter boot message with internal PA.
2	\r\nIP2WiFi APP-Ext.PA\r\n	Normal IP2WiFi adapter boot message with external PA.
3	\r\n Factory Default CheckSum Error\r\n	The factory default section contains invalid data. This comes along with either one of the above boot message.

3.6.7 SSID and PassPhrase

Rules:

1- The IP2WiFi adapter accepts the following ASCII characters for SSID and passphrase.

Category	Accepted Characters
Numerical	0-9
Alphabets	a-z and A-Z
Special characters	SP ! # $ % & ' ( ) * + , - . / : ; < = > ? @ [\ ] ^ _ ` { | } ~ ”

Note: SP = space

2- The SSID or PassPhrase parameter may be captured within or without double quotation marks (“SSID”).

3- The quotation mark (“) may not be use as the first character of the ssid or passphrase.

4- If comma (,) is a part of the SSID, then SSID parameter needs to be framed with double quotation marks (“SS,ID”).

Expected SSID	Input SSID	Remarks
TEST	TEST	Valid (satisfies rule 2)
TEST	“TEST”	Valid (satisfies rule 2)
TE”ST	TE”ST	Valid (satisfies rule 3)
TE”ST	“TE”ST”	**Invalid** (*breaks rule 3*)
TE,ST	“TE,ST”	Valid (satisfies rule 4)
TE,ST	TES,T	**Invalid** (*breaks rule 4*)
TE,S”T	“TE,S”T”	**Invalid** (*breaks rule 3 and 4*)

4 Commands for Command Processing Mode

This section provides a list of IP2WiFi commands and their effects. Formatting and processing of commands was described in section 3.3 above. Parameters are generally ASCII characters, e.g. ATEn with n=1 is the series of ASCII characters 'A', 'T', 'E', and '1'. Where some parameters are optional, mandatory parameters are denoted by < > and optional parameters by [ ]. If a parameter is mandatory, any associated sub-parameters are also mandatory; sub-parameters of an optional parameter are optional. Parameters must always be provided in the order given in the command description. When an optional parameter is not supplied, the comma delimiters must still be included in the command. Every command starts with the characters "AT"; any other initial characters will cause an error to be returned.

Command Response: In most cases, valid commands return the characters `OK` if verbose mode is enabled and `0` verbose mode is not enabled. Invalid inputs return `ERROR: INVALID INPUT` if verbose is enabled and `2` if it is not. Exceptions to this rule are noted explicitly below.

4.1 Command Interface

4.1.1 Interface Verification

The command AT can be issued to verify that the interface is operating correctly; it should return a successful response `OK` (or `0` if verbose mode is disabled).

4.1.2 Echo

The command to enable/disable echo is

```
ATEn
```

If *n* is 0, echo is disabled and if *n* is 1, echo is enabled.

If echo is enabled, every character received on the serial port is transmitted back on the serial port. This command returns the standard command response (section 4) to the serial interface. By default echo is enabled in IP2WiFi adapter.

4.1.3 Verbose

The command to enable/disable verbose responses is

```
ATVn
```

If *n* is 0, verbose responses is disabled and if *n* is 1, verbose responses is enabled.

If verbose mode is disabled, the status response is in the form of numerical response codes. If verbose mode is enabled, the status response is in the form of ASCII strings. Verbose Mode is enabled by default.

This command returns the standard command response (section 4) to the serial interface.

4.2 UART Interface Configuration

4.2.1 UART Parameters

The command to set the UART communication parameters is

```
ATB=<baudrate>[[,<bitsperchar>][,<parity>][,<stopbits>]]
```

All standard baud rates are supported.

Allowed baud rates include: 9600, 19200, 38400, 57600, 115200, 230400,460800 and 921600.

Parity is *n* for no parity, *e* for even parity and *o* for odd parity.

Allowed values are 5, 6, 7 or 8 bits/character, with 1 or 2 stop bits (1.5 in the case of a 5-bit character).

The new UART parameters take effect immediately. However, they are stored in RAM and will be lost when power is lost unless they are saved to a profile using `AT&W` (section 4.5.1). The profile used in that command must also be set as the power-on profile using `AT&Y` (section 4.5.3).

This command returns the standard command response (section 4) to the serial interface with the new uart configuration.

4.2.2 Software Flow Control

The command to configure software flow control is

```
AT&Kn
```

If *n* is 0 for software flow control to be disabled and if *n* is 1 for software flow control to be enabled.

The use of software flow control is described in section 3.4.2 above. This command returns the standard command response (section 4) to the serial interface.

4.2.3 Hardware Flow Control

The command to configure hardware flow control is

```
AT&Rn
```

If *n* is 0, hardware flow control is disabled. If *n* is 1, hardware flow control is enabled. This command returns the standard command response (section 4) to the serial interface.

The use of software flow control is described in section 3.4.3 above.

4.3 SPI Interface Configuration

4.3.1 SPI Parameters

The command to set the SPI clock phase and clock polarity parameter is

```
AT+SPICONF=<clockpolarity>, <clockphase>
```

If clock polarity is 0, then inactive state of serial clock is low.

If clock polarity is 1, then inactive state of serial clock is high.

If clock phase is 0, then data is captured on the first toggling edge of the serial clock (clock phase zero), after the falling edge of slave select signal.

If clock phase is 1, then data is captured on the second edge of the serial clock (clock phase 180), after the falling edge of slave select signal.

Default is clock polarity 0 and clock phase 0.

The new SPI parameters take effect after node reset/restart. However, they are stored in RAM and will be lost when power is lost unless they are saved to a profile using `AT&W` (section 4.5.1). The profile used in that command must also be set as the power-on profile using `AT&Y` (section 4.5.3).

This command returns the standard command response (section 4) to the serial interface with the new SPI configuration.

4.4 Identification information

The command to obtain identification information from the application is

```
ATIn
```

n is the ID of the information to obtain. The responses are listed in Table . These responses are provided as ASCII strings.

Table 4: Application Information.

Information ID	Description
0	OEM identification
1	Hardware version

Information ID	Description
2	Software version

4.5 IP to Wi-Fi Configuration Profiles

Adapter configuration parameters can be stored and recalled as a Profile; see 3.2.2 for a detailed description of the profile parameters.

4.5.1 Save Profile

The command to save the current profile is

```
AT&Wn
```

n shall either be 0 for profile 0 or 1 for profile 1. (Higher values are allowed if more profiles are configured at compile time.)

Upon deployment of this command, the current configuration settings are stored in non-volatile memory under the specified profile. Note that, in order to ensure that these parameters are restored after power cycling the adapter, the command `AT&Y` (section 4.5.3) must also be issued, using the same profile number selected here.

This command returns the standard command response (section 4) or ERROR, (1, if verbose disabled) if the operation failed.

4.5.2 Load Profile

The command to load a profile is

```
ATZn
```

n shall either be 0 for profile 0 or 1 for profile 1. (Higher values are allowed if more profiles are configured at compile time.)

Upon deployment of this command, the currently configured settings are set to those stored in non-volatile memory under the specified profile.

This command returns the standard command response (section 4) or ERROR, (1, if verbose disabled) if the operation failed.

4.5.3 Selection of Default Profile

The command to select the default profile is

```
AT&Yn
```

n shall either be 0 for profile 0 or 1 for profile 1. (Higher values are allowed if more profiles are configured at compile time.)

The settings from the profile that is chosen as the default profile are loaded from non-volatile memory when the device is started.

In addition to the standard status responses, this command returns `ERROR` or `1`, based on verbose settings, if a valid input cannot be executed.

4.5.4 Restore to Factory Defaults

The command to reset to factory defaults is

```
AT&F
```

Upon deployment of this command, the current configuration variables are reset to the factory defaults. These defaults are defined by macro values in the configuration header, and can be modified at compile time. Issuing this command resets essentially all configuration variables ***except*** the IEEE MAC address. Only the command `AT+NMAC` (section 4.6.1) changes the MAC address.

This command returns the standard command response (section 4) to the serial interface.

4.5.5 Output current configuration

The command to output the configuration is

```
AT&V
```

Upon deployment of this command, the current configuration and the configuration of the saved profiles are output on the serial port in ASCII format. In addition to the standard command response (section 4). The details of the profile parameters are described in section 3.1.3.

4.6 Wi-Fi Interface Configuration

4.6.1 MAC Address Configuration

The command to set the configuration is

```
AT+NMAC=<MAC ADDRESS>
```

Upon deployment of this command, the Adapter sets the IEEE MAC address as specified. The format of the MAC address is an 8-byte colon-delimited hexadecimal number. An example is shown below:

```
AT+NMAC=00:1d:c9:00:01:a2
```

The MAC address is used in the 802.11 protocol to identify the various nodes communicating with an Access Point and to route messages within the local area (layer 2) network. Fixed MAC addresses issued to network interfaces are hierarchically structured and are intended to be globally unique. Before issuing a MAC address to a given Adapter, ensure that no other local device is using that address.

The MAC address supplied in the `AT+NMAC` command is saved to flash memory, and will be used on each subsequent cold boot (from power off) or warm boot (from Standby).

The alternative command

```
AT+NMAC2=<MAC ADDRESS>
```

Stores the MAC address in RTC RAM. Each warm boot (from Standby) will use the MAC address stored in RTC RAM (from the most recent `AT+NMAC2=` command), but if power to the device is lost, the next cold boot will use the MAC address stored in flash memory (from the most recent `AT+NMAC=` command). This command is particularly useful in cases where writing to flash memory is undesirable.

4.6.2 Output MAC Address

The command to output the configuration is

```
AT+NMAC=?
```

Upon deployment of the command, the Adapter outputs the current MAC address of the wireless interface to the serial port, in addition to the usual status responses. The alternate command is

```
AT+NMAC2=?
```

may also be used, and returns the same value.

4.6.3 Regulatory Domain Configuration

The command to set the regulatory domain is

AT+WREGDOMAIN=<Regulatory Domain>

This command sets the regulatory domain as per the Regulatory Domain parameter passed. The supported regulatory domains are:

- FCC → supported Channel range is 1 to 11.
- ETSI → supported Channel range is 1 to 13.
- TELEC → supported Channel range is 1 to 14.

The corresponding values for this regulatory domain that needs to be passed as the parameter are:

- FCC : 0
- ETSI : 1
- TELEC : 2

The default regulatory domain is FCC. The Regulatory domain set is required only once since it is being updated in the flash. This command returns the standard command response (section 4) to the serial interface.

4.6.4 Regulatory Domain Information

The command to get the configured regulatory domain in the IP2WiFi adaptor is

AT+WREGDOMAIN=?

Upon reception of the command, the Adapter outputs the current Regulatory domain of the wireless interface to the serial port as the following format:

REG_DOMAIN=FCC or ETSI or TELEC, in addition to the standard command responses.

4.6.5 Scanning

The command to scan for access points or *ad hoc* networks is

```
AT+WS[=<SSID>[,<BSSID>][,<Channel>][,<Scan Time>]]
```

Upon deployment of the command, the Adapter scans for networks with the specified parameters and displays the results. Scanning can be performed to find networks with specific SSID or specific BSSID or in a particular operating channel, or a combination of these parameters. Scanning for a specific SSID or BSSID employs active scanning, in which probe requests are transmitted with the SSID and/or BSSID fields being filled appropriately.

The SSID is a string containing between 1 and 32 ASCII characters, Refer section 3.7.6 for details.

This command does not support scan based on the BSSID.

The Scan Time is in units of Milliseconds with a range of 0-65535.

Upon completion, the adapter reports the list of networks and information for each network along with the standard command response (section 4) one per line, in the following format to the serial interface

<space><BSSID>,<space><SSID>,<space><Channel>,<space><space><Type><space>,<space><RSSI><space>,<space><Security>

Also this sends out the total number of networks found as follows (after send out the above information to the serial interface).

"No. Of AP Found:<n><CR><LF>"

Where n is the total number of networks found during scan.

`Type` is `INFRA` for an infrastructure network and `ADHOC` for an ad hoc network.

4.6.6 Mode

The command to set the wireless mode:

```
AT+WM=n
```

If *n* is 0, the mode is set to *infrastructure*; if *n* is 1, the mode is set to *ad hoc*.

If n is 2, the mode is set to limited AP so that the adapter can act as a limited wireless Access Point. IP2WiFi Adapter uses infrastructure (0) as the default mode.

This command returns the standard command response (section 4) to the serial interface.

4.6.7 Associate with a Network, or Start an Ad Hoc or Infrastructure (AP) Network

The command to associate to an access point, to join an ad hoc network or to create an ad hoc/ infrastructure (AP)/ network is

```
AT+WA=<SSID>[,[<BSSID>][,<Ch>]]
```

In infrastructure mode (section 4.6.6, n is 0), the adapter will attempt to associate with the requested network. In ad hoc mode (section 4.6.6, n is 1), if a network with the desired SSID or channel or both is not found, then a new network is created. However, if the BSSID was specified in the request and the applicable BSSID is not found, the Adapter will report an error and will not create an ad hoc network.

In AP mode (section 4.6.6, n is 2), the adapter creates an infrastructure network (limited AP) with the ssid passed.

The SSID is a string containing between 1 and 32 ASCII characters. Refer section 3.6.7 for details.

In adhoc and AP mode the radio should be on in active mode (section 4.7.10)

In addition to the usual status responses, this command will return `ERROR` or `1` (depending on verbose status) if a valid command was issued but association failed.

4.6.8 Disassociation

The command to disassociate is

```
AT+WD
```

Upon deployment of this command, the interface disassociates from the current infrastructure or *ad hoc* network, if associated. This command returns the standard command response (section 4) to the serial interface.

4.6.9 WPS

The command to associate to an AP using WPS is

AT+WWPS=<METHOD>[,PIN]

- METHOD is push button (1) or pin (2).
- PIN is the pin for PIN method.

Upon execution of this command, the adaptor uses either push button or pin method as per the METHOD parameter to associate to the WPS enabled AP. The PIN is optional and is valid for pin method only.

In addition to the usual status responses this command returns the following information to the serial host on success case:

- SSID=<ssid>
- CHANNEL=<channel>
- PASSPHRASE=<passphrase> for wpa/wpa2 security;
- WEP KEY=<wep key> for WEP security;
- WEPKEYINDEX=<key index> for WEP security.
- ;

The above information is send to the serial interface with one information element per line.

This command returns `ERROR` or `1` (depending on verbose status) if a valid command was issued but WPS failed.

4.6.10 Status

The command to retrieve information about the current network is

```
AT+WSTATUS
```

Upon deployment of this command, the adapter reports the current network configuration to the serial host:

- Mode;
- Channel;
- SSID;
- BSSID;
- Security;

if the adaptor associated to an Access Point. If no association is present, the error message `NOT ASSOCIATED` is returned, in addition to the usual status response.

4.6.11 Get RSSI

The command obtains the current RSSI is

```
AT+WRSSI=?
```

Upon deployment of this command, the current RSSI value (in dBm) is output on the serial port in ASCII format, in addition to the status response.

4.6.12 Get Transmit Rate

The command obtains the current transmit rate is

```
AT+WRATE=?
```

Upon deployment of this command, the current transmit rate used is output on the serial port in ASCII format.

4.6.13 Set Retry count

The command to set the wireless retry count is

```
AT+WRETRY=<retrycount>
```

Upon deployment of this command, the current wireless retry count is set to the supplied value. The transmission retry count determines the maximum number of times a data packet is retransmitted, if an 802.11 ACK is not received. The valid range is 4 to 7 with default value 5. *Note that the count includes the initial transmission attempt.*

This command returns the standard command response (section 4) to the serial interface.

4.7 Wi-Fi Security Configuration

4.7.1 Authentication Mode

The command to choose the authentication mode to use is

```
AT+WAUTH=n
```

n is:

- 0- None
- 1 – Open
- 2 – Shared with WEP

Note that this command configures the authentication mode, but any required encryption key must be set using the key commands described on the following page. This authentication mode command is specific to WEP encryption; if WPA/WPA2 operation is employed, the authentication mode may be left at the default value “None”. This command returns the standard command response (section 4) to the serial interface.

4.7.2 Security Configuration

The S2w adapter supports a strict security configuration. The command required to configure this feature is

AT+WSEC= n

Where *n* is:

- 0 – Auto security (All)

- 1 – Open security
- 2 – Wep security
- 4 – Wpa-psk security
- 8 – Wpa2-psk security
- 16 – Wpa Enterprise
- 32 – Wpa2 Enterprise

The s2w adapter supports either one of the above value with default security configuration as auto. This strict security compliance is not applicable for WPS feature. This command returns the standard command response (section 4) to the serial interface.

4.7.3 WEP Keys

The command to set WEP keys is

```
AT+WWEPn=<key>
```

n is the key index, between 1 and 4, and key are either 10 or 26 hexadecimal digits corresponding to a 40-bit or 104-bit key. Some examples:

```
AT+WWEP1=123456abdc
```

```
AT+WWEP3=abcdef12345678901234567890
```

Upon receiving a valid command, the relevant WEP key is set to the value provided. This command returns the standard command response (section 4) to the serial interface.

4.7.4 WPA-PSK and WPA2-PSK Passphrase

The command to set the WPA-PSK and WPA2-PSK passphrase is

```
AT+WWPA=<passphrase>
```

The passphrase is a string containing between 8 and 63 ASCII characters, used as a seed to create the WPA ***pre-shared key*** (PSK).

If the comma (,) is a part of the passphrase, then the passphrase parameter is to be framed in double quotation marks (“passphrase”). Refer section 3.7.6 for details.

Upon receiving the command, the PSK passphrase is reset to the value provided. This command returns the standard command response (section 4) to the serial interface.

4.7.5 WPA-PSK and WPA2-PSK KEY CALCULATION

Computation of the PSK from the passphrase is complex and consumes substantial amounts of time and energy. To avoid recalculating this quantity every time the adapter associates, the adapter provides the capability to compute the PSK once and store the resulting value. The key value is stored in the SRAM copy of the current profile; the profile needs to be saved in flash memory for this value to persist during a transition to Standby. The command to compute and store the value of the WPA/WPA2 PSK, derived from the passphrase and SSID value is

```
AT+WPAPSK=<SSID>,<PASSPHRASE>
```

The passphrase is a string containing between 8 and 63 ASCII characters, used as a seed to create the PSK. The SSID is a string of between 1 and 32 ASCII characters. Refer section 3.6.7 for details. Each Parameter of the above command separated by comma (,). If the comma(,) is a part of the SSID or PASSPHRASE, then SSID and PASSPHRASE parameters is to be framed in double quotation marks (“SSID”,”PASSPHRASE”).

When the command is issued, the adapter immediately responds with `Computing PSK from SSID and PassPhrase`. When it is complete, the adapter will issue the usual `OK` or `0`. Invalid inputs will result in `ERROR: INVALID INPUT` or `2`, as usual.

Upon receiving the command, the adapter computes the PSK from the SSID and passphrase provided, and stores those values in the current profile. The current profile parameters PSK Valid, PSK-SSID, and WPA Passphrase are updated, and can be queried with `AT&V` (4.5.5). The next time the adapter associates to the given SSID, the PSK value is used without being recalculated.

After the PSK has been computed, the commands `AT&W` (to save the relevant profile) and `AT&Y` (to ensure that the profile containing the new PSK is the default profile) should be issued. The PSK will then be available when the adapter awakens from Standby. Refer to sections 4.5.1 and 4.5.3 for more information on profile management.

4.7.6 WPA-PSK and WPA2-PSK KEY

The command to configure the WPA / WPA2 PSK key directly is

```
AT+WPSK=<PSK>
```

This command directly sets the pre-shared key as provided. The argument is a 32-byte key formatted as an ASCII hexadecimal number. Any other length or format is considered invalid.

Example:

```
AT+WPSK= 0001020304050607080900010203040506070809000102030405060708090001
```

After the PSK has been entered, the commands `AT&W` (to save the relevant profile) and `AT&Y` (to ensure that the profile containing the new PSK is the default profile) should be issued. The PSK will then be available when the adapter awakens from Standby. Refer to sections 4.5.1 and 4.5.3 for more information on profile management.

4.7.7 EAP-Configuration

The command to configure the EAP-security is

AT+ WEAPCONF=<Outer Authentication>,<Inner Authentication>,<user name>,<password>

Upon execution of this command, the adaptor set the Outer authentication, Inner authentication, user name and password for EAP Security. This command returns the normal response codes.

The valid outer authentication values are:

Eap-FAST: 43

Eap-TLS: 13

Eap-TTLS: 21

Eap-PEAP: 25

The valid Inner Authentication values are:

Eap-MSCHAP: 26

Eap-GTC: 6

4.7.8 EAP

The command to configure certificate for EAP-TLS is

AT+ WEAP=< Type >,< Format >,< Size >,< Location >

<ESC>W <data of size above>

- Type: CA certificate(0)/ Client certificate(1)/ Private Key(2)
- Format: Binary(0)/Hex(1)
- Size: size of the file to be transferred.
- Location: Flash(0)/Ram(1)

This command enables the adaptor to receive the certificate for EAP-TLS. This command stores the certificate in flash or RAM, depending on the parameter. Upon deployment of this command, the interface returns the standard command response (section 4) or ERROR, 1 (verbose disabled), if the operation failed.

4.7.9 Certificate Deletion

The command to delete an EAP-TLS certificate from memory is

AT+TCERTDEL=<certificate name>

This command deletes the EAP_TLS certificate stored in flash/ram by name. In the case of EAP-TLS certificate names are:

- TLS_CA
- TLS_CLIENT

- TLS_KEY

4.7.10 Disable 802.11 Radio

The command to enable or disable the radio is

```
AT+WRXACTIVE=n
```

If *n* is 0, the radio is disabled and if *n* is 1, the radio is enabled.

If `WRXACTIVE` = 1, the 802.11 radio receiver is always on. This minimizes latency and ensures that packets are received at the cost of increased power consumption. The GainSpan SOC cannot enter Deep Sleep (section 4.9.1) even if it is enabled (`PSDPSLEEP=1`). Power Save mode (section 4.7.1111) can be enabled but will not save power, since the receiver is left on.

If `WRXACTIVE` = 0, the receiver is switched off after association is complete. If Power Save mode is not enabled (`WRXPS` not issued or `WRXPS`=0), the receiver will not be turned on again unless `WRXACTIVE` = 1 is received. Packets will not be received and disassociation could occur. If Power Save mode is enabled (`WRXPS`=1) prior to issuing `WRXACTIVE` = 0, the receiver will be turned off, but will turn on again when it is time to listen for the next beacon from the Access Point. If Deep Sleep is also enabled, the receiver will turn off, and the SOC will enter Deep Sleep when all pending tasks are completed, but again the system will be awakened to listen to the next beacon. If a transition to Standby is requested and occurs (section 4.9.2), the SOC will remain in Standby for the requested period, and will ***not*** awaken to receive a beacon during that time.

4.7.11 Enable/Disable 802.11 Power Save Mode

The command to configure 802.11 Power Save Mode is

```
AT+WRXPS=n
```

If *n* is 0, Power Save is disabled and if *n* is 1, Power Save is enabled.

In 802.11 Power Save Mode, the node (in this case, the IP2WiFi Adapter) will inform the Access Point that it will become inactive, and the Access Point will buffer any packets addressed to that node. In this case, the GainSpan SOC radio receiver is turned off between beacons. The node will awaken to listen to periodic beacons from the Access Point, that contain a ***Traffic Indication Map*** (TIM) that will inform the Station if packets are waiting for it. Buffered packets can be retrieved at that time, using ***PSPoll*** commands sent by the node. In this fashion, power consumed by the radio is reduced (although the benefit obtained depends on traffic load and beacon timing), at the cost of some latency..

The latency encountered depends in part on the timing of beacons, set by the Access Point configuration. Many Access Points default to 100msec between beacons; in most cases this parameter can be adjusted.

4.7.12 Enable/Disable Multicast Reception

The command to configure multicast reception is

```
AT+MCSTSET=n
```

If n = 0, multicast reception is disabled; if n = 1, multicast reception is enabled.

4.7.13 Transmit power

The command to set the transmit power is

```
AT+WP=<power>
```

On deployment of this command, the transmit power is set to the supplied value. The desired power level shall be specified in ASCII decimal format. The value of the parameter can range from 0 to 7 for internal PA GS101x,with a default value of 0(for maximum RF output) and from 2 to 15 for external PA GS101x with default value of 2(for maximum RF output).

This command returns the standard command response (section 4) to the serial interface.

4.7.14 Sync Loss Interval

The command to configure the sync loss interval is

AT+WSYNCINTRL=<n>

On execution of this command the adaptor set the sync loss interval for n times the beacon interval so that if the adaptor does not receive the beacon for this time it informs the user this event as “Dissociation event”. The default value of sync loss interval is 30.This command accept the sync loss interval from 1 to 65535.

This command returns the standard command response (section 4) to the serial interface.

4.7.15 External PA

The command to enable the external PA is

AT+EXTPA=<n>

n=1 to enable the external PA

n=0 to disable external PAThis command forces the adaptor to go to standby and comes back immediately and causing all configured parameters and network connection will be lost.

This command returns the standard command response (section 4) to the serial interface.

4.7.16 . Association Keep Alive Timer

The command to configure the keep-alive timer interval is

AT+PSPOLLINTRL=<n>

On execution of this command, the adaptor will set the keep-alive time interval for n seconds. This keep-alive timer will fire for every n seconds once the adaptors associated. This timer will keep the adaptor in associated state even there is no activity between AP and adaptor. The default vale is 45 seconds. This command accepts keep-alive timer interval from 0 to 65535 seconds. The value 0 disables this timer.

This command returns the standard command response (section 4) to the serial interface.

4.7.17 Unsolicited Data Transmission

The adaptor supports unsolicited data transmission (data transmission without association). The Command to enable this is:

AT+UNSOLICITEDTX=<Frame Control>,<Sequence Control>,<Channel>,<Rate>,<WmmInfo>,

<Receiver Mac>,<Bssid of AP>,<Frame Length>

This command enables the unsolicited data transmission with the parameters configured. After issuing this command, the user needs to send the payload data as following:

<ESC>D/d <PayLoad of the above Frame length>

- Frame Control: is the 802.11 frame control field. It should be limited to all data frames and management frames like beacons, association requests and probe responses.
- Sequence Control: is the seq number of the frame. This field consists of 12 bits(LSB) fragment number and 4 bit (MSB)sequence number. (0-65535).
- Channel: is the channel on which the data to be sent.
- Rate: is the rate at which the data to be send and the possible values are:

RATE_1MBPS	=	130,
RATE_2MBPS	=	132,
RATE_5_5MBPS	=	139,
RATE_11MBPS	=	150

- WmmInfo: is the wmm information to be sent.
- Receiver Mac: is the remote MAC address of the frame to be sent.
- Bssid: is bssid of the AP.
- Frame Length: is the length of the payload. The maximum size of the frame is limited to 1400 bytes.

This command returns standard command response (section 4).

4.8 BATTERY CHECK

4.8.1 Battery Check Start

The command to initiate battery checking is:

```
AT+BCHKSTRT=<Batt.chk.freq>
```

The valid range for the parameter Batt.chk.freq is between 1 and 100. Upon deployment of this command, the adapter performs a check of the battery voltage each Batt.chk.freq number of sent packets and stores the resulting value in nonvolatile memory (only the most recent value is stored). Note that battery checks are performed during packet transmission to ensure that they reflect loaded conditions. Battery checks can be used to ensure that a battery-powered system is provided with sufficient voltage for normal operation. Low supply voltages can result in data corruption when profile data is written to flash memory.

This command returns standard command response (section 4) or ERROR, if the operation fails.

4.8.2 Battery Warning/Standby Level Set

The command to set the battery warning/standby level to enable the adaptor's internal battery measuring logic:

AT+ BATTLVLSET=<Warning Level>,<Warning Freq>,<Standby Level>

Upon execution of this command the adaptor's internal battery level monitoring logic starts. This command should be executed before the battery check start command.

Warning Level: is the battery voltage, in millivolts, When the adapter battery voltage is less than this level sends the message "Battery Low" to the serial interface.

Warning Freq: is the frequency at which the adaptor sends the "Battery Low" message to the serial interface once the adaptor's battery check detected low battery.

Standby Level: is the battery voltage, in millivolts, When the adapter battery voltage reaches this level the adaptor sends the message "Battery Dead" to the serial interface and goes to long standby.

This command returns standard command response (section 4).

4.8.3 Battery Check Set

The command to set/reset the battery check period after battery check has been started is:

```
AT+BCHK=<Batt.chk.freq>
```

The valid range for the parameter Batt.chk.freq is between 1 and 100. Upon receipt, the adapter records the new value of the battery check frequency so that adapter performs the battery voltage check with the new value set. This command returns standard command response (section 4).

The same command can be used to get the current configured battery check period, the usage as follows

```
AT+BCHK=?
```

This command returns the battery check frequency along with standard command response (section 4).

4.8.4 Battery Check stop

The command to stop checking the battery state is:

```
AT+BCHKSTOP
```

Upon deployment of this command, battery check is halted.

4.8.5 Battery Value Get

The command to retrieve the results of battery check operations is:

```
AT+BATTVALGET
```

This command should return a message with the latest value, e.g. `Battery Value: 3.4 V`, followed by the usual status message.

If this command is issued before issuing the command to start battery checks, it returns `ERROR` or `1`, depending on the current verbose setting.

4.9 Power State Management

4.9.1 Enable/Disable SOC Deep Sleep

The command to enable the GainSpan SOC's power-saving Deep Sleep processor mode is

```
AT+PSDPSLEEP
```

When enabled, the SOC will enter the power-saving Deep Sleep mode when no actions are pending. In Deep Sleep mode, the processor clock is turned off and SOC power consumption is reduced to less than 1 mW (about 0.1 mA at 1.8 V). Note that other components external to the SOC may continue to dissipate power during this time, unless measures are taken to ensure that they are also off or disabled.

The processor can be awakened by sending data on the serial port. However, several milliseconds are required to stabilize the clock oscillator when the system awakens from Deep Sleep. Since the clock oscillator must stabilize before data can be read, the initial data will not be received; "dummy" (discardable) characters or commands should be sent until an indication is received from the application.

This command does not return any response code to the serial interface. The s2w adapter sends the message "Out of Deep Sleep" once it comes out from deep sleep.

4.9.2 Request Standby Mode

The command to request a transition to ultra-low-power Standby operation is

```
AT+PSSTBY=x[,<DELAY TIME>,<ALARM1 POL>,<ALARM2 POL>]
```

The parameters are:

- `x` is the Standby time in milliseconds. If a delay time (see below) is provided, the Standby count begins after the delay time has expired.
- `DELAY TIME` is the delay in milliseconds from the time the command is issued to the time when the SOC goes to Standby.
- `ALARM1 POL` is the polarity of the transition at pin 31 of the SOC which will trigger an alarm input and waken the GainSpan SOC from Standby. A value of 0 specifies a high-to-low transition as active; a value of 1 specifies low-to-high.

- `ALARM2 POL` is the polarity of the transition at pin 36 that triggers an alarm input, using the same convention used for Alarm1.

The parameters `DELAY TIME`, `ALARM1 POL`, and `ALARM2 POL` are optional. Specifying an alarm polarity also enables the corresponding alarm input.

This command does not return any response code to the serial interface .When this command is issued, the GainSpan SOC will enter the ultra-low-power Standby state (after the optional delay time if present), remaining there until *x* milliseconds have passed since the command was issued, or an enabled alarm input is received. Any current CID's are lost on transition to Standby. On wakeup, the adapter sends the message `Out of Standby-<reason of wakeup>` or the corresponding error code, depending on verbose status.

In Standby, only the low-power clock and some associated circuits are active. Serial messages sent to the UART port will not be received. The radio is off and packets cannot be sent or received. Therefore, before requesting a transition to Standby, the requesting application should ensure that no actions are needed from the interface until the requested time has passed, or provide an alarm input to awaken the SOC when needed. The alarm should trigger about 10 msec prior to issuance of any serial commands.

The Standby clock employs a 34-bit counter operating at 131,072 Hz, so the maximum possible Standby time is 131,072,000 milliseconds, or about 36.4 hours. Standby is not entered until all pending tasks are completed, and a few milliseconds are required to store any changes and enter the Standby state; a similar delay is encountered in awaking from Standby at the end of the requested time. Therefore, we do not recommend Standby times less than about 32 milliseconds.

4.10 PROVISIONING

4.10.1 Web Provisioning

The adaptor supports provisioning through web pages. The command to start web provisioning is

AT+WEBPROV=<user name>,<passwd>,<adaptor Ip address>,<adaptor subnet mask>,
<adaptor gateway Ip address>

Prior to issuing this command the adaptor should be in an *ad hoc* or limited AP network. Upon reception of this command the adaptor starts a web server. It returns the normal response code OK or ERROR depends on the success or failure condition.

Once the adaptor returns the success response ("OK"), the user can open a webpage on the PC the *ad hoc* or Limited AP network created with the IP address of the adaptor with a http client application (e.g. IE). User can configure both L2 and L3 level information on the provisioning web pages. Submit button stores all the configured information in the adaptor and logout/boot button presents all provisioned information to the serial host and resets the adaptor.

The size of the username and password is limited to 16 characters.

The provisioned information sends to serial host is:

- SSID=<ssid>
- CHNL=<channel>

- CONN_TYPE=<connType> /* either BSS or IBSS */
- MODE=<mode> /* 0 –> 802.11b */
- WEP_ID=<wep ID>
- WEP_KEY=<wep key>
- PSK_PASS_PHRASE=<psk PassPhrase>
- EAP_USER_NAME=<eap User name>
- EAP_PASS_WORD=<eap PassWord>
- PRIVATE_KEY_LEN=<private Key Length>
- PRIVATE_KEY=<private key file> /* private key file is stream of bytes of length= private Key Length
- CLIENT_CERT_LEN=<client Certificate Length>
- CLIENT_CERT =<client certificate> /* client certificate is stream of bytes of length= client Certificate
- CA_CERT_LEN=<CA certificate Length>
- CA_CERT =<CA certificate> /* CA certificate is stream of bytes of length= CA Certificate
- NEW_USER_NAME<new User Name>
- NEW_PASS=<new Password>

This command returns standard command response (section 4) or ERROR, if the web server starts failed.

4.10.2 Web Provisioning (Logo)

The adaptor supports adding the Logo that will appear on the web pages used for provisioning. The command to add the logo is

AT+WEBLOGOADD=<size>

<Esc>L<Actual File content>

<size> is measured in bytes and the maximum size is 1788 bytes. This command is typically done at the manufacturing line in the factory. This command can be done only once. There is no command to delete the Logo. This command returns standard command response (section 4) to the serial interface.

4.11 RF Tests

The adaptor supports different types of frame transmission for RF capability measurement. It supports asynchronous data transmission/reception and modulated/un-modulated wave transmission.

4.11.1 Asynchronous Frame Transmission

The command to enable the asynchronous frame transmission is:

AT+RFFRAMETXSTART=<Channel>,<Power>,<Rate>,<No.Of.Times>,<Fr.Intrvel>,<FrameControl>,<DurationId>,<Sequence Control>,<frameLen>,<Preamble>,<Scrambler>[,<DstMac>,<Src Mac>]

After issuing this command the user needs to send the payload data as following,

<ESC>A/a <PayLoad of the above Frame length>

- Channel: the channel on which the data is to be sent.
- Power: the power in db at which the frame to be sent (0-7). The value of this parameter can range from 0 to 7 for internal PA and from 2 to 15 for external PA.
- Rate: the rate at which the data can be sent and the possible values are:

  RATE_1MBPS = 2,

  RATE_2MBPS = 4,

  RATE_5.5MBPS = 11,

  RATE_11MBPS = 22
- No. of Times: the number of asynchronous frames to be sent (1-65535).
- Fr.Intrvel: the interval between each frame, in microseconds (1-65535).
- Frame Control: expects only the lower byte (B0...B7) of 802.11 frame control field, which includes protocol version, Type and Subtype. All the higher order bits (B8...B15) are made zero for this command.

E.g. Frame control field of beacon frame is: 128

Higher Byte B15 – B8	Sub Type B7-B4	Type B3- B2	Protocol Version B1 – B0
00000000	1000	00	00

- *Note: This command is intended to transfer* ***only*** *data & a few Management frames like Beacon/Probe request/Probe response/Association request*
- DurationId: duration id information to be sent (0-65535).
- Sequence Control: the seq number of the frame (0-65535). This field consists of 12 bits(LSB) fragment number and 4 bit (MSB)sequence number. (0-65535).
- frameLen: the length of the payload. The maximum size of the frame is limited to 1400 bytes.
- Preamble: the short (1) or long (0) preamble.
- Scrambler: the ON(0) or OFF(1) scrambler field of the frame
- DstMac: the MAC address through which the frame to be send.
- Src Mac: MAC address for the WiFi Bridge.

Example:

AT+RFFRAMETXSTART=1,3,4,2,200,0,11,0,30,0,1,00:1d:c9:00:07:a2

<ESC>A123456789012345678901234567890

Please check the wireless sniffer to see the frame on air. The AT+RFSTOP (section 4.11.4) command should be issued prior to successive frame transmission command.

CSMA/CA is not executed before transmitting this command; hence it could destroy the network.

This command returns standard command response (section 4) to the serial interface.

4.11.2 Asynchronous Frame Reception

The command to enable the asynchronous frame reception is:

AT+RFRXSTART=<Channel>[,<Sendtouser>]

- Channel: the channel on which the data is to be received.
- Sendtouser: is a flag (0/1) which instructs the adaptor to send the received data to the serial interface.

The Frame Transmission/Reception Stop command(section 4.11.4) will send the status information of the received frames to the serial interface.

Example: **AT+RFRXSTART=1,1** → this will send the received data to the serial interface

AT+RFRXSTART=1,0 → this will not send the received data to the serial interface

In both case the received frame information is stored in SRAM and once issue the command AT+RFSTOP sends the received frame information to the user through serial. We recommend using the second option. This command returns standard command response (section 4) to the serial interface.

4.11.3 Modulated/Un-Modulated Wave Transmission

The command to enable the modulated/un-modulated wave transmission is:

AT+RFWAVETXSTART=<Modulated>,<Channel>,<Rate>,<PreambleLong>,<ScamblerOff>,<Cont.Tx>,<Power>,<Ssid>

- Modulated : the flag to tell whether the wave transmission should be modulated (1) or un-modulated (0)
- Channel: the channel on which the data to be received.
- Rate: the rate at which the wave transmission should happen.

  TX_RATE 1mbps = 0,

  TX_RATE 2 mbps = 1,

  TX_RATE 5.5 mbps = 2,

  TX_RATE 11 mbps = 3,
- PreambleLong: long preamble (1) or short preamble (0).
- ScamblerOff: the scrambler field OFF (1) or ON (0).
- Cont.Tx: the wave transmission is continuous (1) or not (0).
- Power: the power in db at which the wave transmission should happen (0-7).
- Ssid: the ssid of the network created for the wave transmission.

Example:

```
AT+RFWAVETXSTART=1,4,2,1,1,1,3,aaa  -→(modulated)
AT+RFWAVETXSTART=0,4,3,0,1,1,3,bbb   --→(un-modulated)
```

4.11.4 Frame Transmission/Reception Stop

The command to stop any of the RF test transmissions/receptions is:

AT+RFSTOP

Upon execution of this command the adaptor stops any of the frame transmission/reception RF tests. This command sends the status information of the received asynchronous frames to the serial interface other than the normal command response if this command issued for the asynchronous frame reception stop.

Example:

AT+RFSTOP (if this command issued after AT+ RFRXSTART, then it sends the following information to the serial interface)

```
Total frames received =xxxx
Correct frames received =xxxx
Incorrect frames received =xxx
FCS Error frames received =xxx
```

4.12 Miscellaneous

4.12.1 Enhanced Asynchronous Notification

IP2WiFi Adapter supports an enhanced asynchronous notification method. The command to enable/disable this feature is

AT+ASYNCMSGFMT=n

n is

- 0 – Disable this feature
- 1 – Enable this feature

This command returns standard command response (section 4) to the serial interface.

Enabling this feature results with all asynchronous messages going to the serial interface with a header. Also during these asynchronous message transfer IP2WiFi adapter make the gpio 19 high. The asynchronous message format is as shown below:

<ESC><TYPE><SUBTYPE><LENGTH><MESSAGE>

TYPE – Type of message and the length is one byte. For asynchronous message, it is 0x41 (Ascii value A)

SUBTYPE – Message subtype and the length of this field is one byte. Normally this field contains the ascii value of the subtype message. Refer section 3.6.4 for subtype values.

LENGTH – Length of the asynchronous message in hex. This field length is 2 bytes.

MESSAGE – Exact asynchronous message as string. Refer section 3.7.4 for all enhanced asynchronous messages.

4.12.2 Node Start Up Handling

For proper synchronization between host micro controller (MCU) and IP2WiFi adapter, the following steps must be followed:

- In case of UART interface, during boot up host MCU shall send dummy ‘AT’ command and wait for response from the IP2WiFi adapter. The host MCU must continuously send these dummy ‘AT’ commands till ‘OK’ response is received from IP2WiFi adapter.
- In case of SPI interface, during boot up host MCU must check the status of host wake-up signal (GPIO#28 of IP2WiFi adapter). Once host wake-up signal is HIGH, then host MCU can send the ‘AT’ commands.

If for some reason host MCU getting reset, then IP2WiFi adapter must be explicitly reset using EXT_RESET pin and the MCU should wait for the wake-up signal(GPIO#28) become high in case of SPI interface. However if reset provision is not available, then host MCU must continuously send dummy ‘AT’ commands till ‘OK’ response is received from IP2WiFi adapter.

4.12.3 SPI interface handling

In the case of SPI interface, the GS101X node acts as slave and will communicate to master SPI controller. By default, SPI interface supports Motorola protocol with clock polarity 0 and clock phase 0. For more detailed specification of SPI frame format and timing characteristics refer GS1011 data sheet.

Since SPI data transfer works in full duplex mode, its required to make use of special octet to indicate idle data. Similarly if host MCU is sending data at higher rate flow control mechanism is required. In order differentiate these special control codes (such as idle pattern , flow control codes and other control octets) from user data, byte stuffing mechanism is incorporated.

SPI transmit data handling procedure:
The SPI data transfer layer makes use of an octet (or byte) stuffing procedure. The Control Escape octet is defined as binary 11111011 (hexadecimal **0xFB**), most significant bit first. Each special control pattern is replaced by a two octet sequences consisting of the Control Escape octet followed by the original octet exclusive-or'd (XOR) with hexadecimal **0x20**. Receiving implementations must correctly process all Control Escape sequences.

Escaped data is transmitted on the link as follows:

Pattern	Encoded as	Description
0xFD	0xFB 0xDD	***Flow control XON***
0xFA	0xFB 0xDA	***Flow control XOFF***
0x00	0xFB 0x20	Inactive link detection
0xFB	0xFB 0xDB	Control ESCAPE
0xF5	0xFB 0xD5	***IDLE character***
0xFF	0xFB 0xDF	Inactive link detection
0xF3	0xFB 0xD3	SPI link ready indication

One dedicated GPIO signal (***GS_SPI _HOST_WAKEUP***: ***GPIO#28)*** is available for data ready indications from Slave GS1011 node to Master Host controller. This ***GS_SPI _HOST_WAKEUP*** signal is asserted high during valid data transmission period, so that the host (master SPI) starts pulling out data by giving SPI clock and ***GS_SPI _HOST_WAKEUP*** signal is de-asserted once transmission is completed. Master host controller must provide clock as long as ***GS_SPI_HOST_WAKEUP*** signal is active.

Special character (***GS_SPI _IDLE***) will be transmitted during idle period (if there is no more data to transmit) and must be dropped at the receiving Host.

SPI receive data handling procedure:
Since byte stuffing is used, each Control Escape octet must be removed and the next immediate octet is exclusive-or'd (XOR) with hexadecimal **0x20**. If received buffer has reached the upper water mark, then ***XOFF*** character will be sent out informing the host to stop transmitting actual data. After receiving ***XOFF*** character host must stop transmitting actual data and can send IDLE bytes, until the XON is received. Once the host receives ***XON***, then it may resume the valid data transmissions.

Special control byte ***IDLE*** will be dropped at receiver.

4.12.4 Pin connection for SPI Interface

Host MCU	IP2WiFi Adapter	Remarks
MSPI_DOUT	SSPI_DIN	
MSPI_DIN	SSPI_DOUT	
MSPI_SS	SSPI_SS	
MSPI_CLK	SSPI_CLK	
GPIO	GPIO#28	Host wake-up signal
Ground	Ground	

4.12.5 Factory Defaults

The IP2WiFi adaptor stores factory defaults to its flash. Currently supporting only the MAC address as factory default one. If the factory default MAC address location contains a valid MAC address then the IP2Wifi adaptor uses this as its MAC address otherwise it use the default factory MAC ID as its MAC address.

T
fa

Checksum(1 byte)	Length (1 byte)	Mac address (6 byte)

Checksum : the simple byte wise xor of both length and MAC address

Length : the length in bytes of MAC address and length (here it is 7)

Mac address : the MAC address

The user can override the factory default mac address by using the AT commands mentioned in section 4.7.1.

4.12.6 Firmware Upgrade

The command to upgrade the firmware is

```
AT+FWUP= <SrvIp>, <SrvPort>, <SrcPort>, <retry>,<adapator Ip address>,
         <adaptor subnet mask>,<adaptor gateway Ip address>
```

This command starts the firmware upgrade procedure over the wireless link.

- SrvIp is the IP address of the firmware upgrade server;
- SrvPort is the server port number to be used for firmware upgrade;

- SrcPort is the adapter port number to be used for firmware upgrade.
- Retry is the number of times the node will repeat the firmware upgrade attempt if failures are encountered. The default value is 10 and the retry count ranges from 0 to 0xffffffff.

When a valid command has been received, the adapter returns the message: `Firmware upgrade is going on, Please wait....` followed with the status message OK or 0, applies only to the validity of the command. After attempting to upgrade the firmware, the node sends an additional message describing the result of the actual firmware upgrade attempt.

After a successful firmware upgrade, the Adapter will reset and boot up using the updated firmware

and issue the message `APP Reset-FW-UP-FAILURE`.

If the firmware upgrade attempt failed after successful upgrade of one flash image(flash0), the Adapter will reset and boot up, issue the message APP Reset-FW-UP-RECOVERY, associate back to the nework with previous settings and try to upgrade the firmware again. The retry count decides how many times this can be done.

If the node is not associated, the adapter returns `ERROR` or `1`, based on verbose settings.

4.12.7 Set System Time

The command to set the adaptor system time is

AT+SETTIME=<dd/mm/yyyy>,<HH:MM:SS>

Upon execution of this command the adaptor set its system time to the time specified as the parameters and returns the standard command response.

4.12.8 Get System Time

The command to get the current system is

AT+ GETTIME=?

Upon reception of this command the adaptor sends the current system time in milliseconds since epoch(1970) followed by the standard command response to the serial interface. The time format comes on the serial interface as follows:"Current Time in msec since epoch=xxxxxxx"

4.12.9 GPIO Out HIGH/LOW

The command to set/reset (high/low) a gpio pin is

AT+DGPIO=<GPIO-NO>,<SET/RESET(0/1)>

This command sets the Gpio 'GPIO-NO' pin level to high or low as per the SET/RESET parameter.

Note: Only the Gpio Pins that are not mixed with the any used IOs (like UART/SPI, etc.) can be set high/low with this command.

The supported Gpios and the corresponding numbers are:

- Gpio10 : 10
- Gpio11 : 11

- Gpio30 : 30
- Gpio31 : 31

4.12.10 Error Counts

The command to get the error count statistics is

AT+ERRCOUNT=?

This command returns error count information to the serial host interface followed by the standard command response (section 4).

The error counts includes

- Watchdog reset counts
- Software reset counts
- Wlan abort/assert counts

4.12.11 Version

The command to output the current version information is

```
AT+VER=?
```

The command returns version information followed by the standard command response (section 4) to the serial host:

- IP-to-Wi-Fi version;
- GainSpan Embedded Platform Software version;
- WLAN firmware version.

5 References

GS1011 Ultra-Low-Power Wireless System-On-Chip Datasheet

GS1011-DS-AB
GainSpan Corporation, www.gainspan.com

GS1011M Module Data Sheet

GS1011M-DS

IEEE Standard for Information Technology-Telecommunications and information exchange between systems-Local and metropolitan area networks

Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 802.11-2007
IEEE, www.ieee.org

V.250, Serial asynchronous automatic dialing and control

V.251, Procedure for DTE-controlled call negotiation

International Telecommunications Union, www.itu.int

Communications Networks

A. Leon-Garcia and I. Widjaja, McGraw-Hill 2000, p. 582

GainSpan Corporation, www.gainspan.com