Anybus® Wireless Bolt/Bridge II

AT Commands

REFERENCE GUIDE

SCM-1202-004 1.2 ENGLISH

Connecting Devices™

Important User Information

Liability

Every care has been taken in the preparation of this document. Please inform HMS Industrial Networks AB of any inaccuracies or omissions. The data and illustrations found in this document are not binding. We, HMS Industrial Networks AB, reserve the right to modify our products in line with our policy of continuous product development. The information in this document is subject to change without notice and should not be considered as a commitment by HMS Industrial Networks AB. HMS Industrial Networks AB assumes no responsibility for any errors that may appear in this document.

There are many applications of this product. Those responsible for the use of this device must ensure that all the necessary steps have been taken to verify that the applications meet all performance and safety requirements including any applicable laws, regulations, codes, and standards.

HMS Industrial Networks AB will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features, timing, or functional side effects found outside the documented scope of this product. The effects caused by any direct or indirect use of such aspects of the product are undefined, and may include e.g. compatibility issues and stability issues.

The examples and illustrations in this document are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular implementation, HMS Industrial Networks AB cannot assume responsibility for actual use based on these examples and illustrations.

Intellectual Property Rights

HMS Industrial Networks AB has intellectual property rights relating to technology embodied in the product described in this document. These intellectual property rights may include patents and pending patent applications in the USA and other countries.

Table of Contents

Page

This page intentionally left blank

1 Preface

1.1 About This Document

This document describes the available AT commands for Anybus Wireless Bolt/Bridge II.

The reader of this document is expected to be familiar with the product and have a good knowledge of wireless communication and network technology.

For additional related documentation, file downloads and technical support, please visit the Anybus support website at www.anybus.com/support.

1.2 Document history

Revision list

Version	Date	Description
1.0	2016-06-27	Beta release
1.1	2016-10-01	First public release
1.2	2017-03-31	Updated for Wireless Bridge II

1.3 Trademark Information

2 Introduction

AT commands allow more configuration options than the web interface and can be scripted for batch configuration of multiple units. A string of AT commands can for example be sent from a PLC for automatic configuration during initial setup or when replacing units.

Each command line can only contain a single command and must not exceed 300 characters. Some commands may have additional limitations. This document describes the structure and syntax of each command and also includes examples for most of them.

Some of the commands require that the unit is rebooted before they become effective. This is indicated in the description of the command.

UPPER CASE is only used for clarity in this manual, AT commands are not case sensitive.

2.1 Data Types

The description of each command also specifies the data types used for the parameter values. There are five different data types:

String

Strings can contain all the printable characters from the ISO 8859-1 (8-bit ASCII) character set except `"` (double quote) `,` (comma) and `\` (backslash).

The string does not need surrounding quotes.

Integer

Integer values can be entered in decimal form or as a hexadecimal string beginning with `0x`; e.g. `15` can also be entered as `0x0000000F`.

Boolean

Boolean values can be either `0` (false) or `1` (true).

NetworkAddress

Used for IP addresses. Must be entered as four integer values in the range 0 to 255 separated by periods, e.g. `192.168.0.98`.

MACAddress

Used for Ethernet and Bluetooth MAC addresses. Addresses must be entered as six groups of two hexadecimal digits in one of the following formats:

```
00A0F7101C08
00:A0:F7:10:1C:08
00-A0-F7-10-1C-08
```

2.2 Sample Scripts

The following command scripts can be copied and pasted directly into the **AT Commands** text box in the web interface. The order of the commands in the scripts is important and should not be changed. All scripts start from the factory default configuration.

See also the descriptions of the individual AT commands for more information.

These scripts are only intended as examples. The IP addresses, channels, passwords, etc. in the scripts should be changed as required for your application.

2.2.1 WLAN Access Point with 7 Clients

This example will set up one unit as a WLAN access point and up to 7 units as WLAN clients. The access point unit has the SSID “MyNewAP” and is using channel 3 on the 2.4 GHz band. Security mode is WPA2-PSK with the passkey “Sesame2016”.

Each device will reboot after applying the new configuration. The IP addresses will change to 192.168.0.100 for the access point, and 192.168.0.101–107 for the clients.

WLAN Access Point

```
AT*ANDHCP=0,1
AT*ANIP=192.168.0.100,255.255.255.0,192.168.0.100,1
AT*WMODE=1,1
AT*WASSID=MyNewAP,1
AT*WACH=3,1
AT*WAAM=2,1
AT*WKEY=Sesame2016,1
AT*AMREBOOT
```

WLAN Client 1

```
AT*ANDHCP=0,1
AT*ANIP=192.168.0.101,255.255.255.0,192.168.0.100,1
AT*WMODE=0,1
AT*WSAM=2,1
AT*WKEY=Sesame2016,1
AT*WSCP=,MyNewAP,3
AT*AMREBOOT
```

The IP address of each additional client must be unique on the network. This is achieved by changing the first parameter in the `AT*ANIP` command:

WLAN Client 2

```
AT*ANIP=192.168.0.102,255.255.255.0,192.168.0.100,1
```

WLAN Client 3

```
AT*ANIP=192.168.0.103,255.255.255.0,192.168.0.100,1
```

...and so on.

2.2.2 WLAN Client Roaming Between 2 Access Points

This example will set up two units as WLAN access points, and a third unit as a WLAN client roaming between them.

The access points use different channels in the 2.4 GHz band (channels 1 and 6) to avoid interference. They share the same SSID “MyNetwork”, and use WPA2-PSK security with the same passkey “Sesame2016”.

Each unit will reboot after applying the new configuration. The IP addresses will change to 192.168.0.100/101 for the two access points, and 192.168.0.102 for the client.

WLAN Access Point 1

```
AT*ANDHCP=0,1
AT*ANIP=192.168.0.100,255.255.255.0,192.168.0.100,1
AT*WMODE=1,1
AT*WASSID=MyNetwork,1
AT*WACH=1,1
AT*WAAM=2,1
AT*WKEY=Sesame2016,1
AT*AMREBOOT
```

WLAN Access Point 2

```
AT*ANDHCP=0,1
AT*ANIP=192.168.0.101,255.255.255.0,192.168.0.100,1
AT*WMODE=1,1
AT*WASSID=MyNetwork,1
AT*WACH=6,1
AT*WAAM=2,1
AT*WKEY=Sesame2016,1
AT*AMREBOOT
```

WLAN Client

The example MAC addresses in the `AT*WSCLW` command should be replaced with the actual MAC addresses of the units configured as access points.

```
AT*ANDHCP=0,1
AT*ANIP=192.168.0.102,255.255.255.0,192.168.0.100,1
AT*WMODE=0,1
AT*WSAM=2,1
AT*WKEY=Sesame2016,1
AT*WSCLW=0,020133004E00,MyNetwork,1,1
AT*WSCLW=1,020136004B00,MyNetwork,6,1
AT*WSC
AT*AMREBOOT
```

About WLAN Roaming

The client unit will initially connect to the access point with the best RSSI. If the RSSI falls below a certain value the client will poll the access points in the list that was set up with the `AT*WSCLW` command. If any of them is available and has a better RSSI the current connection will be dropped and a new connection is initiated.

3 Standard Commands

3.1 AT&F Restore to Factory Settings

AT&F

This command instructs the unit to set all parameters to their defaults as specified by the manufacturer.

Syntax:

```
AT&F
```

3.2 AT* List Available Commands

Returns a list of all available AT commands

AT*

Syntax:

```
AT*
```

Example:

```
TODO
```

3.3 AT Attention

AT

Attention command determining the presence of a DCE

Syntax:

```
AT
```

4 Network Commands

4.1 AT*ANDHCP DHCP Mode

Set/get the DHCP mode. If activated, this will take precedence over settings made with AT*ANIP

AT*ANDHCP=

Set the DHCP mode

Syntax:

```
AT*ANDHCP=<dhcp_client>,<dhcp_server>,<store>
```

Input Parameters:

Name	Type	Description
dhcp_client	Integer	0: Off, use static IP address (default value) 1: On, acquire an IP address using DHCP
dhcp_server	Integer	0: Off, Disable DHCP server(default value) 1: ON, Enable DHCP server 2: DHCP Relay, Relay DHCP messages to an external DHCP server.
store	Boolean	If store is 1 the new value is stored permanently.

Requires a reboot for the changes to take effect

AT*ANDHCP?

Read the current DHCP setting

Syntax:

```
AT*ANDHCP?
```

Example:

```
AT*ANDHCP? *ANDHCP:<dhcp_client>,<dhcp_server>
```

4.2 AT*ANIP IP Settings

Set/get IP settings for the device

AT*ANIP=

Short description for AT*ANIP=

Syntax:

```
AT*ANIP=<ip_addr>,<netmask>,<gateway>,<store>
```

Input Parameters:

Name	Type	Description
ip_addr	NetworkAddress	IP address for the device
netmask	NetworkAddress	Netmask for the device
gateway	NetworkAddress	The IP address of the gateway
store	Boolean	If store is 1 the new value is stored permanently.

Requires a reboot for the changes to take effect

AT*ANIP?

Get the IP settings

Syntax:

```
AT*ANIP?
```

Example:

```
AT*ANIP? *ANIP:<ip_addr>,<netmask>,<gateway>
```

4.3 AT*ANHN Hostname

Set/get the hostname used with dynamic DNS

AT*ANHN=

Set hostname

Syntax:

```
AT*ANHN=<hostname>,<store>
```

Input Parameters:

Name	Type	Description
hostname	String	The hostname to set. Maximum of 240 characters.
store	Boolean	If store is 1 the new value is stored permanently.

Requires a reboot for the changes to take effect

AT*ANHN?

Get hostname

Syntax:

```
AT*ANHN?
```

Example:

```
AT*ANHN? *ANHN:<hostname>
```

5 Bluetooth Classic Commands

5.1 AT*BCP Connect Peer

AT*BCP=

Bluetooth Connect to Peer. The connection will not be retried if unsuccessful.

Syntax:

```
AT*BCP=<bd_addr>,<name>,<role>
```

Input Parameters:

Name	Type	Description
bd_addr	MACAddress	If specified the MAC address of the remote Bluetooth device must match this value.
name	String	If name is specified and S register 2017 is 1 the remote name must match this value exactly. If S register 2017 is 0 this is a case sensitive substring of the remote name to connect to, e.g. if specified to DUT it will try to connect to DUT, DUTx, xDUT and xDUTx, but not to dut.
role	Integer	The role of the remote device: 100: PAN User role, PAN Profile 101: Network Access Point role, PAN Profile, 103: PAN, This will first try to connect to PANU, and if it fails, connect to NAP All others:Reserved

Example:

```
Input: AT*BCP=8C8B83EE2ACB,,101 will return the handle of the connection
and OK if the connection succeeds, ERROR otherwise.
```

5.2 AT*BCC Close Connection

AT*BCC=

Bluetooth Close Connection

Syntax:

```
AT*BCC=<handle>
```

Input Parameters:

Name	Type	Description
handle	Integer	The handle of the connection to close. If set to 0 and there is no connection with handle 0 any ongoing connection attempts and retries will be aborted.

Example:

```
Input: AT*BCC=0 gives OK when the connection with handle 0 is closed.
```

5.3 AT*BC Connect

AT*BC

Bluetooth Connect (according to the Connection List).

Syntax:

```
AT*BC
```

Example:

```
Input: AT*BC will return the handle of the connection and OK if the
connection succeeds, ERROR otherwise.
```

5.4 AT*BND Name Discovery

AT*BND=

Bluetooth Name Discovery

Syntax:

```
AT*BND=<bd_addr>
```

Input Parameters:

Name	Type	Description
bd_addr	MACAddress	MAC address of the Bluetooth device to get the name of.

Example:

```
Input: AT*BND=8C8B83EE2ACB gives the name of the device and OK if
successful, ERROR otherwise.
```

5.5 AT*BDD Device Discovery

AT*BDD

Perform a Bluetooth Device Discovery i.e. an Inquiry followed by a named lookup for any device that does not report a name in the inquiry response.

Syntax:

```
AT*BDD
```

Example:

```
Input: AT*BDD returns *BDD:<bd_addr>,<cod>,<device_name_
valid>,<bluetooth_name>,<rssi> for each found device followed by OK or
ERROR.
```

5.6 AT*BI Inquiry

AT*BI

Perform a Bluetooth inquiry.

Syntax:

```
AT*BI
```

Example:

```
Input: AT*BI returns *BI:<bd_addr>,<cod>,<device_name_
valid>,<bluetooth_name>,<rssi> for each found device followed by OK or
ERROR.
```

5.7 AT*BSP Server Profile

AT*BSP=

Sets the Bluetooth server profile. A reboot is needed for the setting to take effect. Please note that following values will be affected depending on what role is selected: NAP: AT*BMSP Master Slave policy will be set to 0, ATS2010 max number of connections will be set to 7, AT*BCM Connectability mode will be set to 2 PANU: AT*BMSP Master Slave policy will be set to 1, ATS2010 max number of connections will be set to 1, AT*BCM Connectability mode will be set to 1 IMPORTANT: As the device is connectable after NAP has been set an appropriate Security Mode should be configured.

Syntax:

```
AT*BSP=<role>
```

Input Parameters:

Name	Type	Description
role	Integer	The role of the device: 100: PAN User role, PAN Profile 101: Network Access Point role, PAN Profile.

Requires a reboot for the changes to take effect

Example:

```
Input: AT*BSP=101 sets the device to the Network Access Point role.
```

AT*BSP?

Gets the Bluetooth server profile.

Syntax:

```
AT*BSP?
```

Example:

```
Input: AT*BSP? returns the server profile. See AT*BSP= for values.
```

5.8 AT*BFP Fixed PIN

AT*BFP=

Set the fixed pin/passkey used for BT authentication

Syntax:

```
AT*BFP=<pin>,<store>
```

Input Parameters:

Name	Type	Description
pin	String	The pin/passkey to set. A numerical value 0..999999.
store	Boolean	If store is 1 the new value is stored permanently.

AT*BFP?

Get the fixed pin/passkey used for BT authentication.

Syntax:

```
AT*BFP?
```

5.9 AT*BPM Pairing Mode

AT*BPM=

Set the pairing mode for BT

Syntax:

```
AT*BPM=<mode>,<store>
```

Input Parameters:

Name	Type	Description
mode	Integer	The mode to set. Pairing off = 1, Pairing on = 2.
store	Boolean	If store is 1 the new value is stored permanently.

AT*BPM?

Get the pairing mode for BT. Pairing off = 1, Pairing on = 2.

Syntax:

```
AT*BPM?
```

Example:

```
Input: AT*BPM?
```

5.10 AT*BSM Security Mode

AT*BSM=

Set the security mode to use for BT.

Syntax:

```
AT*BSM=<pin>,<store>
```

Input Parameters:

Name	Type	Description
pin	Integer	The security mode to set. 1 = Security disabled 2 = Fixed pin (Only works if remote device is of the same type. Use Just works for general BT 2.1 devices.) 3 = Just works (default)
store	Boolean	If store is 1 the new value is stored permanently.

AT*BSM?

Get the security mode used for BT. See AT*BSM= for values.

Syntax:

```
AT*BSM?
```

Example:

```
Input: AT*BSM?
```

5.11 AT*BBD Bonded Devices

AT*BBD?

Get the bonded devices.

Syntax:

```
AT*BBD?
```

Example:

```
Input: AT*BBD?
```

5.12 AT*BUB Unbond

AT*BUB=

Un-bonds a previously bonded device.

Syntax:

```
AT*BUB=<bd_addr>
```

Input Parameters:

Name	Type	Description
bd_addr	MACAddress	MAC address of the Bluetooth device to un-bond. If address FFFFFFFFFFFF is selected, all bonded devices will be removed.

Example:

```
Input: AT*BUB=8C8B83EE2ACB
```

5.13 AT*BLEM Low Emission Mode

AT*BLEM=

Set current Low Emission Mode.

Syntax:

```
AT*BLEM=<mode>,<store>
```

Input Parameters:

Name	Type	Description
mode	Integer	The Low Emission mode to set: 0: (Default) Connection period: 10 000 ms Paging timeout: 2000 ms Inquiry timeout: 5000ms 1: Connection period: 5000ms Paging timeout: 300 ms Inquiry timeout: 600 ms 2: Connection period: 3000ms Paging timeout: 200 ms Inquiry timeout: 300 ms 3: Connection period: 3000ms Paging timeout: 80 ms Inquiry time-out: 80 ms 4 - 63: Reserved 64: User specified times, see the ATS General Settings S Register Manipulation command
store	Boolean	If store is 1 the new value is stored permanently.

AT*BLEM?

Get the current Low Emission Mode. See AT*BLEM= for values.

Syntax:

```
AT*BLEM?
```

Example:

```
Input: AT*BLEM?
```

5.14 AT*BDM GAP Discoverability Mode

AT*BDM=

Set current GAP discoverability mode.

Syntax:

```
AT*BDM=<mode>,<store>
```

Input Parameters:

Name	Type	Description
mode	Integer	The GAP discoverability mode to set: 1: GAP non-discoverable mode (default value for PANU mode) 2: GAP limited discoverable mode 3: GAP general discoverable mode (default value for NAP mode)
store	Boolean	If store is 1 the new value is stored permanently.

AT*BDM?

Get the current GAP discoverability mode. See AT*BDM= for values.

Syntax:

```
AT*BDM?
```

Example:

```
AT*BDM? *BDM:<mode>
```

5.15 AT*BCM GAP Connectability Mode

AT*BCM=

Set current GAP connectability mode.

Syntax:

```
AT*BCM=<mode>,<store>
```

Input Parameters:

Name	Type	Description
mode	Integer	The GAP connectability mode to set: 1: GAP non-connectable mode (default value) 2: GAP connectable mode
store	Boolean	If store is 1 the new value is stored permanently.

AT*BCM?

Get the current GAP connectability mode. See AT*BCM= for values.

Syntax:

```
AT*BCM?
```

Example:

```
AT*BCM? *BCM:<mode>
```

5.16 AT*BCA Connection Accept

Used when external connection control is enabled (see ATS2012)

AT*BCA=

Accept or reject a connection attempt. This must be sent to answer the *BCI Connect Indication.

Syntax:

```
AT*BCA=<handle>,<accept>
```

Input Parameters:

Name	Type	Description
handle	Integer	The handle of the connection, received in the *BCI Connect Indication.
accept	Boolean	Set to 1 to accept the connection, 0 to reject it.

5.17 AT*BLN Local Name

AT*BLN=

Set the unit's Bluetooth name. A reboot is needed for the setting to take effect.

Syntax:

```
AT*BLN=<name>
```

Input Parameters:

Name	Type	Description
name	String	The Bluetooth name to use. The maximum length is 31 characters.

AT*BLN?

Get the unit's Bluetooth name.

Syntax:

```
AT*BLN?
```

Example:

```
AT*BLN? *BLN:<name>
```

5.18 AT*BRSS Read RSSI

AT*BRSS=

Get the RSSI for a connection.

Syntax:

```
AT*BRSS=<handle>
```

Input Parameters:

Name	Type	Description
handle	Integer	The handle of the connection to get the RSSI for.

Example:

```
AT*BRSS=<handle> *BRSS:<rssi>
```

5.19 AT*BLQ Read Link Quality

AT*BLQ=

Get the link quality for a connection.

Syntax:

```
AT*BLQ=<handle>
```

Input Parameters:

Name	Type	Description
handle	Integer	The handle of the connection to get the link quality for.

Example:

```
AT*BLQ=<handle> *BLQ:<link_quality>
```

5.20 AT*BLP Limited Pairing

AT*BLP=

Enables or disables limited pairing, only valid for current power cycle. If the device should be pairable after power cycle, see S register 2007.

Syntax:

```
AT*BLP=<enable>,<time_limit>
```

Input Parameters:

Name	Type	Description
enable	Boolean	0: Disable pairing 1: Enable. Pairing will be limited.
time_limit	Integer	The time (in seconds) the unit will be pairable. Valid time is 0 to 300 seconds. Values less than 0 will be treated as 0.

5.21 AT*BCHM Channel Map

AT*BCHM=

Write the Bluetooth channel map. Note that at least 20 channels must be enabled.

Syntax:

```
AT*BCHM=<ch0to15>,<ch16to31>,<ch32to47>,<ch48to63>,<ch64to78>,<store>
```

Input Parameters:

Name	Type	Description
ch0to15	Integer	Bit mask used to enable or disable channels 0 to 15 (Bit 0 = Channel 0). Default value is 0xFFFF.
ch16to31	Integer	Bit mask used to enable or disable channels 16 to 31. Default value is 0xFFFF. (Bit 0 = Channel 16)
ch32to47	Integer	Bit mask used to enable or disable channels 32 to 47 (Bit 0 - Channel 32). Default value is 0xFFFF.
ch48to63	Integer	Bit mask used to enable or disable channels 48 to 63 (Bit 0 = Channel 48). Default value is 0xFFFF.
ch64to78	Integer	Bit mask used to enable or disable channels 64 to 78 (Bit 0 = Channel 64). Default value is 0x7FFF.
store	Boolean	If store is 1 the new value is stored permanently.

AT*BCHM?

Read the Bluetooth channel map.

Syntax:

```
AT*BCHM?
```

Example:

```
AT*BCHM? *BCHM:<ch0to15>,<ch16to31>,<ch32to47>,<ch48to63>,<ch64to78>
```

5.22 AT*BPP Packet policy

AT*BPP=

Set the Bluetooth packet policy. This policy is used for subsequent connections. Any ongoing connections are not affected.

Syntax:

```
AT*BPP=<policy>,<store>
```

Input Parameters:

Name	Type	Description
policy	Integer	0: Long Range (only DM1 packets). 1: Short Latency, basic rates (all DM packets). 2: High Throughput, basic rates (DM + DH packets). 3: As 2 but with 2-EDR enabled. Default. 4:As 3 but with 3-EDR enabled
store	Boolean	If store is 1 the new value is stored permanently.

AT*BPP?

Get the Bluetooth packet policy.

Syntax:

```
AT*BPP?
```

Example:

```
AT*BPP? *BPP:<policy>
```

5.23 AT*BMSP Master Slave policy

AT*BMSP=

Set the Bluetooth Master Slave Role Policy.

Syntax:

```
AT*BMSP=<policy>,<store>
```

Input Parameters:

Name	Type	Description
policy	Integer	0: Always attempt to become master on incoming connections. Should be used for a unit configured as NAP. 1: Always let the connecting device select master/slave role on incoming connections (default).
store	Boolean	If store is 1 the new value is stored permanently.

AT*BMSP?

Get the Bluetooth Master Slave Role Policy.

Syntax:

```
AT*BMSP?
```

Example:

```
AT*BMSP? *BMSP:<policy>
```

5.24 AT*BLCOD Local class of device.

AT*BLCOD=

Set the Bluetooth Local Class Of Device code.

Syntax:

```
AT*BLCOD=<cod>,<store>
```

Input Parameters:

Name	Type	Description
cod	Integer	Valid values for this parameter are specified in the Bluetooth Assigned Numbers Document, www.bluetooth.com. The parameter has been divided into three segments, a service class segment, a major device class segment and a minor device class segment (bits 2-7). The default value is 131072 (Bit 17, Networking).
store	Boolean	If store is 1 the new value is stored permanently.

AT*BLCOD?

Get the Bluetooth Local Class Of Device code.

Syntax:

```
AT*BLCOD?
```

Example:

```
AT*BLCOD? *BLCOD:<cod>
```

5.25 AT*BRCD Read Connected Devices.

AT*BRCD?

Retrieves the MAC address and handle of every connected Bluetooth device.

Syntax:

```
AT*BRCD?
```

Example:

```
AT*BRCD? returns *BRCD:<bd_addr>,<handle> for each connected device
followed by OK or ERROR.
```

5.26 AT*BCLC Clear the Connection list

AT*BCLC=

Clears all the entries in the Connection list.

Syntax:

```
AT*BCLC=<store>
```

Input Parameters:

Name	Type	Description
store	Boolean	If store is 1 the new value is stored permanently.

Example:

```
AT*BCLC=1
```

5.27 AT*BCLR Read the Connection list

AT*BCLR=

Reads an entry in the Connection list.

Syntax:

```
AT*BCLR=<index>
```

Input Parameters:

Name	Type	Description
index	Integer	The index of the entry to read.

Example:

```
AT*BCLR=2 *BCLR:<2>,<bd_addr>,<name>,<role>
```

AT*BCLR?

Reads the list of Connections that the unit can use.

Syntax:

```
AT*BCLR?
```

Example:

```
AT*BCLR? returns *BCLR:<index>,<bd_addr>,<name>,<role> for each entry in
the list followed by OK.
```

5.28 AT*BCLW Write an entry in the Connection list

AT*BCLW=

Writes an entry in the Connection list. NOTE: If store is set to 1 all entries in the connection list will be stored.

Syntax:

```
AT*BCLW=<index>,<bd_addr>,<name>,<role>,<store>
```

Input Parameters:

Name	Type	Description
index	Integer	The index of the entry to write.
bd_addr	MACAddress	If specified the MAC address of the remote Bluetooth device must match this value.
name	String	If name is specified and S register 2017 is 1 the remote name must match this value exactly. If S register 2017 is 0 this is a case sensitive substring of the remote name to connect to, e.g. if specified to DUT it will try to connect to DUT, DUTx, xDUT and xDUTx, but not to dut.
role	Integer	The role of the remote device: 100: PAN User role, PAN Profile 101: Network Access Point role, PAN Profile 103: PAN, This will first try to connect to PANU, and if it fails, connect to NAP, All others:Reserved
store	Boolean	If store is 1 the new value is stored permanently.

Example:

```
AT*BCLW=0,00026F668FA8,dut1,101,1 OK
```

6 WLAN General Commands

6.1 AT*WMODE WLAN Mode

AT*WMODE=

Set WLAN mode, Station or AP.

Syntax:

```
AT*WMODE=<mode>
```

Input Parameters:

Name	Type	Description
mode	Integer	The mode to set, Station (0) or AP (1).

Requires a reboot for the changes to take effect

Example:

```
Input: AT*WMODE=1 sets WLAN mode to AP.
```

AT*WMODE?

Get WLAN mode, Station (0) or AP (1)

Syntax:

```
AT*WMODE?
```

Example:

```
AT*WMODE? *WMODE:<mode>
```

6.2 AT*WKEY Encryption/Authentication Key

AT*WKEY=

Write encryption/authentication key at index 1. This command is a shortcut for AT*WKEYI=1,

Syntax:

```
AT*WKEY=<key>,<store>
```

Input Parameters:

Name	Type	Description
key	String	The key to use. Max 63 characters.
store	Boolean	If store is 1 the new value is stored permanently.

Example:

```
Input: AT*WKEY=Sesame,1
```

AT*WKEY?

ToDo: Should it be possible to read under all circumstances?

Syntax:

```
AT*WKEY?
```

Example:

```
Input: AT*WKEY? returns the encryption/authentication key at index 1.
```

6.3 AT*WKEYI Write Encryption/Authentication Key (with Index)

AT*WKEYI=

Write encryption/authentication key at any index.

Syntax:

```
AT*WKEYI=<index>,<pKey>,<store>
```

Input Parameters:

Name	Type	Description
index	Integer	1...4
pKey	String	The key to use. Max 63 characters.
store	Boolean	If store is 1 the new value is stored permanently.

Example:

```
Input: AT*WKEYI=2,Sesame,1.
```

6.4 AT*WACTKEY Active Encryption/Authentication Key

AT*WACTKEY=

Set the index of the active Encryption/Authentication Key

Syntax:

```
AT*WACTKEY=<index>,<store>
```

Input Parameters:

Name	Type	Description
index	Integer	1..4
store	Boolean	If store is 1 the new value is stored permanently.

AT*WACTKEY?

Get the index of the active Encryption/Authentication Key, 1..4.

Syntax:

```
AT*WACTKEY?
```

Example:

```
Input: AT*WACTKEY?
```

7 WLAN Access Point Commands

7.1 AT*WASSID Access Point SSID

AT*WASSID=

Sets the SSID for AP mode.

Syntax:

```
AT*WASSID=<pssid>
```

Input Parameters:

Name	Type	Description
pssid	String	The SSID to set. Max 32 characters.

Requires a reboot for the changes to take effect

AT*WASSID?

Gets the SSID for AP mode.

Syntax:

```
AT*WASSID?
```

Example:

```
AT*WASSID? *WASSID:<ssid>
```

7.2 AT*WACH Access Point Channel

AT*WACH=

Sets the channel for AP mode.

Syntax:

```
AT*WACH=<channel>
```

Input Parameters:

Name	Type	Description
channel	Integer	The channel to use. Valid channels are 1-11 for 2.4 GHz and 36, 40, 44, 48 for 5 GHz.

Requires a reboot for the changes to take effect

AT*WACH?

Gets the channel for AP mode.

Syntax:

```
AT*WACH?
```

Example:

```
AT*WACH? *WACH:<channel>
```

7.3 AT*WAAM Authentication Mode for AP

AT*WAAM=

Set the AP Authentication Mode.

Syntax:

```
AT*WAAM=<mode>
```

Input Parameters:

Name	Type	Description
mode	Integer	Authentication Mode: 0 = Open, 2 = WPA/WPA2 PSK (default)

Requires a reboot for the changes to take effect

AT*WAAM?

Get the AP Authentication Mode.

Syntax:

```
AT*WAAM?
```

Example:

```
Input: AT*WAAM?
```

8 WLAN Client Commands

8.1 AT*WSMAC WLAN MAC address

Can be used to clone a MAC address from an Ethernet device. This will allow layer 2 data to be bridged by one device. Can be combined with IP forwarding.

AT*WSMAC=

Set the WLAN MAC address. If set to all 0 or all FF the unit's default address will be used.

Syntax:

```
AT*WSMAC=<mac>
```

Input Parameters:

Name	Type	Description
mac	MACAddress	The MAC address to set.

Requires a reboot for the changes to take effect

AT*WSMAC?

Get the MAC address.

Syntax:

```
AT*WSMAC?
```

Example:

```
AT*WSMAC? *WSMAC:<mac>
```

8.2 AT*WSBM WLAN Bridge Mode

AT*WSBM=

Set the WLAN Bridge Mode.
In layer 2 tunnel mode all layer 2 data will be bridged over WLAN. Please note that this option uses a custom protocol and can only be used when the AP is another device of the same type.
Layer 2 cloned MAC only mode is used in combination with AT*WSMAC. In this mode only data from the cloned MAC will be bridged over WLAN. When using MAC clone two devices will use the same MAC - hence there is no way for a DHCP server to distinguish them from one another and both devices will receive the same IP address.
In layer 3 IP forward mode IP data from all data will be bridged over WLAN. Please note that this mode can be combined with AT*WSMAC to enable layer 2 data for one device.

Syntax:

```
AT*WSBM=<mode>
```

Input Parameters:

Name	Type	Description
mode	Integer	The Bridge Mode to set. 0: Layer 2 tunnel 1: Layer 2 cloned MAC only 2: Layer 3 IP forward (default)

Requires a reboot for the changes to take effect

AT*WSBM?

Get the Bridge Mode.

Syntax:

```
AT*WSBM?
```

Example:

```
AT*WSBM? *WSBM:<mode>
```

8.3 AT*WSC Connect

AT*WSC

Connect to Access Points as specified in the Connection List.

Syntax:

```
AT*WSC
```

Example:

```
Input: AT*WSC returns OK if the connection succeeds, ERROR otherwise.
```

8.4 AT*WSCC Close Connection

AT*WSCC

Close WLAN connection in Station mode. If there is no connection but a connect as specified by the Connection List is in progress this is terminated.

Syntax:

```
AT*WSCC
```

Example:

```
Input: AT*WSCC, returns OK when the connection is closed.
```

8.5 AT*WSCP Connect Peer

AT*WSCP=

Connect to a WLAN AP.

Syntax:

```
AT*WSCP=<bssid>,<ssid>,<channel>
```

Input Parameters:

Name	Type	Description
bssid	MACAddress	If specified the AP BSSID must match this value, if left blank the BSSID is ignored.
ssid	String	The SSID of the AP to connect to. If not specified the AP BSSID must be specified for a connection to be possible.
channel	Integer	The channel to use. 0 for any channel in the channel map.

Example:

```
Input: AT*WSCP=8C8B83EE2ACB,dut,0 will return OK if the connection
succeeds, ERROR otherwise.
```

8.6 AT*WSSCAN Scan

AT*WSSCAN=

Scan the surroundings for access points with a specific Network Name (SSID) on a specified channel.

Syntax:

```
AT*WSSCAN=<pssid>,<channel>
```

Input Parameters:

Name	Type	Description
pssid	String	The SSID to scan for. Max 32 characters.
channel	Integer	The channel to scan for

Example:

```
Input: AT*WSSCAN=dutAP,1 will return 0...48 access points in the immediate
surroundings, then return OK.
```

AT*WSSCAN?

Scan the surroundings for access points. Will return 0...48 access points in the immediate surroundings, then return OK.

Syntax:

```
AT*WSSCAN?
```

Example:

```
Input: AT*WSSCAN? *WSSCAN:
[bssid],[ssid],[channel],[rssi],[authentication_suit],[unicast_
ciphers],[group_cipher]
```

8.7 AT*WSAM Authentication Mode for Station

AT*WSAM=

Set the Station Authentication Mode.

Syntax:

```
AT*WSAM=<mode>,<store>
```

Input Parameters:

Name	Type	Description
mode	Integer	Authentication Mode: 0 = Open, 1 = WEP64/128 (shared secret), 2 = WPA/WPA2 PSK (default), 3 = LEAP, 4 = PEAP
store	Boolean	If store is 1 the new value is stored permanently.

AT*WSAM?

Get the Station Authentication Mode.

Syntax:

```
AT*WSAM?
```

Example:

```
Input: AT*WSAM?
```

8.8 AT*WSRSS Read RSSI

AT*WSRSS?

Read RSSI value of the connection.

Syntax:

```
AT*WSRSS?
```

Example:

```
Input: AT*WSRSS? returns ERROR if there currently is no Station mode
connection.
```

8.9 AT*WSLNK Read Link Status

AT*WSLNK?

Read current WLAN link status.

Syntax:

```
AT*WSLNK?
```

Example:

```
AT*WSLNK? *WSLNK:<link_status>,<bssid>
```

8.10 AT*WSUSER User name for WLAN LEAP/PEAP authentication.

AT*WSUSER=

Set the user name.

Syntax:

```
AT*WSUSER=<userName>,<store>
```

Input Parameters:

Name	Type	Description
userName	String	The user name to set (max 63 characters)
store	Boolean	If store is 1 the new value is stored permanently.

Example:

```
Input: AT*WSUSER=Joe,1
```

AT*WSUSER?

Get the user name.

Syntax:

```
AT*WSUSER?
```

Example:

```
Input: AT*WSUSER?
```

8.11 AT*WSDOMAIN Domain for WLAN LEAP/PEAP authentication.

AT*WSDOMAIN=

Set the domain.

Syntax:

```
AT*WSDOMAIN=<domain>,<store>
```

Input Parameters:

Name	Type	Description
domain	String	The domain to set. Max 63 characters.
store	Boolean	If store is 1 the new value is stored permanently.

Example:

```
Input: AT*WSDOMAIN=Cool,1
```

AT*WSDOMAIN?

Get the domain.

Syntax:

```
AT*WSDOMAIN?
```

Example:

```
Input: AT*WSDOMAIN?
```

8.12 AT*WSPASS Pass phrase for WLAN LEAP/PEAP authentication.

AT*WSPASS=

Set the pass phrase.

Syntax:

```
AT*WSPASS=<passPhrase>,<store>
```

Input Parameters:

Name	Type	Description
passPhrase	String	The pass phrase to set. Max 63 characters.
store	Boolean	If store is 1 the new value is stored permanently.

Example:

```
Input: AT*WSPASS=Secret,1
```

AT*WSPASS?

Get the pass phrase.

Syntax:

```
AT*WSPASS?
```

Example:

```
Input: AT*WSPASS?
```

8.13 AT*WSCHL Channel list

AT*WSCHL=

Sets the Channel list for Station mode.

Syntax:

```
AT*WSCHL=<channelListStr>,<store>
```

Input Parameters:

Name	Type	Description
channelListStr	String	A comma separated string of channels to use. Valid channels are 1-11 for 2.4 GHz and 36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 132, 136, 140 for 5 GHz.
store	Boolean	If store is 1 the new value is stored permanently.

Example:

```
AT*WSCHL=<channel0>,<channel1>,<channel2>...,1
```

AT*WSCHL?

Gets the Channel list for Station mode.

Syntax:

```
AT*WSCHL?
```

Example:

```
AT*WSCHL? *WSCHL:<channel0>,<channel1>,<channel2>...
```

8.14 AT*WSCLC Clear the Connection list

AT*WSCLC=

Clears all the entries in the Connection list.

Syntax:

```
AT*WSCLC=<store>
```

Input Parameters:

Name	Type	Description
store	Boolean	If store is 1 the new value is stored permanently.

Example:

```
AT*WSCLC=1
```

8.15 AT*WSCLR Read the Connection list

AT*WSCLR=

Reads an entry in the Connection list.

Syntax:

```
AT*WSCLR=<index>
```

Input Parameters:

Name	Type	Description
index	Integer	The index (0..49) of the entry to read.

Example:

```
AT*WSCLR=2 *WSCLR:<2>,<bssid>,<ssid><channel>
```

AT*WSCLR?

Reads the list of Connections (Access Points) that the unit can use.

Syntax:

```
AT*WSCLR?
```

Example:

```
AT*WSCLR? returns *WSCLR:<index>,<bssid>,<ssid>,<channel> for each entry
in the list followed by OK.
```

8.16 AT*WSCLW Write an entry (AP) in the Connection list

AT*WSCLW=

Writes an entry in the Connection list. NOTE: If store is set to 1 all entries in the connection list will be stored.

Syntax:

```
AT*WSCLW=<index>,<bssid>,<ssid>,<channel>,<store>
```

Input Parameters:

Name	Type	Description
index	Integer	The index (0..49) of the AP to write.
bssid	MACAddress	The BSSID of the AP.
ssid	String	The SSID of the AP. Max 32 characters.
channel	Integer	The channel of the AP.
store	Boolean	If store is 1 the new value is stored permanently.

Example:

```
AT*WSCLW=0,00026F668FA8,dutAP1,11,1 AT*WSCLW=1,00026F668FA8,dutAP2,6,1
```

9 Informational Commands

9.1 AT*AILVI Local Version Info

AT*AILVI?

Reads the local version info for the product

Syntax:

```
AT*AILVI?
```

Example:

```
AT*AILVI? *AILVI:<vendor>,<fw_version>
```

9.2 AT*AIMAC Read MAC

AT*AIMAC=

Reads the the MAC for the specified interface

Syntax:

```
AT*AIMAC=<interface>
```

Input Parameters:

Name	Type	Description
interface	Integer	The MAC to get. 0: Ethernet 1: WLAN 2: Bluetooth

Example:

```
AT*AIMAC=<interface> *AIMAC:<mac>
```

10 Miscellaneous Commands

10.1 AT*AMLI Login

AT*AMLI=

Log in to the AT command interface

Syntax:

```
AT*AMLI=<password>
```

Input Parameters:

Name	Type	Description
password	String	The password set using AT*AMPW

Example:

```
AT*AMLI=<password>
```

AT*AMLI?

Returns 1 if logged in

Syntax:

```
AT*AMLI?
```

Example:

```
AT*AMLI?
```

10.2 AT*AMLO Logout

AT*AMLO

Log out from the AT command interface

Syntax:

```
AT*AMLO
```

Example:

```
AT*AMLO
```

10.3 AT*AMPW Password

AT*AMPW=

Set password to the AT command interface

Syntax:

```
AT*AMPW=<password>,<store>
```

Input Parameters:

Name	Type	Description
password	String	Max length is 16 characters
store	Boolean	If store is 1 the new value is stored permanently.

10.4 AT*AMSTAT System status

Get the system status.

AT*AMSTAT=

Get the system status.

Syntax:

```
AT*AMSTAT=<verbose>
```

Input Parameters:

Name	Type	Description
verbose	Boolean	0: Terse, 1: Verbose.

Example:

```
AT*AMSTAT=0 *AMSTAT: Uptime:25 *AMSTAT: WLAN Mode:Station,
MAC:02:01:2E:00:24:00, state:3 *AMSTAT: Connected to
AP:02:01:2E:00:28:00, channel:1, rssi:-35 *AMSTAT: Bluetooth:
MAC:8C:8B:83:EE:2A:E6, State:1 *AMSTAT: Local name:dut *AMSTAT: Ethernet:
MAC:02:00:2E:00:24:00, State:1, Type:1 *AMSTAT: IP_ADDR:192.168.0.99 OK
```

10.5 AT*AMESS Event and Status Subscriber

AT*AMESS=

Set event and status subscriber configuration

Syntax:

```
AT*AMESS=<mac_addr>,<eth_type>,<ip_addr>,<udp_
port>,<protocol>,<store>
```

Input Parameters:

Name	Type	Description
mac_addr	String	MAC address of event subscriber. Only used when protocol bit 1 is set
eth_type	Integer	The 16 bit Ethernet type to use. Only used when protocol bit 1 is set
ip_addr	String	IP address of event subscriber. Only used when protocol bit 2 is set
udp_port	Integer	The UDP port to use. Only used when protocol bit 2 is set
protocol	Integer	The protocol to use for sending events. Bit 0: Send events over TCP AT connections Bit 1: Send events over Layer-2 (mac_address must be specified) Bit 2: Send events over Syslog
store	Boolean	If store is 1 the new value is stored permanently.

10.6 AT*AMEECM Execute Easy Configuration Mode

Executes the specified Easy Configuration Mode

AT*AMEECM=

Executes the supplied Easy Configuration Mode.

Syntax:

```
AT*AMEECM=<mode>
```

Input Parameters:

Name	Type	Description
mode	Integer	The mode number to execute.

Example:

```
AT*AMEECM=3 OK
```

10.7 AT*AMECFL Read/Write Easy Configuration Modes Function List

Reads/Writes the list of supported Easy Configuration Modes

AT*AMECFL=

Sets the list of supported Easy Configuration Modes.

Syntax:

```
AT*AMECFL=<functions>,<store>
```

Input Parameters:

Name	Type	Description
functions	String	Comma-separated string of up to 15 modes and their order to be supported. Valid modes are 1 to 15. Using mode=0 (INVALID MODE) will terminate the list at the given position.
store	Boolean	If store is 1 the new value is stored permanently.

Example:

```
AT*AMECFL=1,2,3,4,5,6,1 OK
```

10.8 AT*AMTFTP TFTP Upgrade

AT*AMTFTP=

Trigger a firmware update via TFTP. Device will automatically be rebooted into bootloader mode.

Syntax:

```
AT*AMTFTP=<device_ip>,<server_ip>,<filename>
```

Input Parameters:

Name	Type	Description
device_ip	NetworkAddress	The IP that the device shall use during the upgrade procedure
server_ip	NetworkAddress	TFTP server IP address
filename	String	Firmware filename (.fwz)

10.9 AT*AMPID Product ID

AT*AMPID?

Get product ID

Syntax:

```
AT*AMPID?
```

Example:

```
AT*AMPID? *AMPID:<vendorId>-<platformId>-<variantId>
```

10.10 AT*AMGD General Data

General data storage for custom data

AT*AMGD=

Short description for AT*AMGD=

Syntax:

```
AT*AMGD=<data>,<store>
```

Input Parameters:

Name	Type	Description
data	String	A custom string to store. Max length is 32 characters.
store	Boolean	If store is 1 the new value is stored permanently.

Example:

```
Input: AT*AMGD=1,1,1,1 gives OK when...
```

AT*AMGD?

Read general data

Syntax:

```
AT*AMGD?
```

Example:

```
AT*AMGD? *AMGD:<data>
```

10.11 AT*AMTL TCP Listener

Configures the AT over TCP server

AT*AMTL=

Set TCP listener settings

Syntax:

```
AT*AMTL=<port>,<enable>,<store>
```

Input Parameters:

Name	Type	Description
port	Integer	TCP port to listen for incoming connections
enable	Boolean	0: Disables TCP Listener 1: Enables TCP Listener
store	Boolean	If store is 1 the new value is stored permanently.

Requires a reboot for the changes to take effect

AT*AMTL?

Get TCP listener settings

Syntax:

```
AT*AMTL?
```

Example:

```
AT*AMTL? *AMTL:<port>,<enabled>
```

10.12 AT*AMBD Bridging Disable

AT*AMBD=

Set bridging enable/disable

Syntax:

```
AT*AMBD=<disable>,<store>
```

Input Parameters:

Name	Type	Description
disable	Boolean	0: Bridging Enabled 1: Bridging Disabled
store	Boolean	If store is 1 the new value is stored permanently.

10.13 AT*AMLCR Layer 2 Configuration Receiver

Configure AT over layer 2 (Ethernet)

AT*AMLCR=

Set AT over layer 2 configuration

Syntax:

```
AT*AMLCR=<eth_type>,<enable>,<store>
```

Input Parameters:

Name	Type	Description
eth_type	Integer	16 bit Ethernet type that should be used for AT commands
enable	Boolean	0: Disable AT over Ethernet 1: Enable AT over Ethernet
store	Boolean	If store is 1 the new value is stored permanently.

Requires a reboot for the changes to take effect

AT*AMLCR?

Get AT over layer 2 configuration

Syntax:

```
AT*AMLCR?
```

Example:

```
AT*AMLCR? *AMLCR:<eth_type>,<enabled>
```

10.14 AT*AMREBOOT Reboot

AT*AMREBOOT

Reboot device

Syntax:

```
AT*AMREBOOT
```

11 S Registers

Miscellaneous Registers

Register	Name	Value Range	Default Value	Description
1000	Reserved			
1001	Reserved			
1002	Reserved			
1003	Reserved			
1004	Reserved			
1005	Reserved			
1006	Reserved			
1007	Easy Config LED Mode	0..3	3	Bit mask representing Easy Config LED Mode when smart mode is finished Value 0: Link quality LEDs are off Value 1: WLAN RSSI Value 2: BT Link Quality Value 3: Auto (default) show WLAN if active otherwise show BT if active
1008	Reserved			
1009	Reserved			
1010	Reserved			
1011	Reserved			
1012	Reserved			
1013	Diagnose Mode	0..4294967296	0x00010001	Set diagnose mode bitmask. The following events will be sent when the bit is set: Bit 0: *WSCO - Connection to AP up *WSCC - Connection to AP down *WASA - Station has connected *WASR - Station has disconnected *WSDST - Digital signal transition (only applicable if SETTINGS_ROS_WL_ROAMING_LIST_TRIGGER_INPUT is 1) *WSRSSC - RSSI value falls below or rises above the value in SETTINGS_ROS_WL_TRIGGER_SCAN_RSSI Bit 1: *WSRSS - RSSI periodically sent while connected, with interval set by SETTINGS_WTS_RSSI_POLL_INTERVAL *WSCH - Used WLAN channel, sent upon connection setup Bit 2: *WSSCAN - Background scan result, regardless of the current roaming operation. Note: Will not work while a connection is in progress, use AT*WSCC to cancel. Bit 3: *WSSCAN - Background scan result for possible handover candidates. *WSRHC - Roaming handover candidate *WSRH - Roaming handover Bit 16: *BCI - Connection indication (incoming connection) *BCO - Connection opened *BCC - Connection closed *BDST - Digital signal transition (only applicable if SETTINGS_ROS_BT_ROAMING_LIST_TRIGGER_INPUT is 1) *BLQC - Link Quality value falls below or rises above the value in SETTINGS_ROS_BT_TRIGGER_SCAN_LINK_QUALITY Bit 17: *BLQ - Link Quality periodically sent while connected, with interval 5000ms

Miscellaneous Registers (continued)

Register	Name	Value Range	Default Value	Description
				Bit 18: *BI - Background scan result, regardless of the current roaming operation. Bit 19: *BI - When background scan is active results for the currently connected device and possible handover candidates are sent. *BRSS - When background scan is active an aproximate RSSI value for the current connection is sent periodically. *BRHC - Roaming handover candidate *BRH - Roaming handover
1014	Easy Config Mode Timeout	0..4294967296	5000	Maximum time to wait for first push on SMART button in milliseconds
1015	Radio Mode	0..3	3	Configures which radios should be enabled 0: All radio off 1: Enable only Bluetooth 2: Enable only WLAN 3: Enable Bluetooth and WLAN

Bluetooth Registers

Register	Name	Value Range	Default Value	Description
2000	Reserved			
2001	Reserved			
2002	Reserved			
2003	Reserved			
2004	Reserved			
2005	Reserved			
2006	Reserved			
2007	Reserved			
2008	Reserved			
2009	Reserved			
2010	Max Number of Connections	0..7	1	
2011	Reserved			
2012	External Connection Control	0..1	0	
2013	Reserved			
2014	Reserved			
2015	Reserved			
2016	Reserved			
2017	Reserved			
2018	Background Scanning Interval	0..4294967296	5000	Time between background scans in milliseconds

WLAN Registers

Register	Name	Value Range	Default Value	Description
3000	Reserved			
3001	Reserved			
3002	Reserved			
3003	Reserved			
3004	Reserved			
3005	Background Scanning Interval	0..4294967296	5000	Time between background scans in milliseconds
3006	Reconnect Interval	0..4294967296	5000	The interval in milliseconds
3007	RSSI poll interval	0..65535	1000	RSSI poll interval in milliseconds
3008	Reserved			

Roaming WLAN Registers

Register	Name	Value Range	Default Value	Description
4000	Roaming list trigger RSSI	0..1	1	0: Do not trig on RSSI 1: Trig on RSSI (default)
4001	Roaming list trigger input	0..1	1	0: Do not trig on digital input 1: Trig on digital input (default)
4002	Roaming RSSI diff threshold	0..127	10	When the difference between the connected AP RSSI and the RSSI from the best AP from background scan exceeds this threshold a roaming handover is done.
4003	Trigger Scan RSSI	-128..127	-70	Defines at what RSSI level a background scan should be initiated to find a better connection.

PROFINET Registers

Register	Name	Value Range	Default Value	Description
5100	Reserved			
5101	Profinet Prioritization	0..1	0	If this is set to 1 packets with Ethernet type matching the Prioritized Ethernet Type
5102	Prioritized Ethernet Type	0..65535	0x8892	The 16-bit Ethernet type to prioritize
5103	Reserved			
5104	Reserved			

Box 4126
300 04 Halmstad, Sweden

info@hms.se
SCM-1202-004 1.2.3741 / 2017-03-30 15:26