A.N.Solutions
INDUSTRIAL GRADE WIRELESS

Smart MAC Suite

AT Command Reference v3.2

AT-ANY SMS Pro

AT-ANY SMS Pro allows the development of complex applications and supports tree network topologies.

AT-ANY SMS Base

AT-ANY SMS Base offers some basic functionality and is designed for simple network topologies and evaluation purpose.

20 February 2020

A.N. Solutions GmbH
Am Brauhaus 5
01099 Dresden
Germany

List of Contents

1 Glossary

Coordinator:	This is a device role for Bricks and Dongles. A coordinator creates a network and is waiting for end devices and FF devices to connect to it. This is comparable to a network host that takes over the coordination in the network. There is only one coordinator in a network.
FF-device:	This is a device role for Bricks and Dongles. A foundation fieldbus (FF) device can exchange data with other ff devices and the coordinator, but cannot create or coordinate a wireless sensor network (WSN).
END-device:	This is a device role for Bricks and Dongles. An end device can only connect to a coordinator and only send or receive data from a coordinator. It cannot create networks.

2 Introduction

Bundled with AT-ANY platform solution that consists of AT-ANY Modules, USB-Dongles and Brick development boards, A.N. Solutions' Smart MAC Suite (“SMS”) software tool helps to develop and deploy a wireless sensor network (WSN) based on the IEEE 802.15.4 standard with a simplified, adjustable programming interface that is widely common.

A.N. Solutions' “SMS” provides control over the majority of AT-ANY platform features through any provided communication interface using a standardized AT-command set.

It provides capabilities for easy setup of wireless networks for specific demands without developing custom firmware and thus enables flexible commissioning proceedings as well as easy debugging and testing. It allows the setup of flexible routines for installation and maintenance of AT-ANY based solutions and simplifies network monitoring at the same time.

- Smart MAC Suite running on the AT-ANY platform provides the following advantages for an end-user: AT-ANY modules can be connected directly to a host processor as communication extension whereas the interfaces of the module can enhance the system capabilities by adding additional sensors and actuators.
- The user can program and operate the AT-ANY platform without embedded programming knowledge by simply using S-Register mappings and AT-Commands.
- Smart MAC Suite provides IEEE 802.15.4 functionality to users. Users can set up basic network topologies (star, peer-to-peer) using static routing. Data packages can be transmitted directly or indirectly, acknowledged or broadcasted.
- Sensor support for a variety of AT-ANY based boards is included: Temperature sensor support on AT-ANY-BRICK (LM73), Humidity/Temperature sensor and Acceleration sensor on AT-ANY-BRICK-SC (Silicon Labs SI7006-A20, Sensirion SHT21 and Bosch BMA222E) as well as support for up to 10 GPIO lines are already included.
- AT-ANY900 and AT-ANY2400 module's integrated flash memory can be accessed.
- Integration of external host processor can be done directly into the AT-ANY module.
- More hardware support such as additional sensors (for example using ATMEGA TWI), more GPIO lines, ATMEGA AD/DA, SPI and UART features, interrupt and wakeup behaviour or different UART baud rates can be added by customer or on request.
- Advanced network topologies (Tree and Mesh) are available with SMS Pro version
- Programming library is included in SMS Pro for custom extensions to integrate into the AT-ANY module

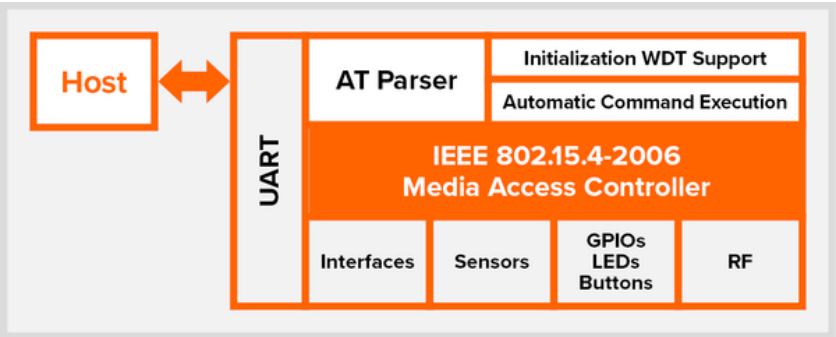


Figure 1: basic SMS structure

2.1 SMS Usage Scenarios

As shown in Figure 1, Smart MAC Suite consists of several software parts. In addition to the support of the RF portion, it also takes care of the GPIO control logic, the supported temperature, humidity and acceleration sensors on the AT-ANY Brick boards (different sensor support is planned for future releases, currently supported sensors can be found in table 2) and several other interfaces, e.g. the SPI, to access the module's internal flash and the UART interface to interact with a host. SMS is utilizing a Media Access Controller (MAC), which implements the IEEE802.15.4-2006 functionality, including:

- direct data transmission (acknowledged or unacknowledged)
- indirect data transmission (acknowledged or unacknowledged)
- data broadcasting (Pro version only)
- device scans (searching for general or special networks)
- device association and disassociation (joining or leaving a network)
- and several others, which can be found in [1].

Since there are applications out in the field, which cannot afford an additional host intelligence, A.N. Solutions implemented two features in SMS, which allow it to run a network node **without any further intelligence**. This means that **no host is required**.

This is realized with the two white blocks in Figure 1:

- “Initialization/WDT support”
- “Automatic command execution”

If a setup described and illustrated with several examples below included in the AT-ANY Development Kit with the above-mentioned blocks, it is possible to run star networks (Base and Pro), tree networks (Pro only) and mesh networks (Pro only) without a host. Therefore, the initialization can be configured in several aspects depending on device type for several scenarios using AT typical profiles. The “Automatic command execution” block executes a previously stored shadow command from the active profile and executes it on an event or repetitively in adjustable periods.

As stated above, SMS Pro has a build-in frames redirection feature, which can be used to forward data to different network points. Using this feature allows to create tree networks without host intelligence. More general routing algorithms are out of the scope of Smart MAC Suite, since they are in the scope of a network layer, which can be executed on the side of the host intelligence.

There are two versions of this software. This document describes the Pro (full featured) version. The basic version does not support data broadcasting and *FF device* functionality, so all commands or register settings (especially *S220*) referring to this functionality will only be available in SMS Pro.

SMS Base provides some basic functionality designed for simple network topologies and evaluation purposes. It can be used to set up Coordinator - End device (star, peer-to-peer) topologies which are useful in basic sensor networks for instance.

SMS Pro provides additional *FF device* functionality. The data redirect feature enables users to set up basic tree network topologies. SMS Pro has an additional option to enable routing from LWmesh implementation to establish self-forming mesh networks. Beyond that, SMS Pro is designated to be a code base for customer requested extensions.

More advanced network topologies, tree and mesh structures, can be facilitated with the integrated support of different routing schemes in SMS Pro. A separate “functional layer” can be enabled to support typical application scenarios.

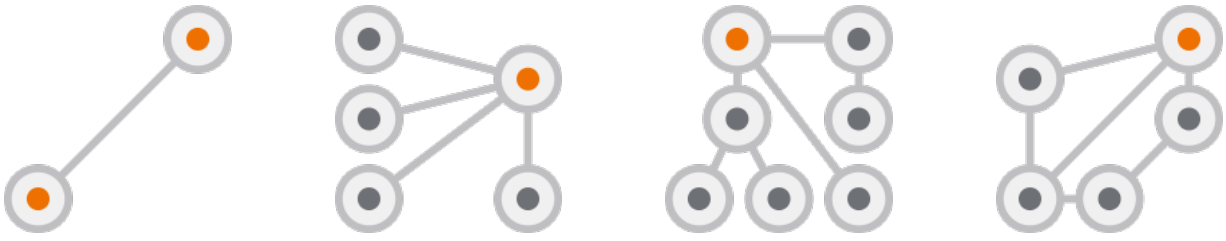
Furthermore, SMS Pro is available as a library version, which empowers the customer to fully integrate all external host intelligence into the @ANY module. It helps to develop further extensions, such as a custom routing scheme or support for additional sensors independently from the already provided feature set. For development, an open source software toolchain can be used.

Feature comparison chart:

Feature	SMS Base	SMS Pro
Peer-to-peer network	✓	✓
Star network	✓	✓
Tree network	-	✓
Mesh network	-	✓
Frame redirection	-	✓
Temperature sensor support	✓	✓
Humidity sensor support	✓	✓
Acceleration sensor support	✓	✓
ADC support	-	✓
GPIOs (8 in/output lines, 2 lines for input and event trigger)	✓	✓
Sleep mode	✓	✓
Data transmission (direct / indirect)	✓	✓
Data transmission (acknowledged)	✓	✓
Data transmission (non-acknowledged, broadcast)	-	✓
Radio register access	-	✓
EEPROM content accessible via UART/USB	✓	✓
@ANY900 and @ANY2400 flash memory support	-	✓
CW Mode	-	✓

Table 1 SMS Features

Available Networking structures with ANS Smart MAC Suite:



Peer-to-Peer Network **Star Network** **Tree Network** **Mesh Network**

A.N.Solutions

Am Brauhaus 5 • 01099 Dresden • Germany • **phone** +49 351 30900 199 • **fax** +49 351 30900 189 • info@an-solutions.de • **www.an-solutions.de**

2.2 Smart MAC Suite Program Flow

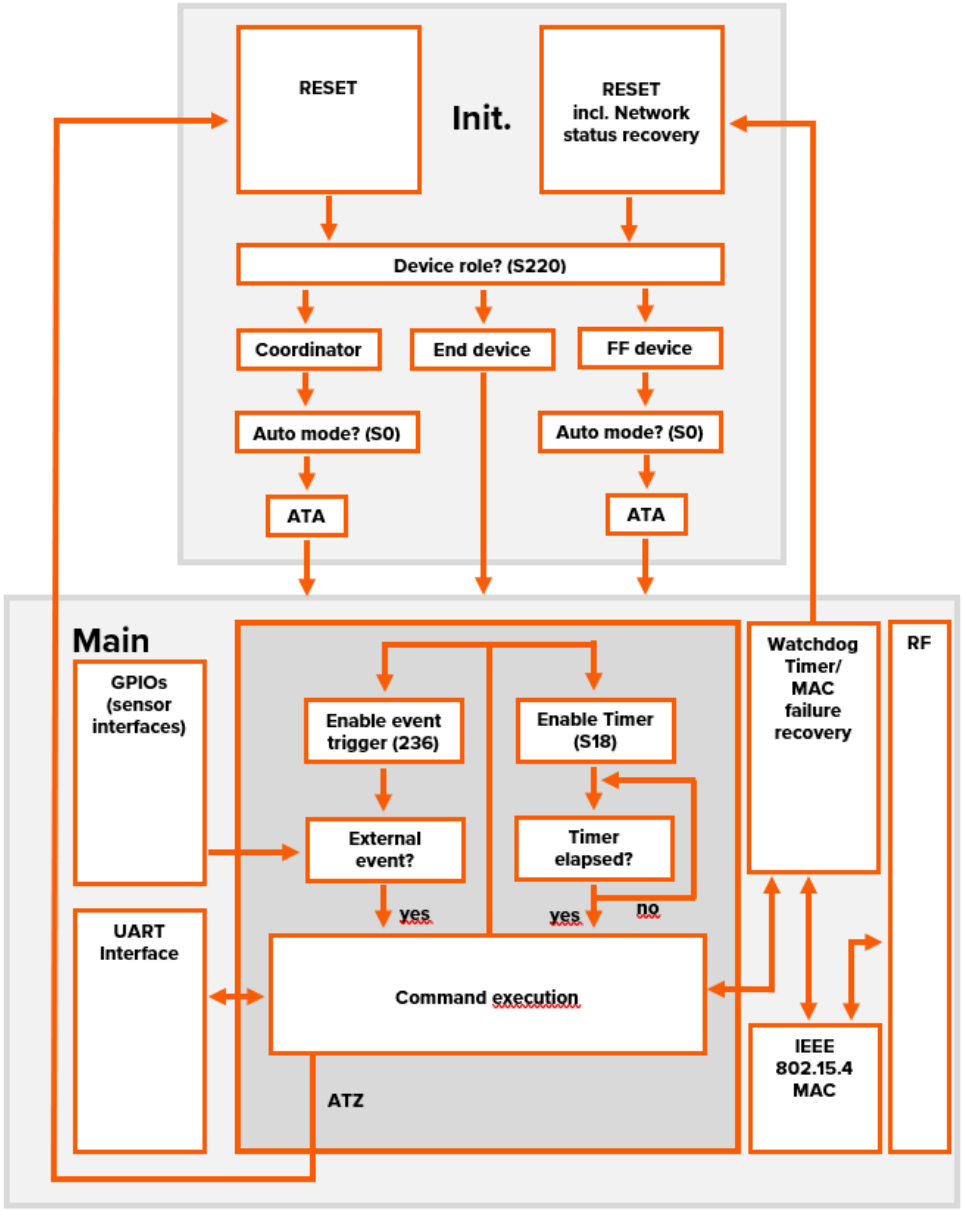


Figure 2 Smart MAC Suite Programme Flow

3 AT Commands

3.1 General Commitments

- Each line starts with AT
- Letters are not case sensitive (except 'AT&Waddr' used to save a changed IEEE address)
- A command line can consist of more than one command, they will be processed one after another until either the last command has successfully been processed or a command returns with an error indication. (except 'ATA' for End devices, where execution stops right after association). Ambiguous command sequences can be explicitly separated using the 'X' character.
- The maximum length of a command line is 114 characters.

3.2 Overview AT Commands

3.3 Command Descriptions

ATA – Activate (FF device, Coordinator) / **Associate** (End device)

Note: ATA sets device into an 'active network mode', afterwards some commands and S-register writes will be denied. If an ATA command was issued, devices are able to react on network events, see chapter Asynchronous Messages. When a device is in 'active network mode', changes for certain networking related parameters do not take effect until a re-association has been performed.

Syntax: ATA

Supported device types	Description	Involved S-registers	Result
FF device	Set up network's parameters, enable receiver	S200, S202, S208, S209, S212, S216	OK
Coordinator	Set up network parameters, start a new network, allow end devices to associate	(S200), S202, S208, S209	OK
End device	Set up network parameters, associate with network (coordinator)	S202, S208, S209, S221, out: S200	ASSOCIATED: *addr* * OK in case of success or NOT ASSOCIATED (ec)** ERROR

* *addr*: short address as assigned by coordinator
** *ec*: MAC return code as described in [1] (see Appendix A)

>Back to Overview<

A.N.Solutions

Am Brauhaus 5 • 01099 Dresden • Germany • **phone** +49 351 30900 199 • **fax** +49 351 30900 189 • info@an-solutions.de • **www.an-solutions.de**

ATD – Send Data / Poll Data (End device)

This is used to send data to other devices. Note that a successful completion of this command just means that the data was successfully buffered in the data transfer queue. At this time the data has not been transferred yet. See chapter **Asynchronous Messages** to find out how notification about successful data transfer works.

General rules:

- If a target is a non-associated device, data are always sent directly and acknowledged.
- If a target is an associated end device, data can be transmitted directly or indirectly depending on the device capabilities (communicated during association).
- IEEE address FF.FF.FF.FF.FF.FF.FF.FF and PANID 0xFFFF are used for broadcasting of the messages (make sure to use the "-" operand to send data without acknowledgement).

Syntax:	FF device, Coordinator:	End device:
	ATD<target> ↵ <len><data> ↵	ATD ↵ <len><data> ↵
	ATD<target>; ↵ <ASCII data> ↵	ATD; ↵ <ASCII data> ↵
	ATD<target><register set> ↵	ATD<register set> ↵
	ATD<target>/ ↵	ATD/ ↵
		ATD? ↵

Supported device types	**Arguments**	**Description**	**Involved S-registers**	**Result**
FF device, Coordinator	<target> <_data_>	determine target address and send data	(S202)	ID *xxx*: *n* BYTES TO *addr** OK or ERROR (format failure, device not in network mode, target device not known or end device not associated)
	<*target*>/	determine target address and send (redirect) last received data	(S202)	
End device	<_data_>	send data to coordinator		
	/	send last received data (back) to coordinator		

>Back to Overview<

A.N.Solutions

Am Brauhaus 5 • 01099 Dresden • Germany • **phone** +49 351 30900 199 • **fax** +49 351 30900 189 • info@an-solutions.de • **www.an-solutions.de**

From ATD				
Supported device types	**Arguments**	**Description**	**Involved S-registers**	**Result**
	?	poll for pending data		OK (if data is pending a data message will follow)
all	- (right after the <_target_> argument)	send data without acknowledge		OK or ERROR (see above)

* *xxx*: the assigned data id, *n*: size of data package, *addr*: destination address

<**target**>	format:	
	x(..x) :	up to 4 digits (hexadecimal) are interpreted as short address
	xxxxx(..x) :	more than 4 digits (up to 16) are interpreted as extended address using PANID from S202
	x(..x):x(..x) :	PANID: short address (as described above)
	x(..x):xxxxx(..x) :	PANID: extended address (as described above)
	S=n :	use address stored as entry no. n (decimal, see AT&Z)
	L :	use last target again
<**len**>	format:	The Byte before a <*data*> block. The next <*len*> received characters will be dealt as the message to transmit. Transmission is withheld until enough characters given.
<**data**>	format:	A binary block following the <*len*> number of bytes to transmit. The maximum number of bytes is 104.
<**ASCII data**>	format:	A line of characters (ASCII) terminated by A (excluded) CR character. The maximum number of characters is 104.
<**register set**>	format:	(more of them can be used in one send command)
	S<*Reg.id*> :	binary content of 8-bit S-register will be transmitted
	SX<Reg.id> :	binary content of extended S-register will be transmitted (size is register dependent)
	SA<*Reg.id*> :	decimal content of 8-bit S-register will be transmitted as a 3-byte ASCII string

>Back to Overview<

A.N.Solutions

Am Brauhaus 5 • 01099 Dresden • Germany • **phone** +49 351 30900 199 • **fax** +49 351 30900 189 • info@an-solutions.de • **www.an-solutions.de**

Examples:

ATD1234:5678;
HELLO
ID <xxx> : 5 BYTES to <target>
OK
+SENT: ID <xxx>
or
ATD1234:5678
{0x05; 0x48, 0x45, 0x4C, 0x4C, 0x4F}
ID <xxx> : 5 BYTES TO <target>
OK
+SENT: ID <xxx>
both send ‘HELLO’ to target PANID 0x1234, Short Address 0x5678

ATD2
{0x02, 0x48, 0x49}
ID <xxx> : 2 BYTES TO 0000
OK
+SENT: ID <xxx>
send ‘HI’ to associated device 2

ATD
{*0x02, 0x4C, 0x4F*}
ID <xxx> : 2 BYTES TO 0000
OK
+SENT: ID <xxx>
sends 'LO' from associated device to its coordinator.

ATD1234:05060708;
HI
ID <xxx> : 2 BYTES TO 1234:00.00.00.00.05.06.07.08
OK
+SEND FAILURE: ID <xxx> (E9)
sends 'HI' to PANID 0x1234, Long Address 0x00.00.00.00.05.06.07.08
Transmission failed with error **0xE9**, because recipient did not send an acknowledgement.

AT&Z0=aaaa:bbbb
OK
ATSX240=2D313044
OK
ATDS=0SX240SA230S243SA235
ID <xxx> : 11 BYTES TO aaaa:bbbb
OK
+SENT: ID <xxx>
builds string "D01-00x-00x" (using user data registers and reading GPIO P0 and GPIO P1) and sends it to the stored target no. 0, initially set to aaaa:bbbb.

>Back to Overview<

A.N.Solutions

Am Brauhaus 5 • 01099 Dresden • Germany • **phone** +49 351 30900 199 • **fax** +49 351 30900 189 • info@an-solutions.de • **www.an-solutions.de**

ATDFFFFFFFFFFFFFFF-;
HELLO
ID <xxx> : 5 BYTES TO FF.FF.FF.FF.FF.FF.FF.FF
OK
+SENT: ID <xxx>
sends 'HELLO' as broadcast to targets that have the same PANID

ATDFFFF:FFFFFFFFFFFFFFFF-;
HELLO
ID <xxx> : 5 BYTES TO FFFF:FF.FF.FF.FF.FF.FF.FF.FF
OK
+SENT: ID <xxx>
sends 'HELLO' as broadcast to all reachable targets

When doing several consecutive transmissions, the next transmission shall be initiated after the +SENT: or +SEND FAILURE: transmission confirmation is received, not right after the OK. Otherwise undesired behaviour can occur. The OK does not mean that the transmission is fully completed, it just confirms that the command has been successfully parsed and the transmission request is handed over to the lower layers.

ATI – Device Information

Syntax: ATI<n>

Supported device types	Argument	Result
all	(0)	<ISM band>
all	1	<firmware rev.>
all	2	<Build-ID>
all	3	ANS – Smart MAC Suite V-<firmware rev.>
all	4	<board type>
all	5	<programme memory CRC>
all	6	<radio chip type>
all	7	<license owner>
all	any	OK (argument in 1 ... 6 (7)) or ERROR (otherwise)

>Back to Overview<

ATH – De-activate/Disassociate

Syntax: ATH<n>

Supported device types	Argument	Description	Involved S-registers	Result
FF device	(0)	Disable receiver	none	OK
End device	(0)	Disassociate from current network (from coordinator)	none	OK
Coordinator	(0)	not supported (use ATH1 to force)	none	ERROR
all	1	force MAC reset	none	OK

ATO – Enable Network

Note: ATO sets device into an 'active network mode', some commands and S-register writes will be denied.

Syntax: ATO

Supported device types	Description	Result
FF device	set up all network parameters to enable device to send data (go online, similar to ATA, no receiver enable)	OK or ERROR (wrong device type)

ATQ – Result Code Option

Syntax: ATQ(0)

Supported device types	Argument	Description	Result
all	0	(ATQ0) compatibility only	OK (for ATQ0) ERROR (otherwise)

>Back to Overview<

A.N.Solutions

Am Brauhaus 5 • 01099 Dresden • Germany • **phone** +49 351 30900 199 • **fax** +49 351 30900 189 • info@an-solutions.de • **www.an-solutions.de**

ATV – Result Code Format

Syntax: ATV1

Supported device types	Argument	Description	Result
all	1	result codes as words (ATV1) compatibility only	OK (for ATV1) or ERROR (otherwise)

ATS – Read/Write S-Register

Syntax: **ATS<Reg.ID>?** **ATSX<Reg.ID>?**
ATS<Reg.ID>=<value> **ATSX<Reg.ID>=<value>**

Supported device types	Argument	Description	Result
all	<Reg.ID>?	Read decimal 8-bit value	OK or ERROR (index out of range)
all	<Reg.ID>=<value>	Write decimal 8-bit value	OK or ERROR *
all	X<Reg.ID>?	Read hexadecimal value (size is register dependent)	OK or ERROR (index out of range)
all	X<Reg.ID>=<value>	Write hexadecimal value (size is register dependent)	OK or ERROR *

* index or value out of range, read-only-register, parser state forbids register change or extended write not supported for this register (ATSX..) See chapter S - Registers for detailed register descriptions.

>Back to Overview<

A.N.Solutions

Am Brauhaus 5 • 01099 Dresden • Germany • **phone** +49 351 30900 199 • **fax** +49 351 30900 189 • info@an-solutions.de • **www.an-solutions.de**

ATZ – Device Reset

Syntax: ATZ<n>, ATZ-

Supported device types	Argument	Description	Involved S-registers	Result
all	(0), 1	Reset device, load register profile no. <*n*>	all	OK (ERROR) *
all	-	Reset device, load default register profile (see AT&Y)	all	OK (ERROR) *

* Something went wrong during device initialization so there's either some hardware problem or the profile to load is corrupted.

AT&C – DCD Options

Syntax: AT&C(0)

Supported device types	Argument	Description	Result
all	(0)	HW handshake is not supported compatibility only	OK (for AT&C0) or ERROR (otherwise)

AT&D – DTR Options

Syntax: AT&D(0)

Supported device types	Argument	Description	Result
all	(0)	HW handshake is not supported compatibility only	OK (for AT&C0) or ERROR (otherwise)

>Back to Overview<

A.N.Solutions

Am Brauhaus 5 • 01099 Dresden • Germany • **phone** +49 351 30900 199 • **fax** +49 351 30900 189 • info@an-solutions.de • **www.an-solutions.de**

AT&F – Load Factory Defaults

Syntax: AT&F

Supported device types	Involved S-registers	Description	Result
all	all	Load default register values from program ROM to the active profile	OK or ERROR (denied in active network states)

AT&K – Local Flow Control Option

Syntax: AT&K(0)

Supported device types	Argument	Description	Result
all	(0)	HW handshake is not supported compatibility only	OK (for AT&C0) or ERROR (otherwise)

AT&Q – Communication Mode

Syntax: AT&Q(0)

Supported device types	Argument	Description	Result
all	(0)	HW handshake is not supported compatibility only	OK (for AT&C0) or ERROR (otherwise)

>Back to Overview<

A.N.Solutions

Am Brauhaus 5 • 01099 Dresden • Germany • **phone** +49 351 30900 199 • **fax** +49 351 30900 189 • info@an-solutions.de • **www.an-solutions.de**

AT&R – RTS/CTS Options

Syntax: AT&R(0)

Supported device types	Argument	Description	Result
all	(0)	HW handshake is not supported compatibility only	OK (for AT&C0) or ERROR (otherwise)

AT&W – Store profile

Syntax: AT&W<n>

Supported device types	Argument	Description	Result
all	(0),1	Write the active profile to profile *n* in NVRAM	OK or ERROR *
all	addr (note: this is case sensitive)	following AT+CF0=<*addr*> only: store the previously entered IEEE address to NVRAM and reset device	see ATZ-

* Something went wrong writing the NVRAM

AT&V – View

Syntax: AT&V<n>

Supported device types	Argument	Description	Result
all	(0)	View profile information	<*profile information*> OK *
all	1	View stored numbers For storing see AT&Z	<*stored number list*> OK *
Coordinator	2	View associated / known devices	<*device list*> OK

* Something went wrong reading the NVRAM

>Back to Overview<

Examples: AT&V

<device list>

idx: short address. / ieee address (capabilities)

00: 0001 / 00.04.00.04.00.04.00.04 (80)
01: 0002 / 00.06.00.06.00.06.00.06 (88)
02: 0000 / 00.08.00.08.00.08.00.08 (00)

The example shows three devices, devices 0 and 1 are currently associated, capabilities of 88 (device 1) means the devices receiver is always on so data can be sent directly, device 2 doesn't have an actual short address which means that this device is known to the coordinator but is currently not associated.

<stored number list>

idx: PAN Id : short / long address

STORED NUMBERS:
00: CAFE:0002
01: C0DE:00.06.00.06.00.06.00.06
04: BABE:0A.01.0B.01.0C.01.0D.01

<profile information>

ACTIVE PROFILE:
E1 F1 Q0 V1 &C0 &D0 &K0 &Q0 &R1 &S0
S00:001 S01:001 S02:043 S03:013 S04:010 S05:008
S10:000 S12:000 S17:016 S18:040 S25:005 S26:001 S38:020
IEEE ADDRESS: 08.07.06.05.04.03.02.01
DEVICE ROLE: COORDINATOR
PAN ID: CAFÉ SHORT ADDRESS: 0000
CHANNEL: 20 (0) CHANNEL MASK: 07FFF800
STORED COMMAND: S231=8

STORED PROFILE 0:
E1 F1 Q0 V1 &C0 &D0 &K0 &Q0 &R1 &S0
S00:000 S02:043 S03:013 S04:010 S05:008 S17:016 S18:000
DEVICE ROLE: FF DEVICE
PAN ID: FFFF SHORT ADDRESS: 0000
CHANNEL: 20 (0) CHANNEL MASK: 07FFF800

STORED PROFILE 1 (DEFAULT):
E1 F1 Q0 V1 &C0 &D0 &K0 &Q0 &R1 &S0
S00:001 S02:043 S03:013 S04:010 S05:008 S17:016 S18:040
DEVICE ROLE: COORDINATOR
PAN ID: CAFÉ SHORT ADDRESS: 0000
CHANNEL: 20 (0) CHANNEL MASK: 07FFF800
STORED COMMAND: S231=8

>Back to Overview<

A.N.Solutions

Am Brauhaus 5 • 01099 Dresden • Germany • **phone** +49 351 30900 199 • **fax** +49 351 30900 189 • info@an-solutions.de • **www.an-solutions.de**

AT+C – Configure
(Set / Get PHY / MAC PIB attribute as described in [1], see Appendix B / Appendix C)

Syntax: AT+C<*attribute*>?, AT+C<*attribute*>=<*value*>

Supported device types	Argument	Description	Result
all	<*attribute*>?	Show attribute value	OK or ERROR (attribute unknown or not accessible)
all	<*attribute*>=<*value*>	Set attribute to value	OK or ERROR (attribute unknown, not accessible or value is out of range)

AT&Z – Store Target Address

Syntax: AT&Z<n>=<target>, AT&Z<n>

Supported device types	Argument	Description	Result
all	<n>=<target>	Store target address <target> at position <n>.	OKor ERROR *
all	<n>-	Delete entry at position <n> For viewing see AT&V1	OKor ERROR *

* Wrong address format or something went wrong writing the NVRAM

>Back to Overview<

AT&S – DSR Options

Syntax: AT&S(0)

Supported device types	Argument	Description	Result
all	(0)	HW handshake is not supported compatibility only	OK (for AT&C0) or ERROR (otherwise)

AT+R – Sleep

Syntax: AT+R

Supported device types	Involved S-registers	Description	Result
all	S17, S18, S236	Sleep **	OK (after wake-up) or ERROR

** A wake-up source should be configured beforehand! See register 236!

When entering sleep mode, the wireless transceiver of the device has to be disabled in order to save energy. This implies that during sleep mode the device is not capable of either sending or receiving data.

Due to the upcoming limitations of possible data loss introduced by this issue, it can be stated that it is not possible to put routing devices into sleep mode for all kinds of applications and network topologies.

Nevertheless, for some of the most common usage scenarios of WSN, energy can be saved by putting the routers into sleep mode. This is introduced by the fact that data transmission intervals in many WSN are very large compared to the idle periods.

The idle period is defined as when no data has to be sent or received by the device. If those idle periods are known in advance, the devices involved can enter sleep mode during these periods. This means that no data loss would occur while the devices are asleep, even without introducing a data polling scheme or something comparable.

A very simple case is the unidirectional data transmission over multiple hops, as illustrated in figure 1.

Figure 3: Simple unidirectional data transmission network structure

Once the transmission interval of the End Device is known to the Routers involved, they can safely enter sleep mode between two distinct transmissions and thus save energy. The impact on the Router's battery lifetime varies with transmission interval of the End Device.

>Back to Overview<

AT+S – Scan

Syntax: AT+S

Supported device types	Involved S-registers	Description	Result
all	S204, S222	Scan for networks	<*result list*> OK or ERROR (active network modes)

<*result list*> **:** List of coordinators found in the format:
PAN ID**:**ieee/short address / channel (channel page) / ed * (* energy detection)
"*xxxx:xxxx / nn (n) / nnn*" or
"*xxxx:xx.xx.xx.xx.xx.xx.xx.xx / nn (n) / nnn*" or
"NONE" if no network was found

AT&Y(n) – Select default profile

Syntax: AT&Y<n>

Supported device types	Argument	Description	Result
all	(0),1	Select profile no. *n* as default (used on HW reset)	OK or ERROR *

* Something went wrong writing the NVRAM

>Back to Overview<

ATRR – Radio Register Access

Note: Requires device to be in 'active network mode' in order to return values different from zero (refer to ATA and ATH commands)

Syntax: ATRR<*attr*>?, ATRR<*attr*>=<*value*>, ATRRX<*attr*>?, ATRRX<*attr*>=<*value*>

Supported device types	Argument	Description	Result
all	<*attr*>?	Show transceiver register value	OK or ERROR (attribute unknown or not accessible)
all	<*attr*>=<*value*>	Set transceiver register to value	OK or ERROR (attribute unknown, not accessible or value is out of range)
all	X*<*attr*>?	Show transceiver register value in hex	OK or ERROR (attribute unknown or not accessible)
all	X*<*attr*>=<*value*>	Set transceiver register to hexadecimal value	OK or ERROR (attribute unknown, not accessible or value is out of range)

* specify register numbers in hex

ATCW – Continuous Transmissions Test Modes

Syntax: ATCW<n>

Supported device types	Argument	Description	Result
all	0	off	-
all	1	PRBS mode 0 (random content)	OK or ERROR
all	2	PRBS mode 1 (fixed content 0x01)	OK or ERROR
all	3	PRBS mode 2 (fixed content 0xFE)	OK or ERROR
all	4	PRBS mode 3 (fixed content 0xAA)	OK or ERROR
all	5	CW mode 0 (below centre frequency, offset modulation dependent)	OK or ERROR
all	6	CW mode 1 (above centre frequency, offset modulation dependent)	OK or EROR

>Back to Overview<

A.N.Solutions

Am Brauhaus 5 • 01099 Dresden • Germany • **phone** +49 351 30900 199 • **fax** +49 351 30900 189 • info@an-solutions.de • **www.an-solutions.de**

AT+V – Access Persistent Module Memory

Note: A.N. Solutions offers a tool to read and write complete EEPROM files.

Syntax: AT+V(n)

Supported device types	Argument	Description	Result
all	1	Dumps content of the stored EEPROM in Intel Hex format partly. Only the part used by the SMS application is shown.	OK (for AT<*>(0)) or ERROR (otherwise)
all	2	Dumps all the EEPROM content in Intel Hex format.	OK (for AT<*>(0)) or ERROR (otherwise)
all	3	Writes one line to the EEPROM. For transferring a complete SMS EEPROM profile, use this command line by line.	OK (for AT<*>(0)) or ERROR (otherwise)
all	4	Reads module's SPI flash memory. Only with SMS Pro on the @ANY900 and @ANY2400 modules with integrated flash. Requires two characters length in hex and six characters starting address in hex.	OK (for AT<*>(0)) or ERROR (otherwise)
all	5	Writes module's SPI flash memory. Only with SMS Pro on the @ANY900 and @ANY2400 modules with integrated flash. Requires two characters length (n) in hex, six characters starting address in hex and n bytes payload data to write in hex. Note: address and (address + length) must not cross a sector boundary.	OK (for AT<*>(0)) or ERROR (otherwise)
all	6	Blocks erasure of module's SPI flash memory. Only with SMS Pro on the @ANY900 and @ANY2400 modules with integrated flash. Requires two characters block size in hex (only 0x40 supported) and six additional characters specifying an address within the sector to be deleted.	OK (for AT<*>(0)) or ERROR (otherwise)
all	7	Full erasure of module's SPI flash memory. This process takes about 6 seconds. Only with SMS Pro on the @ANY900 and @ANY2400 modules with integrated flash.	OK (for AT<*>(0)) or ERROR (otherwise)

>Back to Overview<

Example: AT+V

"AT+V1" ➔ partial EEPROM dump + "OK"
"AT+V2" ➔ full EEPROM dump + "OK"
"AT+V3:1000000001000000000000A0FFFFFFFFFFFFFFFF57" ➔ "OK"
"AT+V3:1000100000430D0A080000C00001A023E101000018" ➔ "OK"
"AT+V41F00000A" ➔ 31 Bytes Flash dump from address 0x00000A ➔ "OK"
"AT+V5050000101122334455" ➔ "OK"
"AT+V640000003" ➔ "OK"
"AT+V7" ➔ "OK"

AT+U – Jump into Bootloader Code

Note: A.N. Solutions offers a tool for simplified firmware updates.

Syntax: AT+U

Jumps to Bootloader Code to enable firmware updates via the stk500v1 protocol.
In order for the command to be executed, bit 5 in S register 94 must be set. This shall prevent accidental use of the bootloader. It should be noted that this bit is volatile, i.e. gets reset after the execution of the following AT command. This implies that the bit is not saved in the NVRAM. The AT+U command will not work, when the receiver is enabled. Use the ATH command to do so.

The supplied bootloader waits approximately 20 seconds for the programmer software on the PC side. When the association procedure times out, a reset is performed. As far as the programming software, we recommend using "avrdude", which is available for the operating systems Linux as well as Windows. A. N. Solutions is providing a firmware update tool for simplified firmware updates as well.

When using the command line-based way, the following steps need to be performed in order to upload a new firmware version:

- connect a terminal software like HTerm to the RS232 of the @ANY module
- type in the commands from the example below
- quit the terminal software and run the flash utility within 20 seconds
- with avrdude, flashing can be done as follows: "avrdude -p m1281 -c stk500v1 -P -P\\.\COM%1 -b 38400 -D -U fl:w:<firmware.hex>"
- when flashing has been completed successfully, open terminal software again and continue working

IMPORTANT NOTE: Please note that omitting the "-D" option from the avrdude command erases the whole flash including the bootloader, preventing further firmware updates without a JTAG programmer. This is not recommended.

Example: AT+U
"ATH" ➔ "OK"
"ATS94=224" ➔ "OK"
"AT+U" ➔ no OK, the 20-second timer was started

A.N. Solutions provides the At-ANY Updater Software, which facilitates the update process. It can be downloaded for free on our website.

>Back to Overview<

4 Automatic Command Line Processing

In order to organize periodic processing of commands, there is a user timer and a "shadow command buffer". This buffer can be set up using the 'AT*' prefix instead of 'AT'. If a command line starts with 'AT*', the line will be processed as usual and simultaneously stored in the shadow buffer.

Note that the command line will be stored as is even if it returns with an error indication. The stored command line is part of the device profile, so any 'AT&W(*n*)' command stores the actual buffer to NVRAM and 'ATZ(*n*)' reloads it to the active profile area. Processing of the stored command line can be started manually by using the 'AT-' command or user timer controlled using the S17 and S18 registers.

Whenever both S17 and S18 contain a value other than '0' the timer starts using these values and on expiring the stored command line will be processed automatically. The interval in 1/16 seconds is calculated by multiplication of the values of S17 and S18. In order to stop the timer, S17 or S18 can be reset to '0'. The application examples delivered with this document are intended to help to understand this better.

Example: AT*
"AT*DS245" ➔ "OK"
"ATS18=10" ➔ "OK"

AT*	Writes the following command line to the Shadow Command Register
D	Sends the following S-Register to the coordinator
245	S-Register
ATS18=10	Set the timer interrupt to 10 seconds

5 Watchdog Timer & MAC Failure Recovery

These two features were implemented in order to increase the stability of the SMS. The Watchdog timers time-out is fixed two seconds except during device association (command ATA on End devices) where it's set to 8 seconds and during network scans (command AT+S) where it's turned off.

The MAC failure recovery feature is used in Automatic command line processing. Whenever a MAC request returns with an error indication, a reset is performed similar to a WDT triggered reset. After reset the global device and network status is re-established. All necessary information (S-registers, network status information, coordinators associated device list) persists without being touched during the reset sequence.

Coordinators and FF-devices just return to their previous states, associated end-devices will re-associate. Register S11 keeps some information about these events: bits 0..2 are used as a WDT reset counter (counting up to 7 WDT events) and bits 4..7 are used as a counter for MAC failure recovery (counting up to 15 of these events). This counter information is cleared on any HW reset or ATZ command. It can be cleared manually using the command ATS11=0.

6 Asynchronous Message

The following asynchronous messages are defined in order to notify about network driven events.

Supported device types	Description	Result
End device	Device successfully associated, assigned short address is *addr*	+ASSOCIATED: *addr*
	Device successfully associated, assigned short address is *addr*	+NOT ASSOCIATED (*ec*) *
Coordinator	Device successfully associated, assigned short address is *addr*	+DEVICE ASSOCIATED: *addr* (*cap*) *
	Coordinator reports disassociation of device *addr*	+DISASSOCIATED: *addr*
all	Device received data	+DATA: *n* BYTES FROM *addr* (*ed*) *
	Data successfully sent	+SENT: ID *xxx*
	Data package *xxx* couldn't be sent for reason *ec* (see Appendix A)	+SEND FAILURE: ID *xxx* (*ec*) *
	Something (*ec*) happened (see Appendix A)	+COMM STATUS: *ec* + Src: *ieee address* + Dst: *ieee address*

* ec: error code, cap: capabilities, ed: energy detection

7 Routing Modes

SMS Pro supports different routing modes, which can be adjusted in S-Register S223. They will be described in this section.

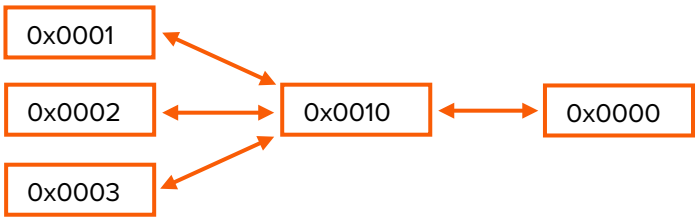
Routing is by default disabled, S223=0. Setting a different value enables one of the different implemented automated packets forwarding or routing schemes. Using this feature requires all nodes in the network to be FF Devices (S220=1) as a common prerequisite.

7.1 Forwarding Incoming Packets to One Predefined Node

This is the simplest automated packet forwarding mechanism and can be achieved by setting S223=1 or S223=2.

When S223=2, any packet incoming over the IEEE 802.15.4 RF link is re-transmitted over the RF link to the “stored target 0” entry. When S223=1, the incoming packets are re-transmitted and additionally printed on the local UART interface. For the mechanism of the “stored target 0”, please refer to the AT commands “AT&V1” and “AT&Z”.

A typical use case for such type of packet forwarding is collecting sensor data over multiple hops at a central point. It shall be noted, that the source node address is not preserved when the message is retransmitted, so it should be included in the message payload if required.



An example setup, limited to the bare minimum commands beside disabling the receiver, setting device role and PAN-ID can be as follows:

ATSX200=0003	ATSX200=0010	ATSX200=0000
ATA	ATS223=1	ATA
AT*D0010SA240	AT&Z0=0000	
ATS18=3	ATA	

7.2 Forwarding Incoming Packets Based on Source Address

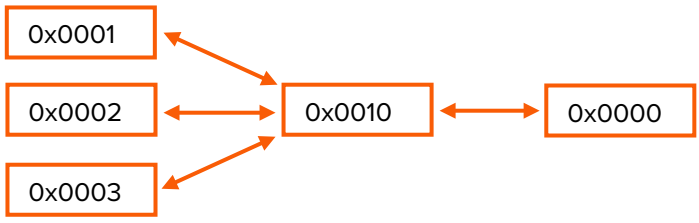
The previous subsection described the simplest case of forwarding data to one specific node, leading to a one-way data transmission in a tree-topology, only from the leaves to the root. This limitation is overcome by the S223=3 and S223=4 settings presented here, allowing to forward messages up and down a statically defined tree structure.

The available target addresses (in total 8) from the AT&V1 / AT&Z mechanism are used for re-transmitting incoming packets. AT&Z<n>=<target> is used to set a target address and AT&V1 to view them.

The first stored target address (AT&Z0=<target>) denotes the parent node in a tree topology. When this field holds the node's own address, it is assumed to be the root node of the tree. The remaining seven entries can be filled with children addresses.

When a routing-enabled device receives a packet from its parent node, S223=4 forwards it to all its children. The other way around, packets from other nodes are forwarded to the parent node. With the S register 223 = 3, forwarded packets can be tapped via the UART interface.

The static tree topology which can be formed has a width of up to 7. A graphical representation of a typical simple use case looks as follows:



An example setup, limited to the bare minimum commands beside disabling the receiver, setting device role and PAN-ID can be as follows.

ATSX200=0003	ATSX200=0010	ATSX200=0000
ATA	ATS223=3	ATA
AT*D0010SA240	AT&Z0=0000	AT*D0010SA240
ATS18=3	AT&Z1=0001	ATS18=10
	AT&Z2=0002	
	AT&Z3=0003	
	ATA	

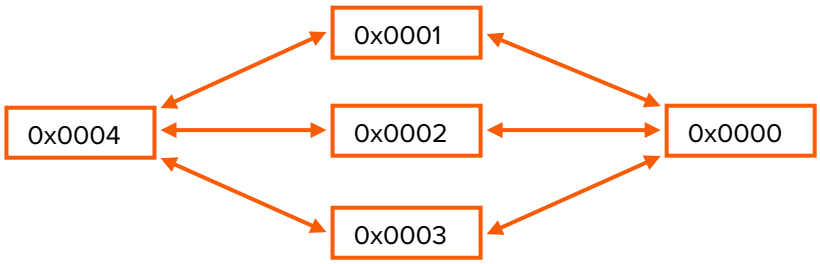
7.3 Mesh Routing

For applications where the approaches described above are not suitable, a full routing layer on top of the IEEE 802.15.4 standard is integrated in SMS Pro. It provides capabilities for establishing self-forming and self-healing mesh networks.

The routing algorithm itself is borrowed from the LwMesh stack. From the different options, “native routing” mechanism with multicast support has been selected and implemented with LwMesh default parameter settings. For details, please refer to the LwMesh stack and documentation.

When using a routing layer, all devices in the network shall have the same S223 setting. S223=5 enables dynamic routing with statically pre-assigned short addresses as a requirement.

A typical mesh structure looks like:



An example setup, limited to the bare minimum commands besides disabling the receiver, setting device role and PAN-ID can be as follows.

ATSX200=0004	ATSX200=0001	ATSX200=0000
ATS223=5	ATS223=5	ATS223=5
ATA	ATA	ATA
AT*D0000SA240		AT*D0004SA240
ATS18=3		ATS18=10

The advantage of the routing layer is that the transmitting entity has only to specify the destination address for the packet and has no need to define which way the data will travel through the network. This is done by the routing mechanism in the background, where LwMesh [4] dynamically reacts on changing environments. In the example structure above, the stack decides whether to send data from 0x0004 to 0x0000 via 0x0001, 0x0002 or 0x0003.

8 Registers

S<Reg.id>	Name	Size	Description	Notes	Default
S00	Auto mode	8 bit	If set to '1' an ATA is performed automatically after reset	Coordinator, FF device, End Device	0
S01	Device count	8 bit	Number of currently associate devices	Coordinator only, read only	0
S02	Escape character	8 bit	Character used for the ESC sequence (break)	compatibility	0x2B('+')
S03	CR character	8 bit	Line termination character	affects only input	0x0D
S04	LF character	8 bit	Output formatting character	affects only input	0x0A
S05	BS character	8 bit	Input controlling character		0x08
S06	UART baud rate low byte	8 bit	1/100th of the configured baud rate	baud rate=100*(256*S07+S06)	0x80
S07	UART baud rate high byte	8 bit	1/100th of the configured baud rate	baud rate=100*(256*S07+S06)	0x01
S09	UI_OPTIONS	8 bit	Define output format for incoming RF messages (+DATA)	Bit 7-3: reserved Bit 2: print out DSN with the data indication message Bit 1-0: type of signal strength indication →0: ED →1: RSSI →2: LQI →3: reserved	0
S10	Lost connection	8 bit		compatibility	0
S11	Spare	8 bit		used in debug mode	0
S12	ESC sequence guard time	8 bit		compatibility	0

S<Reg.id>	Name	Size	Description	Notes	Default
S14	Communication Options	8 bit	Holds CE, OE, RC, RCF and CM bits	Bits 5-7: Communication Mode (AT&Q) Bit 4: ResCode Format (ATV) Bit 2+3: Result Code (ATQ) Bit 1: Online Echo (ATF) Bit 0: Command Echo (ATE)	19
S17	Timer Prescaler	8 bit	Prescaler register for S18	in 1/16 second units see S18 and chapter 4.	16
S18	Timer	8 bit	Timer register	starts user timer if value > 0 depends on S17; see chapter 4.	0
S94	Mode selection	8 bit	Bit 7: enable WDT Bit 6: enable Reset after MAC errors Bit 5: unlock bootloader command AT+U Bit 4: reserved, always write 0 Bit 3: GPIO autosuspend	WDT interval is set to 2 sec. applies on automatic command line processing only Bit 5 will be reset after next command and cannot be saved	0xC0
S200	Short address	16 bit	see [1] for information about network parameters		x0000
S202	PAN Id	16 bit			0xFFFF
S204	Channel mask	32 bit		used on scan requests (AT+S)	
S208	Channel	8 bit			
S209	Channel page	8 bit			0
S210	Frame order	8 bit			
S211	Super frame order	8 bit			
S220	Device role	8 bit	0 – Coordinator 1 - FF device (router) 2 - End device	read only in active network modes, changing this leads to a MAC reset	1 or 2 (Base version)

S<Reg.id>	Name	Size	Description	Notes	Default
S221	Device capabilities	8 bit	By default coordinators send data to associated end devices indirectly. If this is set to 0x88 prior to association the *macRxOnWhenIdle* attribute will be set and data from coordinator will be sent directly.	End device only	0x80
S222	Scan duration	8 bit		used on scan	5
S223	Routing mode	8 bit	set packet redirection or routing mode SMS Pro only see AT&V1 and AT&Z to access stored targets		0

S223 register information:

Bit:	Description:
7	
6	
5	LwMesh Native routing with multicast support default settings only short addressing mode for use with S220=1 and unique SX200 for each dev.
4	same as 3, but no echo
3	source address equals stored target 0 ➜ forward to stored targets 1..7 source address differs from stored target 0 ➜ forward to stored target 0 echo in enabled
2	same as 1, but no echo
1	forward to stored target 0 and echo normally
0	disabled (manual forward)

S<Reg.id>	Name	Size	Description	Notes	Default
S224	Function mode	8 bit	enable extra layer for special functionalities SMS Pro only	0: disabled Bit 0: enable extra layer Bit 1: transparent UART I/O (TX to stored target 0)	0
S230	GPIO P0 in	8 bit	read GPIO input	current status of GPIO 0 .. 7	-
S231	GPIO P0 config	8 bit	GPIO0..7, 0 = IN, 1 = OUT each bit represents one pin	GPIO pin direction 0= IN, 1 = OUT	board dep.

S<Reg.id>	Name	Size	Description	Notes	Default
S232	GPIO P0 set	8 bit	set bit to set pin / pull-up	write 1 to all bits to set	board dep.
S233	GPIO P0 clr	8 bit	set bit to reset pin / pull-up	write 1 to all bits to clear	board dep.
S234	GPIO P0 toggle	8 bit	set bit to toggle	write 1 to all bits to toggle	-
S235	GPIO P1 in / status	8 bit	Bit 0, 1: read GPIO input Bit 2, 3: event history in order to reset event status write 1 to bit 2 / 3	2 interrupt capable input lines: read current status on bit 0, 1 bit 2, 3 are set if an event was triggered by this line (switch)	-
S236	GPIO P1 config	8 bit	Bit 0, 1: enable wake-up Bit 2, 3: enable command execution	trigger automatic command line processing	0

GPIO S-Register 236 information:

Bit:	Description:
7	
6	
5	
4	
3	Interrupt pin 7 Enable command execution
2	Interrupt pin 6 Enable command execution (Brick, Button S2)
1	Interrupt pin 7 Enable wake up
0	Interrupt pin 6 Enable wake up (Brick, Button S2)

S<Reg.id>	Name	Size	Description	Notes	Default
S237	GPIO P1 Debounce	8 bit	Debounce timeout Bit 7-4: debounce timeout for INT 7 Bit 3-0: debounce timeout for INT	Unit is multiples of 16 milliseconds. 0 means no debouncing on that line.	Board dep.
S238	Sensor read MUX	8 bit	Writing to this register triggers a sensor measurement with parameters from S239 and stores result in S240 to S256. Only used S-Registers are overwritten, others remain unchanged. See S239 and S240 for further details. Reading from this register returns the last value written	Board and version dependent, not all sensors might be supported. 0: clear S240 to S256 1: National Semicond. LM73 2: reserved 3: ATmega ADC one-shot (SMS Pro only) 4: Sensirion SHT11 5: Silicon Labs SI7006 / Sensirion SHT21 6: Sensirion STS21 7: BMA222E	0

S<Reg.id>	Name	Size	Description	Notes	Default
S239	Sensor read config	8 bit	Selects a measurement mode for use with S238.	Supported mode and value assignments are sensor dependent.	2

S239 register information:

Register S238:	Bit:	Description
4	7	Use low resolution
	6	Disable temperature measurement
	5	Disable humidity measurement
	4	No reload from OTP
	3	Verify CRCs (not implemented)
	2 - 0	Supply voltage: 0b000 = 2.5 V 0b001 = 3.0 V 0b010 = 3.5 V 0b011 = 4.0 V 0b100 = 5.0 V
5	7	Use low resolution
	6	Disable temperature measurement
	5	Disable humidity measurement
	4	No reload from OTP
	3	Verify CRCs (not implemented)
	2	-
	1	-
	0	-
6	7	Use low resolution
	6	Disable temperature measurement
	5	-
	4	No reload from OTP
	3	Verify CRCs (not implemented)
	2	-
	1	-
	0	-

S<Reg.id>	Name	Size	Description	Notes	Default
S240 to S256	User data	4 x 32 bit	Available through SX240, SX244, SX248 and SX252 as well as S240, S241, S242, ... Byte-order is swapped between different access variants.	When 238 is written, values are updated in format as follow:	0x00000000

S240 to S256 register information:

Register S238:	S<Reg.id>	Description:
0	240-256	All to 0
1	240	most significant byte of raw temperature value
	241	least significant byte of raw temperature value
	244	lowest 7 bits of raw sensor temperature value
	245	signed integer with temperature in degrees Celsius
2	240	most significant byte of temperature value
	241	least significant byte of temperature & 0xF0
3	240	ADCH of ADC0 pin
	241	ADCL of ADC0 pin
	242	ADCH of ADC1 pin
	243	ADCL of ADC1 pin
	244	ADCH of ADC2 pin
	245	ADCL of ADC2 pin
	246	ADCH of ADC3 pin
	247	ADCL of ADC3 pin
4	240	temperature value as unsigned integer with truncated fractional part
	241	rel. humidity value as unsigned integer with truncated fractional part
	242	dew point value as unsigned integer with truncated fractional part
	244 - 247	temperature value in IEEE754 32bit floating point format
	248 - 251	rel. humidity value in IEEE754 32bit floating point format
	252 -255	dew point value in IEEE754 32bit floating point format
5	240	temperature value as unsigned integer with truncated fractional part
	241	rel. humidity value as unsigned integer with truncated fractional part
	242	dew point value as unsigned integer with truncated fractional part
	244 - 247	temperature value in IEEE754 32bit floating point format
	248 - 251	rel. humidity value in IEEE754 32bit floating point format
	252 -255	dew point value in IEEE754 32bit floating point format

9 Appendix A – MAC Enumerations Description

Enumeration	Value	Description
SUCCESS	0x00	The requested operation was completed successfully. For a transmission request, this value indicates a successful transmission.
—	0x01 – 0xDA	Reserved for MAC command status and reason code values.
—	0x80 – 0xDA 0xFE – 0xFF	Reserved.
BEACON_LOSS	0xE0	The beacon was lost following a synchronization request.
CHANNEL_ACCESS_FAILURE	0xE1	A transmission could not take place due to activity on the channel, i.e., the CSMA-CA mechanism has failed.
COUNTER_ERROR	0xDB	The frame counter purportedly applied by the originator of the received frame is invalid.
DENIED	0xE2	The GTS request has been denied by the PAN coordinator.
DISABLE_TRX_FAILURE	0xE3	The attempt to disable the transceiver has failed.
FRAME_TOO_LONG	0xE5	Either a frame resulting from processing has a length that is greater than *aMaxPHYPacketSize* or a requested transaction is too large to fit in the CAP or GTS.
IMPROPER_KEY_TYPE	0xDC	The key purportedly applied by the originator of the received frame is not allowed to be used with that frame type according to the key usage policy of the recipient.
IMPROPER_SECURITY_LEVEL	0xDD	The security level purportedly applied by the originator of the received frame does not meet the minimum-security level required/expected by the recipient for that frame type.
INVALID_ADDRESS	0xF5	A request to send data was unsuccessful because neither the source address parameters nor the destination address parameters were present.
INVALID_GTS	0xE6	The requested GTS transmission failed because the specified GTS either did not have a transmit GTS direction or was not defined.
INVALID_HANDLE	0xE7	A request to purge an MSDU from the transaction queue was made using an MSDU handle that was not found in the transaction table.
INVALID_INDEX	0xF9	An attempt to write to a MAC PIB attribute that is in a table failed because the specified table index was out of range.
INVALID_PARAMETER	0xE8	A parameter in the primitive is either not supported or is out of the valid range.

Enumeration	Value	Description
LIMIT_REACHED	0xFA	A scan operation terminated prematurely because the number of PAN descriptors stored reached an implementation specified maximum.
NO_ACK	0xE9	No acknowledgment was received after *macMaxFrameRetries*.
NO_BEACON	0xEA	A scan operation failed to find any network beacons.
NO_DATA	0xEB	No response data were available following a request.
NO_SHORT_ADDRESS	0xEC	The operation failed because a 16-bit short address was not allocated.
ON_TIME_TOO_LONG	0xF6	A receiver enable request was unsuccessful because it specified a number of symbols that was longer than the beacon interval.
OUT_OF_CAP	0xED	A receiver enable request was unsuccessful because it could not be completed within the CAP. The enumeration description is not used in this standard, and it is included only to meet the backwards compatibility requirements for IEEE Std 802.15.4-2003.
PAN_ID_CONFLICT	0xEE	A PAN identifier conflict has been detected and communicated to the PAN coordinator.
PAST_TIME	0xF7	A receiver enable request was unsuccessful because it could not be completed within the current super frame and was not permitted to be deferred until the next super frame.
READ_ONLY	0xFB	A SET/GET request was issued with the identifier of an attribute that is read only.
REALIGNMENT	0xEF	A coordinator realignment command has been received.
SCAN_IN_PROGRESS	0xFC	A request to perform a scan operation failed because the MLME was in the process of performing a previously initiated scan operation.
SECURITY_ERROR	0xE4	Cryptographic processing of the received secured frame failed.
SUPERFRAME_OVERLAP	0xFD	The device was instructed to start sending beacons based on the timing of the beacon transmissions of its coordinator, but the instructed start time overlapped the transmission time of the beacon of its coordinator.
TRACKING_OFF	0xF8	The device was instructed to start sending beacons based on the timing of the beacon transmissions of its coordinator, but the device is not currently tracking the beacon of its coordinator.
TRANSACTION_EXPIRED	0xF0	The transaction has expired and its information was discarded.
TRANSACTION_OVERFLOW	0xF1	There is no capacity to store the transaction.

Enumeration	Value	Description
TX_ACTIVE	0xF2	The transceiver was in the transmitter enabled state when the receiver was requested to be enabled. The enumeration description is not used in this standard, and it is included only to meet the backwards compatibility requirements for IEEE Std 802.15.4-2003.
UNAVAILABLE_KEY	0xF3	The key purportedly used by the originator of the received frame is not available or, if available, the originating device is not known or is blacklisted with that particular key.
UNSUPPORTED_ATTRIBUTE	0xF4	A SET/GET request was issued with the identifier of a PIB attribute that is not supported.
UNSUPPORTED_LEGACY	0xDE	The received frame was purportedly secured using security based on IEEE Std 802.15.4-2003, and such security is not supported by this standard.
UNSUPPORTED_SECURITY	0xDF	The security purportedly applied by the originator of the received frame is not supported.

10 Appendix B – PHY PIB Attributes

Attribute	Identifier	Type	Range	Description
phyCurrentChannel	0x00	Integer	0 – 26	The RF channel to use for all following transmissions and receptions (see 6.1.2).
phyChannelsSupported	0x01	Array	An R x 32 bit array, where ranges from 1 to 32	The array is composed of R rows, each of which is a bit string with the following properties: The 5 MSBs (b27, ..., b31) indicate the channel page, and the 27 LSBs (b0, b1, ..., b26) indicate the status (1=available, 0=unavailable) for each of the up to 27 valid channels (bk shall indicate the status of channel k as in 6.1.2) supported by that channel page. The device only needs to add the rows (channel pages) for the PHY(s) it supports.
phyTransmitPower	0x02	Bitmap	0x00 – 0xBF	The 2 MSBs represent the tolerance on the transmit power: 00 = ± 1 dB 01 = ± 3 dB 10 = ± 6 dB and shall be read-only. The 6 LSBs, which may be written to, represent a signed integer in twos-complement format, corresponding to the nominal transmit power of the device in decibels relative to 1 mW. The lowest value of *phyTransmitPower* is interpreted as less than or equal to –32 dBm.
phyCCAMode	0x03	Integer	1 – 3	The CCA mode (see 6.9.9).

Attribute	Identifier	Type	Range	Description
phyCurrentPage	0x04	Integer	0 – 31	This is the current PHY channel page. This is used in conjunction with *phyCurrentChannel* to uniquely identify the channel currently being used.
phyMaxFrameDuration	0x05	Integer	55, 212, 266, 1064	The maximum number of symbols in a frame: = *phySHRDuration* + ceiling ([*aMaxPHYPacketSize* + 1] x *phySymbolsPerOctet*)
phySHRDuration	0x06	Integer	3, 7, 10, 40	The duration of the synchronization header (SHR) in symbols for the current PHY.
phySymbolsPerOctet	0x07	Float	0.4, 1.6, 2, 8	The number of symbols per octet for the current PHY.

11 Appendix C – MAC PIB Attributes

Attribute	Identifier	Type	Range	Description	Default
macAckWaitDuration	0x40	Integer	see [1]	The maximum number of symbols to wait for an acknowledgment frame to arrive following a transmitted data frame. This value is dependent on the supported PHY, which determines both the selected logical channel and channel page. The calculated value is the time to commence transmitting the ACK plus the length of the ACK frame. The commencement time is described in 7.5.6.4.2.	Dependent on currently selected PHY, indicated by *phyCurrentPage*
macAssociatedPAN-Coord	0x56	Boolean	TRUE or FALSE	Indication of whether the device is associated to the PAN through the PAN coordinator. A value of TRUE indicates the device has associated through the PAN coordinator. Otherwise, the value is set to FALSE.	FALSE
macAssociationPermit	0x41	Boolean	TRUE or FALSE	Indication of whether a coordinator is currently allowing association. A value of TRUE indicates that association is permitted.	FALSE

Attribute	Identifier	Type	Range	Description	Default
macAutoRequest	0x42	Boolean	TRUE or FALSE	Indication of whether a device automatically sends a data request command if its address is listed in the beacon frame. A value of TRUE indicates that the data request command is automatically sent. This attribute also affects the generation of the MLMEBEACONNOTIFY. indication primitive (see 7.1.5.1.2).	TRUE
macBattLifeExt	0x43	Boolean	TRUE or FALSE	Indication of whether BLE, through the reduction of coordinator receiver operation time during the CAP, is enabled. A value of TRUE indicates that it is enabled. Also, see 7.5.1.4 for an explanation of how this attribute affects the back off exponent in the CSMA-CA algorithm.	FALSE
macBattLifeExtPeriods	0x44	Integer	6 – 41	In BLE mode, the number of back off periods during which the receiver is enabled after the IFS following a beacon. This value is dependent on the supported PHY and is the sum of three terms: Term 1: The value, where x is the maximum value of *macMinBE* in BLE mode (equal to two). This term is thus equal to 3 back off periods. Term 2: The duration of the initial contention window length (see 7.5.1.4). This term is thus equal to 2 back off periods. Term 3: The Preamble field length and the SFD field length of the supported PHY (see Table 19 and Table 20 in Clause 6), summed together and rounded up (if necessary) to an integer number of back off periods.	Dependent on currently selected PHY, indicated by *phyCurrentPage*
macBeaconPayload	0x45	Set of octets	—	The contents of the beacon payload.	NULL

Attribute	Identifier	Type	Range	Description	Default
macBeaconPayload-Length	0x46	Integer	0 – *aMaxBeacon-PayloadLength*	The length, in octets, of the beacon payload.	0
macBeaconOrder	0x47	Integer	0 – 15	Specification of how often the coordinator transmits its beacon. If B O = 15, the coordinator will not transmit a periodic beacon. See 7.5.1.1 for an explanation of the relationship between the beacon order and the beacon interval.	15
macBeaconTxTime	0x48	Integer	0x000000 – 0xFFFFFF	The time that the device transmitted its last beacon frame, in symbol periods. The measurement shall be taken at the same symbol boundary within every transmitted beacon frame, the location of which is implementation specific. This is a 24-bit value, and the precision of this value shall be a minimum of 20 bits, with the lowest four bits being the least significant.	0x000000
macBSN	0x49	Integer	0x00 – 0xFF	The sequence number added to the transmitted beacon frame.	Random value from within the range
macCoordExtended-Address	0x4A	IEEE address	An extended 64-bit IEEE address	The 64-bit address of the coordinator through which the device is associated.	—
macCoordShort-Address	0x4B	Integer	0x0000 – 0xFFFF	The 16-bit short address assigned to the coordinator through which the device is associated. A value of 0xfffe indicates that the coordinator is only using its 64-bit extended address. A value of 0xFFFF indicates that this value is unknown.	0xFFFF
macDSN	0x4C	Integer	0x00 – 0xFF	The sequence number added to the transmitted data or MAC command frame.	Random value from within the range

Attribute	Identifier	Type	Range	Description	Default
macGTSPermit	0x4D	Boolean	TRUE or FALSE	TRUE if the PAN coordinator is to accept GTS requests. FALSE otherwise.	TRUE
macMaxBE	0x57	Integer	3 – 8	The maximum value of the back off exponent, BE, in the CSMA-CA algorithm. See 7.5.1.4 for a detailed explanation of the back off exponent.	5
macMaxCSMABackoffs	0x4E	Integer	0 – 5	The maximum number of back offs the CSMA-CA algorithm will attempt before declaring a channel access failure.	4
macMaxFrameTotal-WaitTime	0x58	Integer	see [1]	The maximum number of CAP symbols in a beacon enabled PAN, or symbols in a no beacon-enabled PAN, to wait either for a frame intended as a response to a data request frame or for a broadcast frame following a beacon with the Frame Pending subfield set to one. This attribute, which shall only be set by the next higher layer, is dependent upon *macMinBE*, *macMaxBE*, *macMaxCSMABackoffs* and the number of symbols per octet. See 7.4.2 for the formula relating the attributes.	Dependent on currently selected PHY, indicated by *phyCurrentPage*
macMaxFrameRetries	0x59	Integer	0 – 7	The maximum number of retries allowed after a transmission failure.	3
macMinBE	0x4F	Integer	0 – *macMaxBE*	The minimum value of the back off exponent (BE) in the CSMA-CA algorithm. See 7.5.1.4 for a detailed explanation of the back off exponent.	
macMinLIFSPeriod		Integer	see [1]	The minimum number of symbols forming a LIFS period.	Dependent on currently selected PHY, indicated by *phyCurrentPage*
macMinSIFSPeriod		Integer	see [1]	The minimum number of symbols forming a SIFS period.	Dependent on currently selected PHY,

Attribute	Identifier	Type	Range	Description	Default
					indicated by *phyCurrentPage*
macPANId	0x50	Integer	0x0000 – 0xFFFF	The 16-bit identifier of the PAN on which the device is operating. If this value is 0xFFFF, the device is not associated.	0xFFFF
macPromiscuous-Mode	0x51	Boolean	TRUE or FALSE	Indication of whether the MAC sublayer is in a promiscuous (receive all) mode. A value of TRUE indicates that the MAC sublayer accepts all frames received from the PHY.	FALSE
macResponseWaitTime	0x5A	Integer	2 – 64	The maximum time, in multiples of *aBaseSuperframeDuration*, a device shall wait for a response command frame to be available following a request command frame.	32
macRxOnWhenIdle	0x52	Boolean	TRUE or FALSE	Indication of whether the MAC sublayer is to enable its receiver during idle periods. For a beacon- enabled PAN, this attribute is relevant only during the CAP of the incoming super frame. For a no beacon-enabled PAN, this attribute is relevant at all times.	FALSE
macSecurityEnabled	0x5D	Boolean	TRUE or FALSE	Indication of whether the MAC sublayer has security enabled. A value of TRUE indicates that security is enabled, while a value of FALSE indicates that security is disabled.	FALSE
macShortAddress	0x53	Integer	0x0000 – 0xFFFF	The 16-bit address that the device uses to communicate in the PAN. If the device is the PAN coordinator, this value shall be chosen before a PAN is started. Otherwise, the address is allocated by a coordinator during association. A value of 0xfffe indicates that the device has associated but has not been allocated	0xFFFF

Attribute	Identifier	Type	Range	Description	Default
				an address. A value of 0xFFFF indicates that the device does not have a short address.	
macSuperframe- Order	0x54	Integer	0 - 15	The length of the active portion of the outgoing super frame, including the beacon frame. If super frame order, SO, = 15, the super frame will not be active following the beacon. See 7.5.1.1 for an explanation of the relationship between the super frame order and the super frame duration.	Implementation specific
macSyncSymbolOffset	0x5B	Integer	0x000-0x100 for the 2.4 GHz PHY 0x000-0x400 for the 868/915 MHz PHY	The offset, measured in symbols, between the symbol boundary at which the MLME captures the timestamp of each transmitted or received frame, and the onset of the first symbol past the SFD, namely, the first symbol of the Length field.	Implementation specific
macTimestamp-Supported	0x5C	Boolean	TRUE or FALSE	Indication of whether the MAC sublayer supports the optional timestamping feature for incoming and outgoing data frames.	
macTransaction-PersistenceTime	0x55	Integer	0x0000 – 0xFFFF	The maximum time (in unit periods) that a transaction is stored by a coordinator and indicated in its beacon. The unit period is governed by *macBeaconOrder,* BO, as follows: For $0 \leq BO \leq 14$, the unit period will be a Base-*Super frame Duration* * 2BO. For BO = 15, the unit period will be *a Base-Super frame Duration.*	0x01F4

12 Appendix D – Device Initialization

In order to perform the following steps, an AVR programmer (JTAG or ISP – we recommend Microchips Atmel ICE , see https://www.microchip.com/DevelopmentTools/ProductDetails/ATATMEL-ICE) as well as some supporting software such as "Atmel Studio" or "avrdude" is needed.

12.1 Program Device Firmware and Set ATMEGA1281 & ATMEGA128RFA1 Fuses

The firmware file to program the device flash is named "SMS_ATANY<ISM><BOARD>[_BL]_<VERSION>.hex".*

Make sure to use the following fuse settings on boards based on ATmega1281: LOW: 0xE2, HIGH: 0x91, EXTENDED: 0xFE.

On Atmega128RFA1 based boards, the following fuse settings should be used: LOW: 0xC6, HIGH: 0x99, EXTENDED: 0xFF

Different fuse settings can be used as well, but might have impact on functionality and power consumption.

- SMS_ATANY09BRICK_147p_xxxx.hex : Smart MAC Suite V-1.47-Pro for @ANY900 Brick
- SMS_ATANY24BRICK_147p_xxxx.hex : Smart MAC Suite V-1.47-Pro for @ANY2400 Brick
- SMS_ATANY09DONGLE_147p_xxxx.hex : Smart MAC Suite V-1.47-Pro for @ANY900 Dongle
- SMS_ATANY24DONGLE_147p_xxxx.hex : Smart MAC Suite V-1.47-Pro for @ANY2400 Dongle

The optional part "_BL" included in the filename stands for the bootloader code included in the hex file. The use of those files is recommended when programming firmware using the JTAG interface. When updating the firmware through the serial interface via bootloader, it is recommended to use the appropriate firmware files without "_BL" in the filename.

12.2 Set IEEE Address

Using a serial terminal

If the device is turned on without a valid (any value except 0 or -1=0xFFFFFFFFFFFFFFFF) IEEE address in EEPROM it sets up a temporary address, which is randomly generated.

This address can be changed using the AT+CF0 and AT&Waddr commands:

AT+CF0=<*xxxxxxxxxxxxxxxx*> OK

AT&Waddr OK

Using an EEPROM data file

The IEEE address is stored in the first eight bytes of the EEPROM.

In order to create a data file, create a binary file containing the device address and use avr-objcopy as follows:

> avr-objcopy -I binary -O ihex *address*.bin *address*.eep

The output-file can now be programmed to the device EEPROM.

A tool to create EEPROM profiles (including the IEEE address) named "SMS Profiler" is available. For information how to read and write EEPROM profiles from/to the device, please refer to the AT+V command or use the tools provided additionally by A.N. Solutions to ease the process.

Reference Documents

[1] – IEEE Std 802.15.4™-2006 (Revision of IEEE Std 802.15.4-2003):

Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)

[2] – ITU-T V.250: Serial asynchronous automatic dialling and control

[3] – @ANY900 / @ANY2400 Module Product data sheets (https://www.an-solutions.de/de/produkte.html)

[4] – Atmel Lightweight Mesh
https://www.microchip.com/DevelopmentTools/ProductDetails/PartNO/Atmel%20Lightweight%20Mesh

Revision History

Revision	Changes	Date
3.2	Dokumentation update	2020/10/02

Disclaimer

A.N. Solutions believes that all information is correct and accurate at the time of issue. A.N. Solutions reserves the right to make changes to this product without prior notice. Please visit A.N. Solutions website for the latest available version.

A.N. Solutions does not assume any responsibility for the use of the described product or convey any license under its patent rights.

A.N. Solutions warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with A.N. Solutions standard warranty. Testing and other quality control techniques are used to the extent A.N. Solutions deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

Trademarks

AT-ANY, @ANY and related naming as well as A.N. Solutions logo are trademarks of A.N. Solutions GmbH. All other product names, trade names, trademarks, logos or service names are the property of their respective owners.

Technical Support

Technical support is provided by A.N. Solutions GmbH on demand and in accordance to service conditions agreed.
E-mail: support@an-solutions.de
Please refer to Support Terms and Conditions for full details.

Contact Information

A.N. Solutions GmbH

Am Brauhaus 5
01099 Dresden
Germany

Tel: +49 351 - 30 900 199
Fax: +49 351 - 30 900 189

Office hours: 8:00am - 5:00pm (Central European Time)