

# Getting Started

@ANY Design Development Kit Sub-1 GHz / 2.4 GHz  
IEEE 802.15.4 Standard



10 July 2020

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



## Inhaltsverzeichnis

Abbildungsverzeichnis	3
Tabellenverzeichnis	3
1 Übersicht	4
1.1 Übersicht Development Kits	6
2 Verbinden der einzelnen Geräte mit dem PC	7
3 Erster Schritt: A.N. Solutions ToolChain	8
3.1 A.N. Solutions ToolChain – SMS Monitor	8
3.2 Späteres Verwenden des SMS Monitors	10
4 Getting Started	11
4.1 Navigieren in der SMS	11
4.2 Alle Geräte auf Werkseinstellungen zurücksetzen	11
4.3 Informationen ausgeben lassen	12
4.4 Profile anzeigen, bearbeiten, speichern	13
4.5 Konfigurieren der SMS per Console	15
4.5.1 Konfigurieren des @ANY USB Dongle als Koordinator	15
4.5.2 Setzen der IEEE Adresse	16
4.5.3 Konfigurieren des @ANY Brick als END-Gerät	17
4.6 Aufbau eines Wireless Sensor Network (WSN)	17
4.6.1 Peer to Peer Netzwerk	17
4.6.2 Stern Netzwerk	19
4.7 Mesh Netzwerk für SMS Pro Nutzer	21
4.7.1 Einrichten der Netzwerkteilnehmer	21
4.7.2 Einrichten und Inbetriebnahme der Testumgebung	23
4.8 ADC Unterstützung für SMS Pro Nutzer	25
4.8.1 Grundeinstellungen des ADCs mit der SMS Pro	25
4.8.2 Beispiel Batteriemessung mit dem ADC	26
5 Firmware flashen/updaten	27
5.1 Struktur der @ANY Hardware	27
5.2 JTAG	29
5.3 Bootloader	29
5.4 Firmware flashen/updaten mit dem SMS Updater	29
5.5 Firmware flashen/updaten über AVRDUDE ohne JTAG/SWD	30
5.6 Firmware flashen/updaten über AVRDUDE mit JTAG/SWD	31
5.7 Firmware flashen/updaten mit JTAG/SWD & Atmel Studio	33



6	Software Tools	34
6.1	Sniffer Tool	34
6.2	Transparent UART – SMS PRO Nutzer	36
	Referenz Dokumente	38
	Änderungsübersicht	38
	Disclaimer	39

## Abbildungsverzeichnis

Figure 1	Produkt Overview	5
Figure 2	Device Manager	7
Figure 3	Download Center	8
Figure 4	Tools	8
Figure 5	A.N. Solutions ToolChain	9
Figure 6	SMS Monitor	10
Figure 7	AT Command	11
Figure 8	ATI long	12
Figure 9	ATI small	12
Figure 10	Part one profile	13
Figure 11	Part two profile	13
Figure 12	Full profile overview	14
Figure 13	Full config dongle	16
Figure 14	Peer to Peer Network	18
Figure 15	Star - Network	19
Figure 16	Mesh peer to peer**	23
Figure 17	Dynamic Mesh**	24
Figure 18	Section @ANY BRICK X3	25
Figure 19	Circuit ADC0	26
Figure 20	@ANY900 Modules	27
Figure 21	@ANY900ARM-SC Modules	28
Figure 22	@ANY2400-SC Modules	28
Figure 23	@ANY2400-SC-3 Modules	28
Figure 24	Atmel ICE Kit	29
Figure 25	ATANY-Updater	30
Figure 26	Tag Connect	32
Figure 27	Device programming head	33
Figure 28	Memories - Flash	33
Figure 29	Sniffer config	34

## Tabellenverzeichnis

Table 1	Overview Development Kits	6
Table 2	ATI Command	12
Table 3	Config BRICK	17



# 1 Übersicht

Wir freuen uns, dass Sie sich für das @ANY DESIGN/Development Kit (DevKit) von A.N. Solutions entschieden haben. Das @ANY DESIGN Development Kit bietet Ihnen die Möglichkeit mit Hilfe der bereitgestellten Tools ein drahtloses IEEE 802.15.4 / Zigbee Netzwerk einzurichten, auf Grundlage unserer branchenführenden Modulplattformen. Auf einfache Weise erfahren Sie, wie Sie die @ANY RF-Module verwenden und einzigartige drahtlose Netzwerk-anwendungen für die Erfassung von Sensordaten und die Steuerung dieser entwickeln. Entwickler können verschiedene Netzwerktopologien ausprobieren, um festzustellen, welche für die Anwendung am besten geeignetste ist. Sensoren, die in die BRICK-Platine integriert sind, vereinfachen das Prototypen eines drahtlosen Sensornetzwerkes.

Die Smart MAC Suite (SMS) ist eine auf dem IEEE 802.15.4 Standard basierende Firmware, welche mit AT-Kommandos oder durch beschreiben des EEPROM konfiguriert werden kann. Dies ermöglicht dem Nutzer das Einrichten drahtloser Netzwerke, ohne sich mit schwierigen Programmiersprachen auseinander setzen zu müssen.

Es stehen zwei Firmwarepakete zu Verfügung:

- Smart MAC Suite Base
- Smart MAC Suite Pro

Im Lieferumfang ist Standardmäßig die SMS Base enthalten. Das erworbene Firmware-Paket befindet sich fertig geflasht auf der Hardware und ist vorkonfiguriert mit dem Beispiel SMS Monitor. Die SMS Pro kann gegen eine einmalige Lizenz auf unserer Webseite [www.an-solutions.de](http://www.an-solutions.de) erworben werden. Haben Sie das DevKit mit der SMS Pro erworben, so befindet sich die Firmware SMS Pro schon auf den Hardwarekomponenten. Haben Sie nachträglich die SMS Pro erworben, muss diese zuerst auf die Hardwarekomponenten geflasht werden. Dazu springen Sie in das Kapitel 5.

Bevor Sie starten können, überprüfen Sie bitte den Verpackungsinhalt auf Vollständigkeit. Außerdem wird ein Kommandozeilen Tool benötigt, welche eine serielle Verbindung zwischen Ihrem PC und den Hardwarekomponenten ermöglicht. Wir empfehlen das kostenlose Tool [Hterm](#).

Dieses Dokument beschreibt die grundlegenden Schritte bei der Verwendung der SMS-Software und das Einrichten einfacher Netzwerke zwischen mehreren Teilnehmern. Das Getting Started ist Firmware unabhängig. Für komplexere Anwendungen lesen Sie bitte das Dokument „SMS Command Reference“. Die hier aufgeführten Befehle dienen dem Einstieg in die SMS. Es werden somit nicht alle Befehle, Möglichkeiten und Argumente vorgestellt. Das Dokument “SMS Command Reference“ steht kostenlos auf unserer Webseite zur Verfügung. [www.an-solutions.de](http://www.an-solutions.de)

Dieses Getting Started beschreibt die Einrichtung in Microsoft Windows ab der Version 8 und wurde mit der Version Windows 10.0.18362 getestet. Die Einrichtung unter Linux verhält sich etwas anders.

Kapitel 5 ist optional. Wird es dennoch behandelt, muss im Vorfeld Zusatz Hard- sowie Software erworben werden!

- ATATMEL-ICE-BASIC - Atmel-ICE Basic Kit
- Atmel Studio 7 (Stand 2019)
- JTAG Connector für den @ANY 2400-SC-Pro USB Dongle - [Tag Connect TC2050-IDC-NL-050](#)
- JTAG Connector für den @ANY900ARM-SC USB Dongle - [Tag Connect TC2030-CTX-NL 6-PIN](#)

Das Atmel-ICE Basic Kit kann kostenpflichtig bei Microchip erworben werden. [www.microchipdirect.com](http://www.microchipdirect.com) Unter anderem wird Atmel Studio benötigt. Atmel Studio kann bei Microchip kostenlos heruntergeladen werden. [www.microchip.com](http://www.microchip.com)



Zu dem Flaschen der @ANY USB Dongels über JTAG, wird ein Adapterstecker benötigt wie der [TC2050-IDC-NL-050](#) oder der [TC2030-CTX-NL 6-Pin](#). Die @ANY USB Dongle besitzen keinen 6/10 Pin 50 mil male Stecker auf dem PCB!

Das Getting Started wurde mit dem AT-ANY-DESIGN-2400-SC-3-2 DevKit erstellt. Dieses DevKit arbeitet mit dem 2.4 GHz Frequenzband. Alle Einstellungen in diesem Dokument beziehen sich auf dieses Frequenzband. Für das 1 GHz Frequenzband müssen Einstellungen an bestimmten Stellen abgeändert werden, diese sind gekennzeichnet.

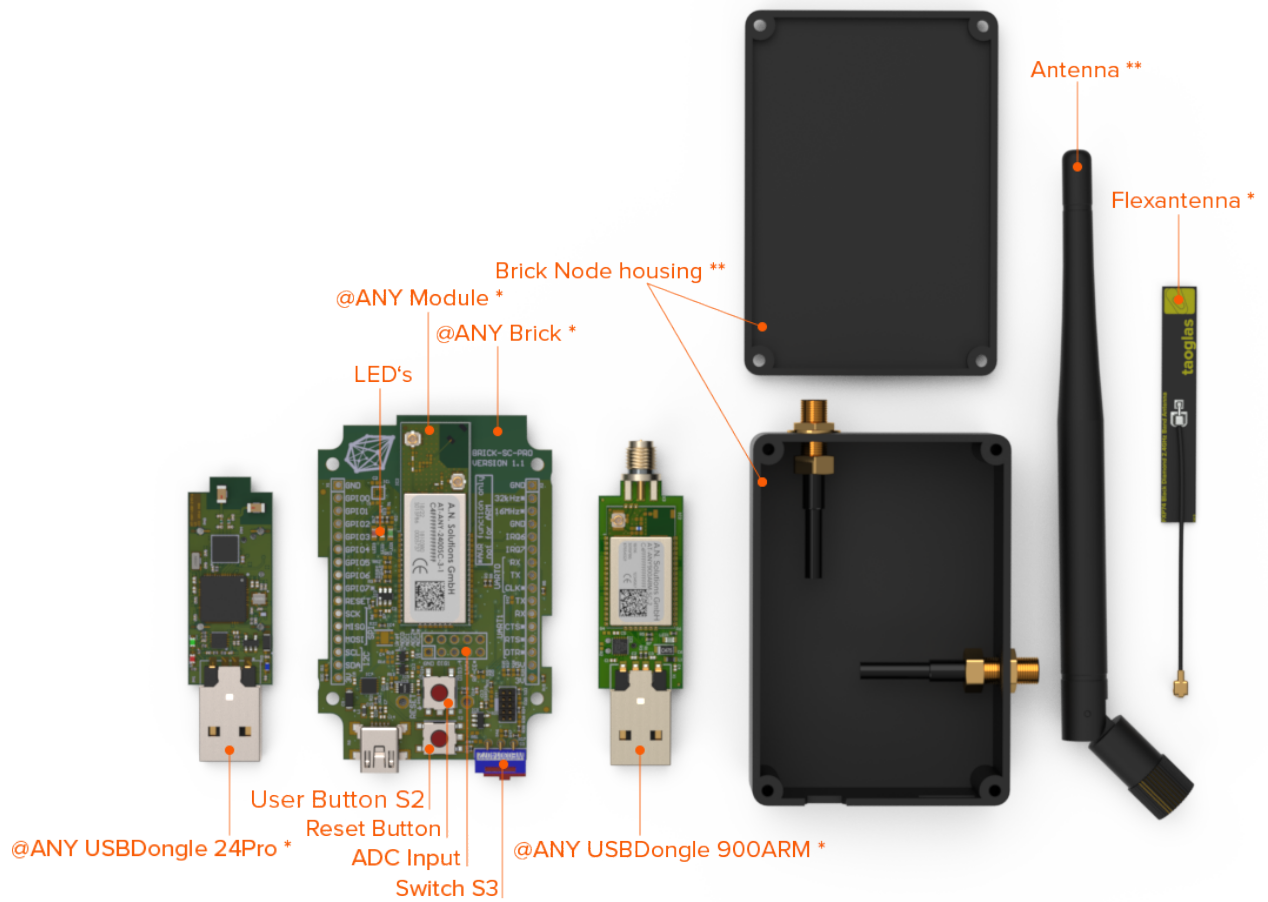


Figure 1 Produkt Overview

\* Development Kit abhängig  
 \*\* optional

## 1.1 Übersicht Development Kits

BRICK Part.-No.:	Sensors & Features on @ANY BRICK boards						Supported Features by @ANY Module			
	LM73 Temp. Sensor	S17006 Humidity & Temp. Sensor	BMA222E Acceleration Sensor	Configurable LEDs	User Button	Reset Button	U.FL Connector	Integrated Antenna	On-Board Data-Flash	
AT-ANY2400-SC-1 BRICK	yes	yes	yes	3	1	1	-	yes	-	
AT-ANY2400-SC-2 BRICK	yes	yes	yes	3	1	1	yes	-	-	
AT-ANY2400-SC-3-1 BRICK	yes	-	-	3	1	1	yes	yes	2 Mbit SST25VF0202	
AT-ANY2400-SC-3-2 BRICK	yes	-	-	3	1	1	2	-	2 Mbit SST25VF0202	
AT-ANY900-1 BRICK	yes	yes	yes	3	1	1	-	yes	2 Mbit SST25VF0202	
AT-ANY900-2 BRICK	yes	yes	yes	3	1	1	yes	-	2 Mbit SST25VF0202	
AT-ANY900ARM-SC BRICK	yes	yes	yes	3	1	1	yes	-	4 Mbit AT25XE04	
Used @ANY USB Dongles:										
AT-ANY2400-SC-Pro USB	-	-	-	2	-	-	-	2	-	
AT-ANY900ARM-SC USB	-	-	-	1	-	-	(SMA)	-	4 Mbit AT25XE04	

Table 1 Overview Development Kits

## 2 Verbinden der einzelnen Gräte mit dem PC

Um die SMS konfigurieren zu können, muss eine serielle Verbindung zwischen der Hardware und Ihrem PC aufgebaut werden. Dazu stecken Sie den @ANY USB Dongle in einen freien USB-Port an Ihren PC. Die @ANY BRICKs werden über ein USB mini Kabel mit dem PC verbunden. Die grüne LED beginnt zu leuchten (Standard Firmware Einstellungen). Ab Windows 8 übernimmt der Hardware-Einrichtungs-Assistent die Installation der benötigten Treiber für FT230X und UART.

Nach der Installation navigieren Sie in den Geräte-Manager, mit der Tastenkombination: „Windows-Logo-Taste + Pause“.

Wählen sie dort den Geräte-Manager aus. Unter dem Punkt „Anschlüsse (COM & LPZ)“ wird ein weiterer USB Serial Port (COM <n>) angezeigt.

Ist das nicht der Fall, muss der Treiber manuell installiert werden. Dazu surfen Sie die Webseite von FTDI an [www.ftdichip.com](http://www.ftdichip.com), navigieren am linken Webseitenrad zu „Drivers“ und laden sich dort den aktuellsten VCP Treiber für Ihr Betriebssystem herunter.

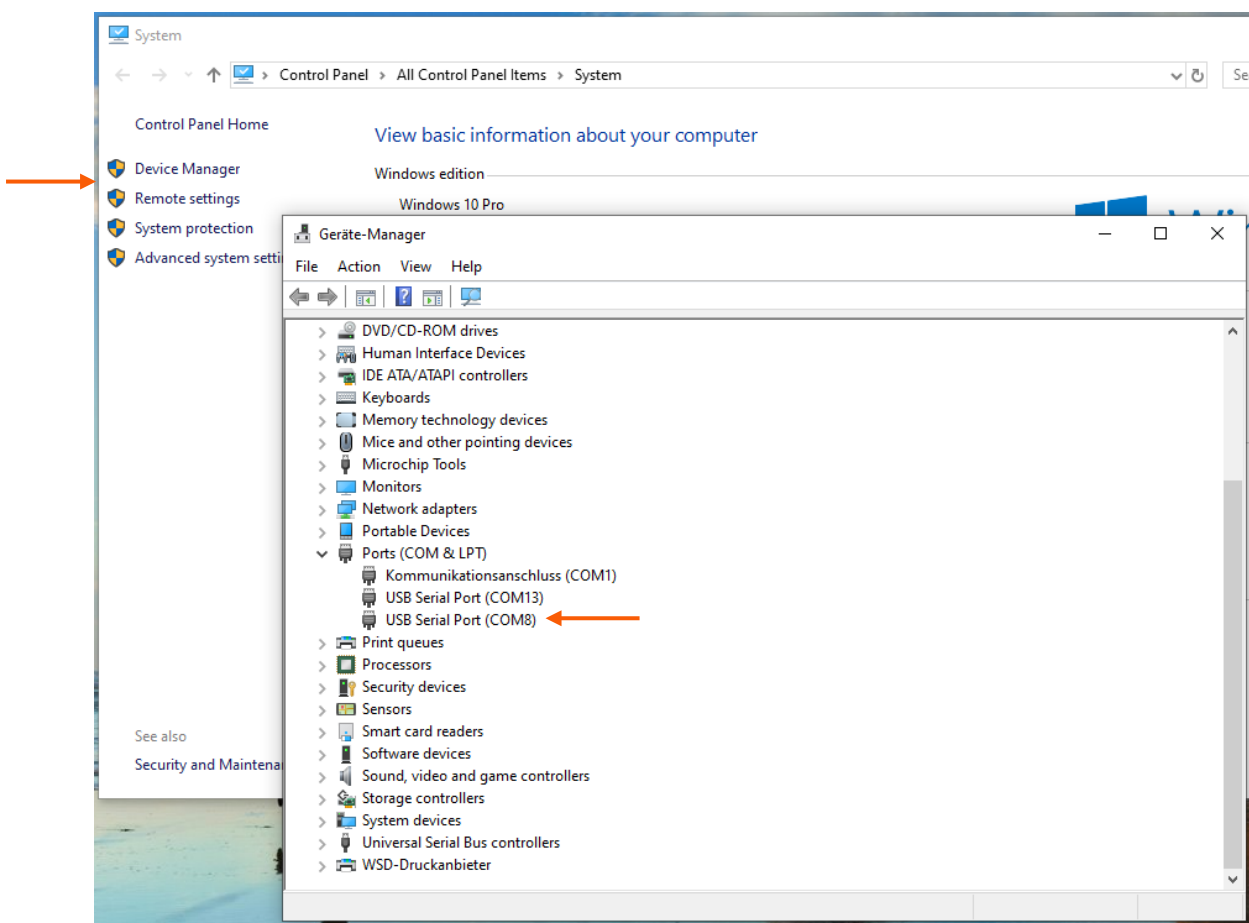


Figure 2 Device Manager

Der hier aufgeführte Serial Port, ist der angeschlossene @ANY USB Dongle. Im Bild 2 wird dieser als COM 8 aufgeführt.

### 3 Erster Schritt: A.N. Solutions ToolChain

#### 3.1 A.N. Solutions ToolChain – SMS Monitor

Der SMS Monitor bietet Ihnen die Möglichkeit ein WSN Netzwerk zu nutzen und visuell darzustellen. Er benötigt keine aufwendige Installation und steht kostenlos in unserer A.N. Solutions ToolChain zur Verfügung. Die A.N. Solutions ToolChain kann auf unserer Webseite heruntergeladen werden. *(Zurzeit nur für Microsoft Windows 10)*

Hierzu navigieren Sie in den Downloadbereich. Unter dem Punkt Tools finden Sie die A.N. Solutions ToolChain.

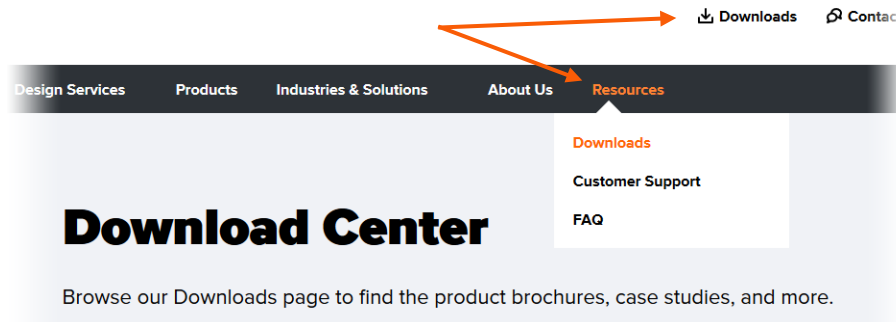


Figure 3 Download Center



Figure 4 Tools

Laden Sie sich die A.N. Solutions ToolChain herunter und entpacken Sie diese an einem beliebigen Ort ihres Windows-PCs. Starten Sie, die sich darin befindende A.N. Solutions ToolChain VXXX.exe.





Das Tool startet und es wird in den TAB SMS Monitor gewechselt. Im SMS Monitor kann sich nun mit dem Koordinator verbunden werden. In Ihrem @ANY Design Kit ist dies der @ANY USB Dongle. Alle mitgelieferten Hardwarekomponenten sind mit dem Beispiel SMS Monitor vorkonfiguriert und auf Funktionalität getestet.

Stecken Sie den @ANY USB Dongle in einen freien USB Port Ihres PCs und stellen Sie sicher, dass dieser ordnungsgemäß erkannt wird. (Kapitel 2)

Im Bereich „Connection Parameters“ wird der COM-Port des @ANY USB Dongles gewählt. Der Port kann von @ANY USB Dongle zu @ANY USB Dongle und System unterschiedlich sein. Durch Drücken der Schaltfläche „Refresh port list“ kann die COM-Portliste aktualisiert werden. Die Einstellungen der Baudrate muss auf 38400bd belassen werden.

Durch drücken auf „OK“ wird sich mit dem @ANY USB Dongle verbunden. Das Fenster „Connection Parameters“ verschwindet und ein weißer Hintergrund erscheint. Es kann immer nur eine Verbindung mit einem Koordinator bestehen. Es ist nicht möglich den SMS Monitor parallel zu einem Kommandozeilen Tool zu nutzen und eine parallele Verbindung mit einem Gerät aufzubauen.

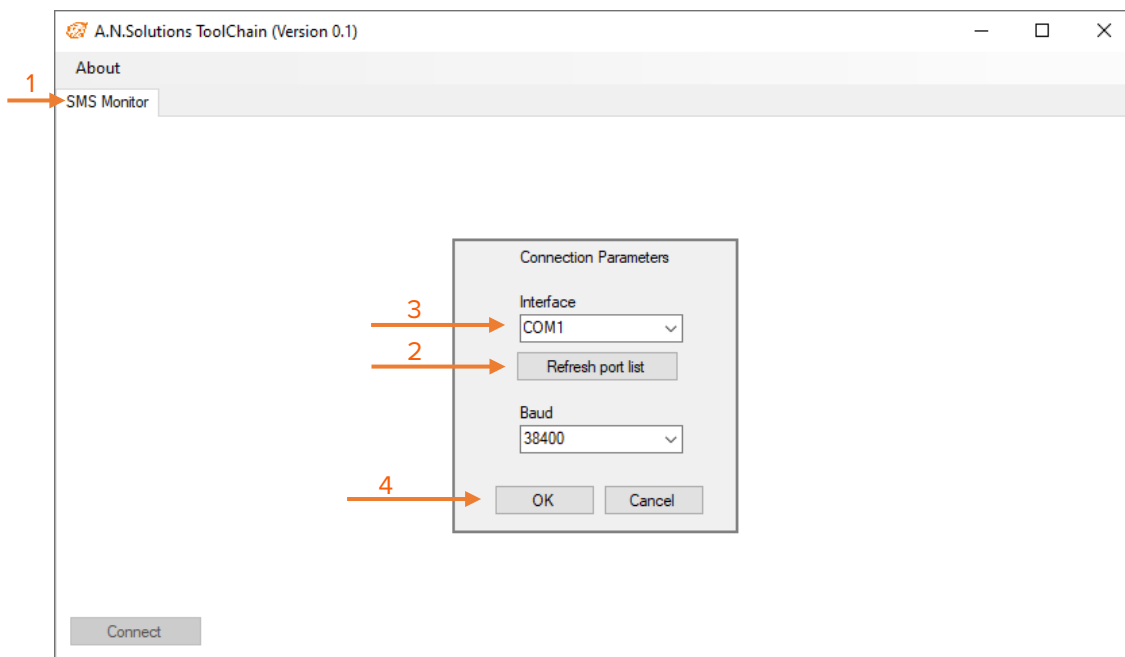


Figure 5 A.N. Solutions ToolChain

Die Batteriehalter sind nun an die @ANY BRICKs anzustecken falls dies noch nicht der Fall ist. Die @ANY BRICKs müssen mit 2x 1,5 V AAA Batterien versorgt werden, welche nicht im Lieferumfang enthalten sind.

Mit dem Schalter S3 wird jeder @ANY BRICK nacheinander eingeschaltet. Die eingeschalteten @ANY BRICKs erscheinen nun im SMS Monitor und senden jede Sekunde einen Temperaturwert an den Koordinator. Die Darstellung auf der nächsten Seite soll dies verdeutlichen.

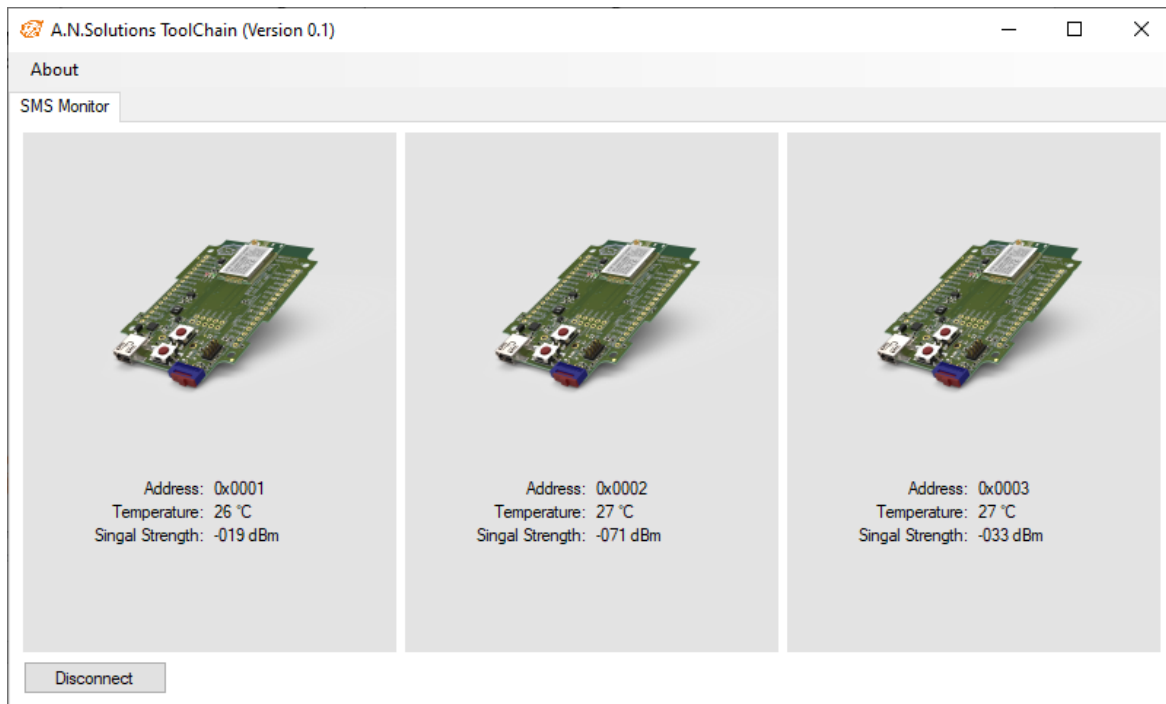


Figure 6 SMS Monitor

Im SMS Monitor wird die Short-Adresse, die Temperatur und die Signalstärke jedes einzelnen @ANY BRICKs aufgeführt. Dies eignet sich zum Beispiel um Reichweitenmessungen durch zu führen.

Herzlichen Glückwunsch, Sie haben soeben Ihr erstes WSN in Betrieb genommen.

### 3.2 Späteres Verwenden des SMS Monitors

Dieses Kapitel ist nicht für den Einstieg geeignet und dient nur der späteren Reaktivierung des SMS Monitors. Hierfür werden Grundkenntnisse im Umgang mit unserer SMS vorausgesetzt. Sollten Sie noch keine Kenntnisse im Umgang mit unserer SMS besitzen, überspringen Sie dies Kapitel.

Sollten Sie zu einem späteren Zeitpunkt den SMS Monitor nutzen wollen, dieser befindet sich aber nicht mehr auf den Geräten, so befolgen Sie die nachfolgenden Schritte für jedes Gerät. Bevor Sie starten, machen Sie sich mit dem Kapitel 2 und 4.1 vertraut.

Um den SMS Monitor zu reaktivieren führen Sie für jedes Gerät die nachfolgenden Befehlsketten aus.

@ANY USB Dongle *Verwenden Sie die erforderlichen Befehlszeilen für die entsprechenden DevKits.*

Vollständiger Befehl:

(2.4 GHz) `ATH1&FS220=0SX202=12ABS208=20S200=0S0=1&W0&Y0Z↵`

(1 GHz EU) `ATH1&FS220=0SX202=12ABS208=0S200=0S0=1&W0&Y0Z↵`

(1 GHz US) `ATH1&FS220=0SX202=12ABS208=1S200=0S0=1&W0&Y0Z↵`

Antwort:

OK & Gerät startet neu

OK & Gerät startet neu

OK & Gerät startet neu

@ANY BRICKs *Verwenden Sie die erforderlichen Befehlszeilen für die entsprechenden DevKits.*

Vollständiger Befehl:

`ATH1&F&Y0&W0&W1Z↵`

`AT*AS234=2S238=1DSA245XS234=2↵`

Antwort:

OK & Gerät startet neu

+NOT ASSOCIATED (EB) &  
ERROR

(2.4 GHz) `ATS220=2SX202=12ABS208=20S200=0S236=4S18=1&W0&Y0Z↵`

(1 GHz EU) `ATS220=2SX202=12ABS208=0S200=0S236=4S18=1&W0&Y0Z↵`

(1 GHz US) `ATS220=2SX202=12ABS208=1S200=0S236=4S18=1&W0&Y0Z↵`

OK & Gerät startet neu

OK & Gerät startet neu

OK & Gerät startet neu

## 4 Getting Started

### 4.1 Navigieren in der SMS

Um mit den @ANY USB Dongle und den @ANY BRICKs kommunizieren zu können, ist ein Kommandozeilen Tool sinnvoll. Theoretisch kann jedes Kommandozeilen Tool genutzt werden, welches es ermöglicht eine serielle Verbindung aufzubauen. Wir empfehlen die Verwendung von HTerm. Wechseln Sie in Ihrem Kommandozeilen Tool auf die serielle Verbindung.

Im Folgenden werden die einzelnen Schritte anhand des @ANY USB Dongels erklärt, dies ist aber ebenso mit den @ANY BRICKs möglich. Stecken Sie den @ANY USB Dongel in einen freien USB-Port an Ihrem PC. Verbinden Sie sich mit dem COM Port, welchem der @ANY USB Dongle zugewiesen wurde und stellen Sie die Baudrate auf 38400bd. Sinnvoll ist noch das Setzen von 8 Datenbits, keinem Paritätsbit und einem Stoppbit (8 N 1).

Es kann immer nur eine Verbindung mit einem Tool und einem @ANY USB Dongel oder @ANY BRICK aufgebaut werden.

Sobald die Verbindung zwischen Hard- und Software hergestellt ist, erfolgt ein bidirektionaler Kommunikationskanal zwischen dem @ANY USB Dongle und dem Nutzer. Der Nutzer kann nun Befehle in das Terminal eingeben und der @ANY USB Dongle antwortet.

Die bestehende Verbindung kann durch eingeben des **AT** Befehls geprüft werden. Der @ANY USB Dongle antwortet mit einem **OK**.



Figure 7 AT Command

Jede Kommandozeile startet mit einem **AT**. Es wird nicht zwischen Groß- und Kleinschreibung unterschieden, außer bei einem bestimmten Befehl zum Speichern der IEEE Adresse (**AT&Waddr**). Eine Kommandozeile kann aus mehreren Kommandos bestehen. Diese werden nacheinander verarbeitet. Ein fehlerhafter Befehl wird mit einer Fehlermeldung zurückgegeben. Mehrdeutige Befehlssequenzen können explizit durch das „X“ getrennt werden. Die maximale Länge einer Kommandozeile beträgt 114 Zeichen. Ein Gerät kann mittels Befehl **ATZ** neugestartet werden. Ein Gerät kann mit dem Befehl **AT&F** auf Werkseinstellungen zurückgesetzt werden.

### 4.2 Alle Geräte auf Werkseinstellungen zurücksetzen



Bevor mit dem Getting Started begonnen werden kann, muss das Beispiel SMS Monitor von dem @ANY USB Dongle und den @ANY BRICKs gelöscht werden.

Das Löschen des Beispiels SMS Monitor, muss auf jedem Gerät einzeln erfolgen. Befolgen Sie deshalb für jedes Gerät die nachfolgenden Schritte.

Um den @ANY USB Dongle oder den @ANY BRICK in die Werkseinstellungen zurück zu setzen, geben Sie folgende Befehlszeile in das Kommandozeilen Tool ein und bestätigen die Eingabe mit Enter ↵.

Vollständiger Befehl:  
**ATH1&F&Y0&W0&W1Z** ↵  
Antwort:  
OK & Gerät startet neu

### 4.3 Informationen ausgeben lassen

Als Beispiel wollen wir uns Informationen über das Gerät ausgeben lassen. Die Syntax für diese Infos, sieht wie folgt aus: `ATI <n>`, wobei das `<n>` für bestimmte Argumente steht, welche genutzt werden können.

Argument	Ausgabe
0	<ISM band>
1	<firmware rev.>
2	<Build-ID>
3	ANS - Smart MAC Suite V-<firmware rev.>
4	<board type>
5	<Programm memory CRC>
6	<radio chip type>
7	<license owner> (Pro version only)
any	OK (argument in 1 ... 6 (7)) or ERROR

Table 2 ATI Command

Beispiel:

Einzel:  
`ATI0 ↵`  
 2400  
 OK  
`ATI3 ↵`  
 ANS – Smart Mac Suite V-1.47  
 OK  
`ATI4 ↵`  
 @ANY2400SC-3  
 OK  
`ATI6 ↵`  
 ATMEGA128RFA1  
 OK

```
ATI0
2400
OK
ATI3
ANS - Smart MAC Suite V-1.47
OK
ATI4
@ANY2400SC-3
OK
ATI6
ATMEGA128RFA1
OK
```

Figure 8 ATI long

Kombiniert:  
`ATI0I3I4I6 ↵`  
 2400  
 ANS – Smart MAC Suite V-1.47  
 @ANY2400SC-3  
 ATMEGA128RFA1  
 OK

```
ATI0I3I4I6
2400
ANS - Smart MAC Suite V-1.47
@ANY2400SC-3
ATMEGA128RFA1
OK
```

Figure 9 ATI small

#### 4.4 Profile anzeigen, bearbeiten, speichern

Die SMS bietet die Möglichkeit 2 Profile zu erstellen. In einem Profil werden verschiedenen Konfigurationen der SMS gespeichert. Ein Profil ist eine Abbildung aller S-Register, wobei nur die wichtigsten S-Register mit dem Befehl **AT&V** als eigentliches Profil ausgegeben werden. Somit ist ein Profil nur eine geordnete Übersicht über die wichtigsten Konfigurationen. Der erste Teil des Profils, beschreibt die Einstellungen des UART und verschiedene Kommunikationsoptionen, welche als S-Register und Adressen aufgeführt sind.

```
E1 Command Echo | F1 Online Echo | Q0 Result Code Opt. | V1 Result Code Format | &C0 DCD Opt.  
&D0 DTR Opt. | &K0 Local Flow Control | &Q0 Communication Mode | &R1 RTS/CTS Opt. | &S0 DSR Opt.
```

```
E1 F1 Q0 V1 &C0 &D0 &K0 &Q0 &R1 &S0  
S00:000 S02:043 S03:013 S04:010 S05:008 S17:016 S18:000
```

Figure 10 Part one profile

**S00** Auto Mode ATA | **S02** Esc character | **S03** CR character | **S04** LF character | **S05** BS character  
**S17** Timer Prescaler | **S18** Timer

Der zweite Teil beinhaltet die Optionen für ein Wireless Sensor Netzwerk (WSN).

```
IEEE ADDRESS: FE.95.3C.83.87.97.4F.FF  
DEVICE ROLE: END DEVICE  
PAN ID: FFFF SHORT ADDRESS: 0000  
CHANNEL: 20 (0) CHANNEL MASK: 07FFF800
```

Figure 11 Part two profile

##### **IEEE ADDRESS:** EEPROM

Die IEEE Adresse ist eine Hardwareadresse. Diese wird nur im AKTIVE PROFILE angezeigt. Sie wird nicht in einem Profil abgelegt, sondern im EEPROM gespeichert. Bei einem Start sucht die SMS die IEEE Adresse zuerst im EEPROM, ist dort keine hinterlegt wird nach dem Hardwarechip DS2411 gesucht, ist dieser auch nicht vorhanden generiert die SMS eine zufällige IEEE Adresse. Das Ändern der IEEE Adresse erfolgt im Kapitel 3.3.2. Jedes Gerät im Netzwerk muss eine feste IEEE Adresse zugewiesen bekommen.

##### **DEVICE ROLE:** Register S220 8bit 0=Koordinator, 1=FF device (SMS\_Pro), 2=END device

Die Rolle des Gerätes legt das Verhalten im späteren WSN fest. Ein **Koordinator** erstellt ein WSN und wartet auf FF Geräte oder End Geräte, die sich verbinden wollen. Dieser ist vergleichbar mit einem Netzwerkhost welcher die Koordination im WSN übernimmt. Ein **END Gerät** kann sich nur mit einem Koordinator verbinden und auch nur an den Koordinator Daten senden oder vom Koordinator daten empfangen. Ein **FF Gerät** (SMS Pro) ist Koordinator unabhängig und kann Daten mit anderen FF Geräten austauschen. Es bietet die Möglichkeit kleine und große Mesh/Tree-Netzwerke zu erstellen.

##### **PAN ID:** Register S202 16bit

Die PANID ist die Netzwerkadresse im physischen Netzwerk. Diese muss bei allen Teilnehmern im selben Netzwerk gleich sein und wird hier im Getting Started als Hexadezimalzahl ausgegeben.

##### **SHORT ADDRESS:** Register S200 16bit

Das ist die Adresse der einzelnen Teilnehmer im Netzwerk und dient als Identifikation. Diese wird vom Koordinator den einzelnen Teilnehmern zugewiesen. Der Koordinator muss immer die SHORT ADDRESS 0x000 manuell zugewiesen bekommen. Ein Ansprechen der einzelnen Teilnehmer über die IEEE Adresse ist auch möglich. Sie wird ebenfalls als Hexadezimalzahl ausgegeben.

### CHANNEL: Register S208 8bit

Dies ist der physikalische Kanal zum Datenaustausch mit anderen Geräten im Netzwerk. Er muss für alle Teilnehmer im gleichen Netzwerk, mit der gleichen PANID gleich sein. Es ist wichtig sicherzustellen, dass das Gerät mit einer Frequenz von 868,3 MHz EU dem Kanal 0 zugeordnet ist, mit der Frequenz 915MHz 1 - 10 USA und für Japan und China wenden Sie sich bitte an den A.N.S. Customer Support. Geräten mit einer Frequenz von 2,4 GHz wird ein Kanal im Bereich von 11 - 26 zugewiesen.

### CHANNEL MASK: Register S204 32bit

Der IEEE 802.15.4 Standard besitzt eine 32 Bit Channel Mask. Jedes einzelne Bit, in der Channel Mask, bildet einen Kanal ab. Die SMS nutzt insgesamt 26 Kanäle. Mit der Channel Mask kann ein Bereich von Kanälen definiert werden. Dies wird zum Beispiel zum Channel scannen genutzt um mehrere Kanäle abzufragen.

Genauere Informationen zu den einzelnen S-Registern entnehmen Sie bitte dem Dokument „SMS Command Reference“. Informationen über Netzwerkparameter entnehmen Sie dem Standard IEEE Std 802.15.4™-2006 (Revision of IEEE Std 802.15.4-2003).

```
AT&V
ACTIVE PROFILE:
E1 F1 Q0 V1 &C0 &D0 &K0 &Q0 &R1 &S0
S00:000 S01:000 S02:043 S03:013 S04:010 S05:008
S10:000 S12:000 S17:016 S18:000 S25:005 S26:001 S38:020
IEEE ADDRESS: FE.95.3C.83.87.97.4F.FF
DEVICE ROLE: END DEVICE
PAN ID: FFFF SHORT ADDRESS: 0000
CHANNEL: 20 (0) CHANNEL MASK: 07FFF800

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: END 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
S00:000 S02:043 S03:013 S04:010 S05:008 S17:016 S18:000
DEVICE ROLE: END DEVICE
PAN ID: FFFF SHORT ADDRESS: 0000
CHANNEL: 20 (0) CHANNEL MASK: 07FFF800

OK
```

Figure 12 Full profile overview

Es kann zwischen dem STORED PROFILE 0 oder 1 gewählt werden, wobei eines der beiden Profile immer das Standardprofil ist. Dies kann mit dem Befehl **AT&Y<n>** definiert werden. Das <n> steht für das zu wählende Profil 0 oder 1. Welches der beiden Profile als Standard gewählt wurde, wird durch ein (DEFAULT) neben der Profilnummer gekennzeichnet.

Wird ein Profil bearbeitet, so geschieht dies immer im ACTIVE PROFILE. Werden die Änderungen nicht in einem der beiden STORED PROFILE gespeichert, so gehen die Änderungen bei einem Neustart verloren. Speichern lässt sich ein geändertes Profil mit dem Befehl **AT&W<n>**. ( 0 oder 1 )

Das ändern des Profils geschieht über das Schreiben in die jeweiligen S-Register. Mit dem Befehl **ATS<Reg.ID>** ist dies möglich. S-Register können mit einer Dezimalzahl oder mit einer Hexadezimalzahl beschrieben werden. Die Größe der Dezimal-/Hexadezimalzahl ist vom jeweiligen Register abhängig.



Beispiel: Schreiben der Zahl 27 in das S-Register 240

Dezimal: `ATS240=27 ↵`      Hex: `ATSX240=1B ↵`

Das **X** in der Syntax sagt der SMS, dass in dem S-Register 240 eine Hexadezimalzahl gespeichert werden soll. Es ist auch möglich S-Register auszulesen, mit dem Befehl `ATS<Reg.ID>?`. Hier ist aber auch darauf zu achten, dass bei Hexadezimalzahl das **X** nicht vergessen wird also, `ATSX<Reg.ID>?`.

## 4.5 Konfigurieren der SMS per Console

### 4.5.1 Konfigurieren des @ANY USB Dongle als Koordinator

Im späteren Verlauf des Getting Started soll der @ANY USB Dongle als Koordinator dienen. Es ist immer Vorteilhaft den @ANY USB Dongle als Koordinator einzurichten, auf Grund seiner höheren Leistung durch den USB-Port.

Der @ANY USB Dongle übernimmt die Rolle des Koordinators. Er sendet auf der Frequenz 2,4 GHz und bekommt somit den Channel 20 zugewiesen. *Für das Sub-1 GHz-Frequenzband müssen je nach Land andere Kanäle eingestellt werden. Siehe Kapitel 4.4 Beschreibung der Channels.* Die PAN ID ist frei wählbar und wird hier auf 0x12AB gesetzt. Die SHORT ADDRESS wird auf Grund der zugewiesenen Rolle auf 0x000 eingestellt und die Channel MASK kann vernachlässigt werden. Die Einstellungen werden im Profil 0 gespeichert und dieses soll gleichzeitig das Standard Profil sein. Nach dem setzen der S-Register soll das Gerät neustarten.

Befehl:	Antwort:	Erläuterung:
<code>ATS220=0 ↵</code>	OK	Setzen des S-Registers 220 auf 0 dec., Koordinator
<code>ATSX202=12AB ↵</code>	OK	Setzen der PANID im S-Register 202 als Hex-Zahl auf 12AB
<code>ATS208=20 ↵</code>	OK	Setzen des Channels im S-Register 208 auf 20 dec.
<code>ATS200=0 ↵</code>	OK	Setzen des gesamten S-Registers 200 auf 0, Short Adresse
<code>AT&amp;W0 ↵</code>	OK	Speichern der Einstellungen im Profil 0
<code>AT&amp;Y0 ↵</code>	OK	Das Profil 0 als Standard Profil definieren
<code>ATZ ↵</code>	OK ...	Das Gerät neu starten

Kombiniert: `ATS220=0SX202=12ABS208=20S200=0&W0&Y0Z`

```

AT&V
ACTIVE PROFILE:
E1 F1 Q0 V1 &C0 &D0 &K0 &Q0 &R1 &S0
S00:000 S01:000 S02:043 S03:013 S04:010 S05:008
S10:000 S12:000 S17:016 S18:000 S25:005 S26:001 S38:020
IEEE ADDRESS: FE.95.3C.83.87.97.4F.FF
DEVICE ROLE: COORDINATOR
PAN ID: 12AB SHORT ADDRESS: 0000
CHANNEL: 20 (0) CHANNEL MASK: 07FFF800

STORED PROFILE 0 (DEFAULT):
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: COORDINATOR
PAN ID: 12AB SHORT ADDRESS: 0000
CHANNEL: 20 (0) CHANNEL MASK: 07FFF800

STORED PROFILE 1:
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: END DEVICE
PAN ID: FFFF SHORT ADDRESS: 0000
CHANNEL: 20 (0) CHANNEL MASK: 07FFF800

OK

```

Figure 13 Full config dongle

#### 4.5.2 Setzen der IEEE Adresse

Die IEEE Adresse ist eine 64 Bit Hardwareadresse, welche als Hexadezimalzahl dargestellt wird und in den ersten acht Byte im EPROM gespeichert ist. Ob eine IEEE-Adresse vorkonfiguriert ist, sehen Sie unter Verwendung des Befehls `AT+CF0?`, die dort vermerkte IEEE Adresse beginnt immer mit einem C47DFE. Ist das nicht der Fall, ist davon auszugehen, dass keine von ANS definierte IEEE Adresse im EEPROM steht.

`AT+CF0=<xxxxxxxxxxxxxxxxxx>` Die Angabe der IEEE Adresse erfolgt in HEX.

OK

`AT&Waddr` Mit diesem Befehl wird die IEEE Adresse im EPROM hinterlegt. Hier ist auf Groß und Kleinschreibung zu achten!

OK

Vollständiger Befehl:

`AT+CF0=0000000000000FF&WaddrZ ↵`

Antwort:

OK & Gerät startet neu



### 4.5.3 Konfigurieren des @ANY Brick als END-Gerät

Der @ANY BRICK wird im späteren Verlauf, des Getting Started, als End-Gerät dienen und mit einer 1,5 V AAA versorgt. Für die Konfiguration als End-Gerät ist es notwendig die @ANY BRICKs per USB mit dem PC zu verbinden. Der Schalter S3 muss sich in der Richtigen Position befinden und die grüne LED beginnt zu leuchten (Standard Firmware Einstellungen). Der @ANY BRICK wird auch, wie der @ANY USB Dongle, im Geräte-Manager als COM Port angezeigt und lässt sich über ein Kommandozeilen Tool konfigurieren.

Der @ANY BRICK soll sich im gleichen Netzwerk wie der @ANY USB Dongle befinden, damit diese miteinander kommunizieren können. Daraus ergeben sich folgende Einstellungen für den @ANY BRICK:

Device Roll	PAN ID	Channel*	Short address
End device	12AB	20*	Vergabe durch Koordinator

Table 3 Config BRICK

\* Für das Sub-1 GHz-Frequenzband müssen je nach Land andere Kanäle eingestellt werden. Siehe Kapitel 4.4 Beschreibung der Channels.

Vollständiger Befehl:

```
ATS220=2SX202=12ABS208=20S200=0&W0&Y0Z ↵
```

Antwort:

OK & Gerät startet neu

Es ist sinnvoll dem End-Geräten eine definierte IEEE Adresse zuzuweisen, weil der Koordinator sich diese während der Laufzeit merkt und eine Short Adresse vergibt. Die End-Geräte selber werden über die Short Adresse im Netzwerk angesprochen. Meldet sich das gleiche Gerät mit einer anderen IEEE Adresse bei Koordinator, bekommt dieses, eine neue Short Adresse zugewiesen und andere ProgrammROUTINEN, die mit der alten Short Adresse arbeiten, erreichen das End-Gerät nicht mehr.

Vollständiger Befehl:

```
AT+CF0=00000000000000EE&WaddrZ ↵
```

Antwort:

OK & Gerät startet neu

## 4.6 Aufbau eines Wireless Sensor Network (WSN)

Um ein funktionierendes Netzwerk mit @ANY USB Dongle und @ANY BRICKs aufzubauen, müssen wie in den vorher behandelten Kapiteln die Geräte eingerichtet werden. Ein Koordinator baut mittels ATA Befehl ein Netzwerk auf und die End-Geräte können sich mittels ATA Befehl einwählen. Der Koordinator speichert die sich bei ihm eingewählten End-Geräte anhand der IEEE-Adressen und verteilt den End-Geräten eine Short Adresse. Der Datenaustausch im Netzwerk erfolgt über die Short Adressen, kann aber auch über die IEEE Adressen realisiert werden. Wird ein Koordinator stromlos geschaltet oder neu gestartet, so verliert er diese Liste. Mittels Befehl AT&V2 kann diese Liste aufgerufen werden. Die Short Adressen werden in der Reihenfolge vergeben, wie die End-Geräte sich einwählen. Ist ein End-Gerät einem Koordinator bekannt aber nicht verbunden, so taucht seine IEEE in dieser Liste auf, besitzt aber eine Short Adresse von 000. Eine bestehende Verbindung kann mittels Befehl ATH beendet werden. Das Suchen von Netzwerken erfolgt über den Befehl AT+S.

### 4.6.1 Peer to Peer Netzwerk

Wir wollen nun ein einfaches Peer to Peer Netzwerk zwischen dem @ANY USB Dongle und einem @ANY BRICK aufbauen. Außerdem sollen Daten zwischen den beiden Teilnehmern ausgetauscht werden.

Mittels Befehl AT&V werden die vorher eingestellten Einstellungen geprüft. Auf dem Koordinator wird der Befehl ATA ausgeführt. Anschließend wird der gleiche Befehl auf dem End-Gerät ausgeführt. Mittels Befehl AT&V2 kann die Liste im Koordinator geprüft werden.



Koordinator:	End-Gerät
ATA ↵ OK	
+DEVICE ASSOCIATED: 0001 (80)	ATA ↵ +ASSOCIATED: 0001 OK
AT&V2 ↵ 00: 0001 / 00.00.00.00.00.00.EE (80) OK	

Der Koordinator hat ein WSN aufgebaut in welches sich das End-Gerät eingewählt hat. Beim erfolgreichen einwählen in ein WSN, meldet das End-Gerät +ASSOCIATED und die Short Adresse welches ihm zugewiesen wurde. Der Koordinator meldet ebenso, dass ein End-Gerät sich bei ihm eingewählt hat und welche Short Adresse diesem zugewiesen wurde.

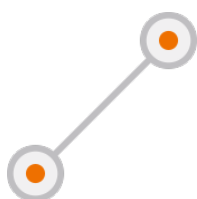


Figure 14 Peer to Peer Network

Es ist möglich mittels Befehl **ATD** Daten zwischen den beiden Geräten auszutauschen. Der Koordinator muss immer das Ziel angeben, wobei das End-Gerät nur an den Koordinator senden kann. Ein Beispiel auf der nächsten Seite, soll dies verdeutlichen.

Beispiel:

Koordinator:	End-Gerät
	ATD; ↵ HALLO ↵ ID 001: 5 BYTES to 0000 OK + SENT: ID 001
+ DATA: 5 BYTES FROM 0001 (074) HALLO	
ATD1; ↵ HALLO ZURUECK ↵ ID 002: 13 BYTES TO 0001 OK	
+SENT: ID 002	ATD? ↵ OK +DATA: 13 BYTES FROM 0000 (077) HALLO ZURUECK

Das End-Gerät sendet einen ASCII Code zum Koordinator. Hinter dem **ATD** Befehl muss ein “ ; „ gesetzt werden, sonst wartet die SMS auf die Angabe der Länge, der zu sendenden Daten (siehe SMS Command Reference Befehl ATD). Die gesendeten Daten erscheinen direkt beim Koordinator. Möchte der Koordinator Daten an ein End-Gerät senden, so muss der Koordinator die Short Adresse des End-Gerätes ansprechen. Hier fehlt die +SENT Benachrichtigung was heißt, dass die Nachricht gespeichert wird bis das angesprochene End-Gerät nach Daten fragt, mittels Befehl **ATD?** (siehe auch S-Register 221). Der Koordinator speichert die Daten nur für begrenzte Zeit (ca. 1min) womit es sinnvoll ist, das End-Gerät in Intervallen nach Daten Fragen zu lassen. Die Zahl in Klammern, beim Empfänger, gibt die Empfangsleistung in dB an.

Herzlichen Glückwunsch, Ihr ersten Peer to Peer Netzwerk wurde aufgebaut.

#### 4.6.2 Stern Netzwerk

Das Stern Netzwerk baut auf dem Peer to Peer Netzwerk auf. In diesem Netzwerk basiert der Datenaustausch auf mehreren Endgeräten, welchen mit einem Koordinator kommunizieren. Die End-Geräte können untereinander nicht kommunizieren. Für dieses Netzwerk werden die übrigen zwei @ANY BRICKs für das bestehende Netzwerk konfiguriert. Das vorhandene Peer to Peer Netzwerk wird aufgelöst, durch neu starten des @ANY USB Dongle und des @ANY BRICKs.

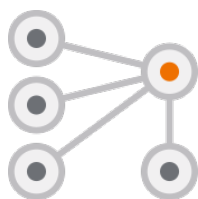


Figure 15 Star - Network

#### Konfiguration @ANY USB Dongle

Damit sich die Teilnehmer automatisch verbinden können, muss der Koordinator so konfiguriert werden, dass er beim Hochfahren direkt das Netzwerk aufbaut. Dazu muss das S-Register S0 auf 1 gesetzt werden.

#### Konfiguration der @ANY BRICKs

Die @ANY BRICKs sollen alle 10 Sekunden oder per Taster S2 die Temperatur des LM73 an den Koordinator senden. Das Senden wird durch das Blinken der LED 2 signalisiert.

Damit auf dem @ANY BRICK der LM73 die Temperatur misst, muss das S-Register 238 mit einer 1 getoggelt werden. Die erhaltenen Temperaturwerte, werden im S-Register 245 als Dezimalzahl gespeichert. Um die Temperatur alle 10 Sekunden zu senden, wird ein Timer-Interrupt benötigt, welcher über das S-Register 18 zur Verfügung steht. Die Gewünschten Sekunden werden als Dezimalzahl eingetragen. Möchte man kleinere Intervalle erreichen, wie 1 Millisekunde, so muss das S-Register 17 ebenfalls als Teiler-Register genutzt werden, dazu siehe SMS Command Reference Kapitel S-Register.

Das Triggern des S238 Registers und das Senden des Inhaltes aus Register 245 soll automatisiert ablaufen. Dies wird über das Shadow Command Register (SCR) realisiert. Im SCR ist es möglich ganze Kommandozeilen zu hinterlegen. Das SCR reagiert auf Interrupts und führt daraufhin seine gespeicherte Kommandozeile aus. Das schreiben einer Kommandozeile geschieht mit dem Befehl **AT\*** welcher das normale **AT** ersetzt. Die Kommandozeile kann in Profilen gespeichert werden.



Als erstes, wird die komplette Befehlskette in den SCR geschrieben:

Vollständiger Befehl:

```
AT*AS234=2S238=1DSA245XS234=2 ↵
```

Antwort:

OK

AT*	Schreibt die nachfolgende Kommandozeile in den SCR
A	Verbindet sich mit dem Netzwerk, falls dies noch nicht geschehen ist
S234=2	Toggelt LED 2 (Infos zum S-Register weiter unten).
S238=1	Der LM73 misst einmal die Temperatur und legt diese im S-Register 245 ab
DS	Sendet das folgende S-Register an den Koordinator
A	Als ASCII
245	Das S-Register welches gesendet wird
X	Das Senden wird hier abgebrochen und die folgende Kommandozeile wird nicht mitgesendet
S234=2	Toggelt LED 2

Damit der Taster S2 den SCR ausführen kann, muss in das S-Register 236 eine 4 geschrieben werden (Infos zum S-Register weiter unten). Das Senden soll auch hier durch das Blinken der LED 2 signalisiert werden.

Als erstes, schreiben der 4 in des S-Registers 236. Als nächstes, wird der Timer im S-Register 18 auf 10 Sekunden eingestellt. Die Konfiguration wird im Profil 0 gespeichert und der @ANY BRICK neu gestartet.

Vollständiger Befehl:

```
ATS236=4S18=10&W0Z ↵
```

Antwort:

OK & Gerät startet neu

Der @ANY Dongle ist über einen USB Port verbunden, über ein Kommandozeilen-Tool ansprechbar und reagiert auf ein AT mit OK. Durch Eingabe des ATA Befehls, baut der @ANY USB Dongle ein Netzwerk auf.

Die Batterieholder können an die @ANY BRICKs gesteckt werden, die +1,5V AAA Batterien eingelegt und die Taoglas Antennen mit den jeweiligen @ANY Modulen verbunden. Zum Starten der @ANY BRICKs wird der Schalter S3 umgelegt. Die LED 1 grün beginnt zu

Nach den ersten 10 Sekunden meldet sich ein @ANY BRICK beim Koordinator mit: +ASSOCIATED: 00X und nach weiteren 10 Sekunden erfolgt die erste Temperaturmeldung, mit: +DATA: 3 BYTES FROM 000X (0XX) 0XX. Beim Senden der Temperatur blinkt die LED 2 gelb kurz auf. Durch drücken des Tasters S2 erfolgt eine direkte Temperaturmeldung an den Koordinator.

Neustarten können Sie einen @ANY BRICK über das drücken des Tasters S1, woraufhin der @ANY BRICK sich neu mit dem Netzwerk verbindet und Temperaturen an den Koordinator sendet.

Herzlichen Glückwunsch Sie haben soeben Ihr zweites Netzwerk konfiguriert und in Betrieb genommen.



## Informationen zu S-Register 234, 236

S-Register 234:

Bit	7	6	5	4	3	2	1	0
	-	-	-	-	LED all	LED red	LED yellow	LED green
GPIO						2	1	0

S-Register 236:

Bit	7-4	3	2	1	0
	-	Interruptpin 7	Interruptpin 6 (Taster S2)	Interruptpin 7	Interruptpin 6 (Taster S2)
	-	Enable command execution		Enable Wake up	



Sie können sich das soeben eingerichtete WSN, auch mit dem SMS Monitor anzeigen lassen. Öffnen Sie dazu die A.N. Solutions ToolChain und navigieren Sie in den TAB „SMS Monitor“. Lösen Sie die Verbindung mit dem Kommandozeilen Tool und Ihrem gesetzten Koordinator. Im SMS Monitor kann sich nun mit dem Koordinator verbunden werden. Nutzen Sie dazu das Fenster „Connection Parameters“. Nach wenigen Sekunden erscheinen die @ANY BRICKs im SMS Monitor. Siehe dazu Kapitel 2

## 4.7 Mesh Netzwerk für SMS Pro Nutzer

### 4.7.1 Einrichten der Netzwerkteilnehmer



Dieses Kapitel baut auf dem gesamten Getting Started auf und setzt die Kenntnisse von der Einrichtung eines Netzwerkes mit unserer Smart Mac Suite (SMS) voraus. Mesh-Netzwerke können nur mit der SMS Pro realisiert werden. Alle Teilnehmer in diesem Netzwerk müssen den gleichen Channel und die gleiche PANID zugewiesen bekommen. Sollten nur die hier aufgeführten Befehle verwendet werden wird das Mesh-Netzwerk nicht funktionieren. In diesem Fall lesen sie Kapitel 3.3.1 und 3.4.1 für volle Funktionalität.

Für Anwendungen, bei denen die oben beschriebenen Ansätze nicht geeignet sind, ist in SMS Pro eine vollständige Routing-Schicht auf dem IEEE 802.15.4-Standard integriert. Sie bietet die Möglichkeit, selbstbildende und selbstheilende Maschennetze aufzubauen.

Der Routing-Algorithmus selbst ist dem LwMesh-Stapel entnommen. Aus den verschiedenen Optionen wurde ein "nativer Routing"-Mechanismus mit Multicast-Unterstützung ausgewählt und mit den Standardparametereinstellungen von LwMesh implementiert. Für Einzelheiten wird auf den LwMesh-Stapel und die Dokumentation verwiesen.

Unser bestehendes Sternnetzwerk muss aufgelöst werden. Alle Geräte werden mit dem PC verbunden um diese per Kommandozeilen Tool einzurichten. Als erstes wird der Timer im S-Register 18 bei allen @ANY BRICKs auf 0 gesetzt und die Verbindung zum Koordinator mit **ATH** gelöst.

End-Device:

ATS18=0 ↵

OK

ATH ↵

OK

Koordinator:

+DISASSOCIATED: 000X

Um dynamisches Routing zu ermöglichen muss das S-Register 223 mit einer 5 beschrieben werden. Bei Verwendung einer Routing-Schicht müssen alle Geräte im Netzwerk die gleichen S223-Einstellungen besitzen. Bei dieser Routing Methode, wird eine statisch vordefinierte Short Adresse benötigt. Außerdem müssen alle Geräte die Rolle des FF-Device zugewiesen bekommen. Eine Kombination aus Koordinator und FF-Device ist auch möglich. Hier wird aber nur ein reines FF-Device Mesh beschrieben.



Die folgenden Einstellungen werden bei allen Geräten, @ANY USB Dongle und @ANY BRICKs vorgenommen. Es ist darauf zu achten, dass jedes Gerät eine andere Short Adresse bekommt!

Befehl:	Antwort:	Erläuterung:
ATS223=5 ↵	OK	Aktivieren der dynamischen Routingfunktion
ATS220=1 ↵	OK	Zuweisen der Geräterolle als FF-Device
ATS200=000X ↵	OK	Setzen der Short Adresse, für jedes Gerät unterschiedlich
AT&W0 ↵	OK	Speichern der Einstellungen im Profil 0
ATZ ↵	OK	Das Gerät neu starten
Kombiniert:		ATS223=5S220=1S200=0001&W0Z

Um die Einstellungen zu testen wird eine Verbindung zwischen dem @ANY USB Dongle und einem @ANY BRICK mit dem Befehl **ATA** hergestellt. Es erfolgt keine +DEVICE ASSOCIATED: 000X (80) Meldung.

@ANY USB Dongle Short Adresse: 0000	@ANY BRICK Short Adresse: 0001
ATA ↵	ATA ↵
OK	OK
	ATD0000; ↵
	TEST1 ↵
+DATA: 5 BYTES FROM 0002 (011)	ID 002: 5 BYTES to 0000
TEST1	OK
ATD0001; ↵	+SENT: ID 002
TEST2 ↵	
ID 002: 5 BYTES TO0001	
OK	+DATA: 5 BYTES FROM 0002 (011)
+SENT: ID 002	TEST2

War die Übertragung Erfolgreich, sind die wichtigsten Schritte für ein Mesh-Netzwerk getan. Ist die Übertragung mit einer Fehlermeldung abgebrochen, dann überprüfen Sie ob alle Geräte die gleiche PANID, den gleichen Channel und eine unterschiedliche Short Adresse besitzen. Außerdem muss vorher der Befehl **ATA** ausgeführt werden.

Um das dynamische Routing besser zu verstehen, wird ein @ANY BRICK als Knoten fungieren und der zweite sendet permanent Temperaturdaten an den @ANY USB Dongle.

Der eine @ANY BRICK welcher als Knoten fungieren soll, muss S-Register 0 auf 1 gesetzt werden. Somit ist die automatische Verbindung zu dem eingerichteten Netzwerk aktiv. Nach dem einstellen des S-Registers wird alles im Profil 0 mit **AT&W0** gespeichert.

Vollständiger Befehl:

ATS0=1&W0 ↵

Antwort:

OK

Bei dem @ANY BRICK welcher als Datenlogger fungiert, kommt wieder das Shadow Register zu Einsatz, welches mit einem 3 Sekunden Timer ausgelöst wird. Der LM73 wird über das S-Register 238 einmal getriggert und legt die Temperatur in das S-Register 245 ab. Dieses soll beim @ANY USB Dongle als ASCII ausgegeben werden.

Vollständiger Befehl:

AT\*AS238=1D0000SA245 ↵

Antwort:

OK

AT*	Schreibt die nachfolgende Kommandozeile in den SCR
A	Verbindet sich mit dem Netzwerk, falls dies noch nicht geschehen ist
S238=1	Der LM73 misst einmal die Temperatur und legt diese im S-Register 245 ab
D0000	Senden an die Adresse 0000 (@ANY USB Dongle)
SA	Das folgende S-Register, welche gesendet werden soll, als ASCII
245	Das S-Register welches gesendet wird



Nun wird der Timer im S-Register 18 auf 3 Sekunden eingestellt und das Ganze im Profil 0 gespeichert.

Vollständiger Befehl:

ATS18=3&W0 ↵

Antwort:

OK

#### 4.7.2 Einrichten und Inbetriebnahme der Testumgebung

Als ersten werden alle @ANY BRICKs vom PC getrennt. Es macht Sinn den zweiten BRICK als permanenten Sender zu markieren, um ihn von den anderen BRICKs unterscheiden zu können. Der @ANY USB Dongle bleibt am PC eingesteckt und ist über ein Kommandozeilen Tool erreichbar. Die BRICKs werden mit 1,5 V AAA Batterien versorgt, diese können eingelegt werden.

Der @ANY USB Dongle ist über ein Kommandozeilen Tool verbunden und reagiert auf ein AT mit OK. Der BRICK, welcher permanent sendet, wird über den Schalter S3 eingeschaltet und neben den @ANY USB Dongle gelegt. Die grüne LED des @ANY BRICKs leuchtet grün auf (Standard Firmware Einstellungen). Der andere @ANY BRICK bleibt ausgeschaltet.

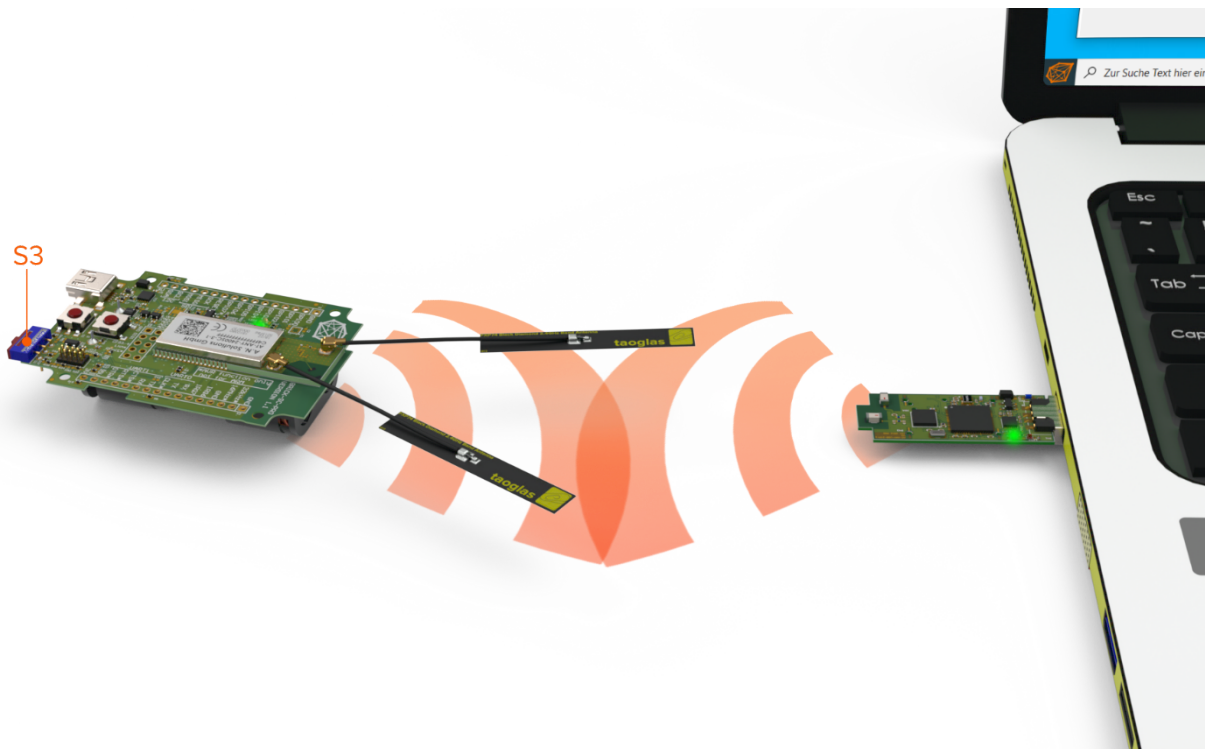


Figure 16 Mesh peer to peer\*\*

\*\* dargestellte Entfernungen entsprechen nicht der Realität

Der @ANY USB Dongle erhält Temperaturwerte von dem @ANY BRICK. Nun wird der @ANY BRICK solange vom @ANY USB Dongle entfernt, bis der @ANY USB Dongle keine Temperaturwerte mehr erhält. Ist das der Fall kommt der zweite @ANY BRICK als Knoten hinzu. Schalten Sie den @ANY BRICK über den Schalter S3 ein und legen ihn zwischen den @ANY USB Dongle und den @ANY BRICK welcher permanent sendet.



Figure 17 Dynamic Mesh\*\*

\*\*dargestellte Entfernungen entsprechen nicht der Realität

Nach wenigen Sekunden sollte der @ANY USB Dongle wieder Daten vom @ANY BRICK erhalten. Dieser geht jetzt nicht mehr den direkten Weg, sondern über den Knoten welcher im Bild oben als „link“ bezeichnet ist. Es sind viele verschiedene Netzanordnungen möglich. Ebenfalls ist es möglich, dass alle @ANY BRICKs und der @ANY USB Dongle Datenlogger sowie Knoten sind. Sollte ein Knoten ausfallen wird automatisch nach einer neuen Route zum Ziel gesucht und vermerkt. Im nächsten Kapitel wird der dritte BRICK mit der ADC-Funktion verwendet und dem Netzwerk hinzugefügt.

Herzlichen Glückwunsch Sie haben soeben Ihr erstes Mesh-Netzwerk eingerichtet und in Betrieb genommen.



## 4.8 ADC Unterstützung für SMS Pro Nutzer



Die Nutzung des Analog Digital Converter wird hier nur für den ATmega128RFA1 beschrieben und ist nur mit der Smart Mac Suite (SMS) Pro möglich. Genauere Informationen, als hier aufgeführt, entnehmen Sie bitte dem Datenblatt des ATmega128RFA1 [5]. Dieses Kapitel baut auf dem Kapitel 3.6 auf. Sollten Sie das hier beschriebene Beispiel alleinstehend durcharbeiten, kann dies womöglich zu keinem Erfolg führen. Ist das der Fall, so lesen sie das Kapitel 3.3.1, 3.4.1 und 3.6 für volle Funktionalität.

### 4.8.1 Grundeinstellungen des ADCs mit der SMS Pro

In das bestehende Mesh Netzwerk soll der letzte @ANY BRICK integriert werden. Er bekommt die Aufgabe Daten des ADC an den @ANY USB Dongle weiter zu leiten, welche dort mit Hilfe eines User Interfaces ausgewertet werden können.

Der ATmega128RFA1 besitzt einen 10 Bit ADC mit einer Vref von 1.8 V. Die Vref sowie der Vorteiler werden von der SMS vorgegeben und können nicht geändert werden. Somit liegt der darstellbare Bereich bei 0x0000 bis 0x03FF. Der ADC wird über das S-Register 238 getriggert und macht genau eine Messung. Die gewonnenen ADC Werte, werden je nach verwendetem Eingang im S-Register 240 – 247 abgelegt. Der ADC hat vier Eingangspins ADC0 bis ADC3.

#### Information zu S-Register 240 bis 247 bei Verwendung S-Register 238=3:

S-Register	247	246	245	244	243	242	241	240
	ADCL	ADCH	ADCL	ADCH	ADCL	ADCH	ADCL	ADCH
ADC Pin	3	3	2	2	1	1	0	0

Der ADC Pin 0 eignet sich nicht für eine direkte ADC Messung, weil dem Pin 0 ein Spannungsteiler zur Vcc-Messung nachgeschaltet ist. Ein kleines Beispiel im nächsten Kapitel dies verdeutlichen.

Möchten Sie den ADC zum Messen externer Spannungen nutzen, so benötigen Sie die Pins 1 bis 3. Aber beachten Sie, dass sich nur Spannungen von 0 V bis 1,8 V (0x0000 bis 0x03FF) abbilden lassen. Das Messen dieser ADC Pins geschieht auf die gleiche Art und Weise, wie im folgenden Kapitel beschrieben, nur das hier nicht die S-Register 240 und 241 ausgegeben werden, sondern nur die S-Register, welche für den jeweiligen ADC Pin benötigt werden. (Siehe dazu „Information zu S-Register 240 bis 247 bei Verwendung S-Register 238=3“)

#### Pin Leiste X3 auf @ANY BRICK:

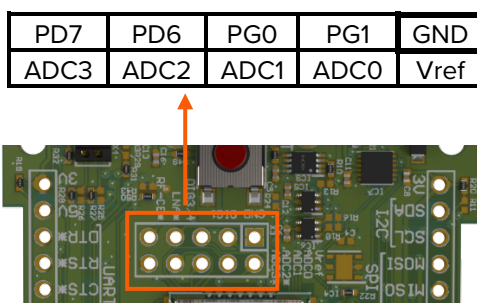
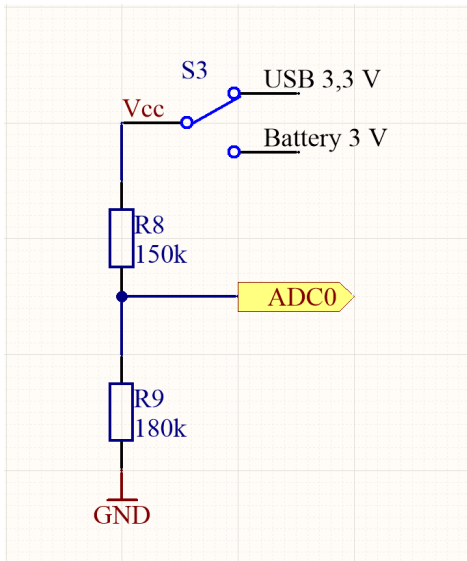


Figure 18 Section @ANY BRICK X3

## 4.8.2 Beispiel Batteriemessung mit dem ADC

### Schaltung:



Der ADC0 Pin ist über einen Spannungsteiler von 180 K $\Omega$  und 150 K $\Omega$  mit dem Schalter S3 verbunden. Über den Schalter S3 lässt sich die Vcc einstellen. Es gibt die Möglichkeit den @ANY BRICK über ein USB Anschluss zu versorgen oder mit Batterien. Die Batterie-Versorgung beträgt 3 V. Die USB Versorgung wird von 5 V auf 3,3 V herabgesetzt.

Um nun die Batterie Spannung zu bestimmen, wird der S3 auf Batterie-Versorgung umgestellt (Schalterposition rechts). Der ADC wird mit dem S-Register 238=3 getriggert und dieser legt die Rohwerte im S-Register 240-247 ab, welche dann auf die Spannung zurückgerechnet werden können.

Figure 19 Circuit ADC0

Weil der Zugriff per Kommandozeilen Tool auf die SMS nur über USB möglich ist, wird der @ANY BRICK so konfiguriert das er alle paar Sekunden den ausgelesenen Wert an unser Mesh-Netzwerk sendet. Somit kann der @ANY BRICK über die Batterien versorgt werden.

Der @ANY BRICK, aus dem vorherigen Kapitel welcher noch keine Funktion hat, wird mit USB am PC verbunden und über ein Kommandozeilen Tool angesprochen. In diesem Fall ist es der @ANY BRICK mit der Short Adresse 0003. Stellen Sie sicher, dass der @ANY BRICK sich automatisch in das Mesh-Netzwerk einwählt (S-Register 0=1). Durch ein kurzes senden von Datenpaketen mit dem Befehl **ATD** kann dies geprüft werden.

Zum Senden der ADC Werte an den @ANY USB Dongle wird das Shadow Command Register (SCR) genutzt. In das SCR wird folgendes geschrieben:

```
Vollständiger Befehl:  
AT*AS238=3D0000S240S241 ↵  
Antwort:  
OK
```

<b>AT*</b>	Schreibt die nachfolgende Kommandozeile in den SCR
<b>A</b>	Verbindet sich mit dem Netzwerk, falls dies noch nicht geschehen ist
<b>S238=3</b>	Der ADC misst einmal und legt dies im S-Register 240 bis 247 ab
<b>D0000</b>	Senden an die Adresse 0000 (@ANY USB Dongle)
<b>S240</b>	Senden des S-Registers 240 ADCH of ADC Pin 0
<b>S241</b>	Senden des S-Registers 241 ADCL of ADC Pin 0

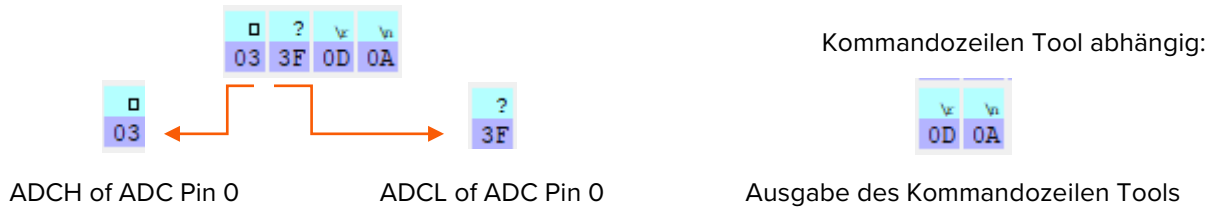
Der Timer Interrupt wird auf 1 Sekunde gesetzt und alles gespeichert.

```
Vollständiger Befehl:  
ATS18=1&W0 ↵  
Antwort:  
OK
```

Der @ANY USB Dongle sollte nun Werte empfangen. Um diese Werte besser zu verarbeiten, muss das Kommandozeilen Tool auf HEX-Ausgabe umgestellt werden. Die Batterien werden eingelegt und der Schalter S3 wird auf

die rechte Position geschaltet. Die grüne LED beginnt zu leuchten (Standard Firmware Einstellungen). Nach wenigen Sekunden bekommt der @ANY USB Dongle wieder ADC Rohdaten.

Die ankommenden ADC Rohdaten sehen wie folgt aus (Kommandozeilen Tool abhängig):



Das ergibt eine Dezimalzahl von 831. Diese kann anhand der unten genannten Formel in eine Spannung zurückgerechnet werden. Die oben ausgegebene Hexadezimalzahl ergibt eine Spannung von rund 2,7 V.

$$V_{cc} = \frac{ADCWert * V_{ref} * (R8 + R9)}{ADCmaxWert * R9}$$

Überprüfen lässt sich das Ganze in dem die Vcc auf USB geschaltet wird (S3 Schalterstellung links) und der gewonnene ADC Wert wieder, mit der Formel von oben, in eine Spannung zurückgerechnet wird. Dabei sollte sich eine Spannung von rund 3,3 V errechnen lassen.

## 5 Firmware flashen/updates

### 5.1 Struktur der @ANY Hardware

Die @ANY Module gibt es in verschiedenen Varianten, entweder als eigenständiges Module oder aufgelötet auf einem fertigen Entwicklungsboard. Die @ANY Module bestehen hauptsächlich aus zwei Teilen, dem Mikrocontroller und einem Funkmodul. An den Modulpins stehen mehrere vom Mikrocontroller bereitgestellte Schnittstellen zur Verfügung.

Als Mikrocontroller wird je nach Modul der Atmel ATmega1281, der Atmel ATmega128RFA1 oder der SAMR30G18 verwendet. Diese Mikrocontroller sind mit einem vom Benutzer austauschbaren Softwareteil ausgestattet, mit der sogenannten Firmware, welche die gewünschte Funktionalität des Gerätes implementiert. Die Software kann über zwei verschiedene Arten auf den Mikrocontroller programmiert werden, welche im nachfolgenden beschrieben werden.

#### @ANY900 Modules:

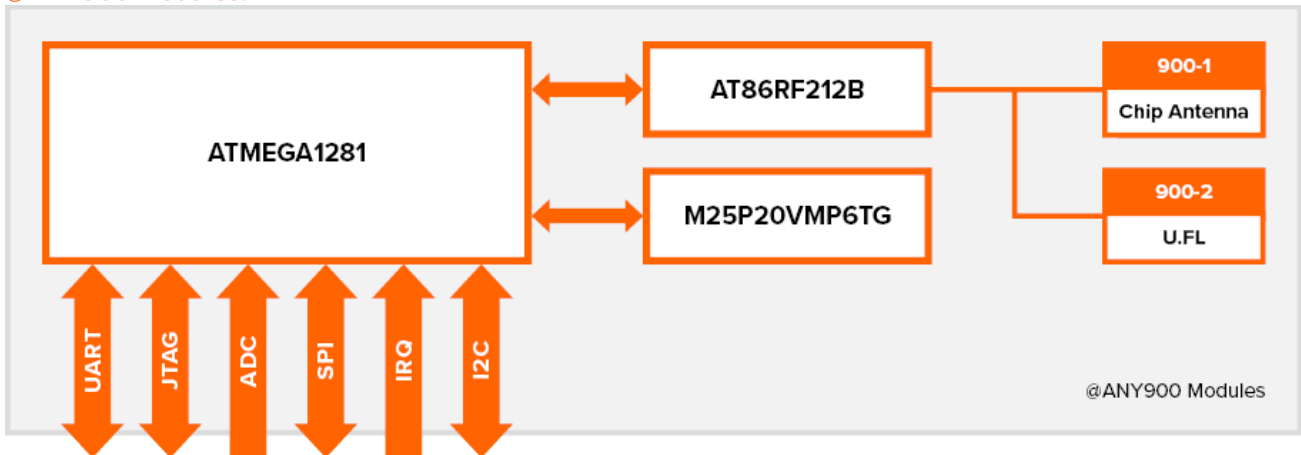


Figure 20 @ANY900 Modules

**@ANY900ARM-SC Modules:**

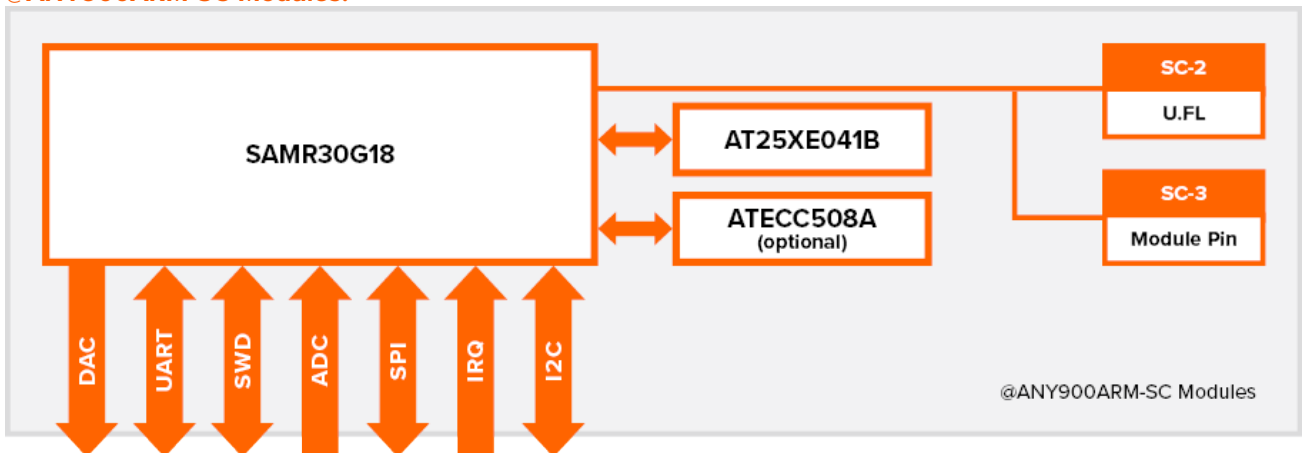
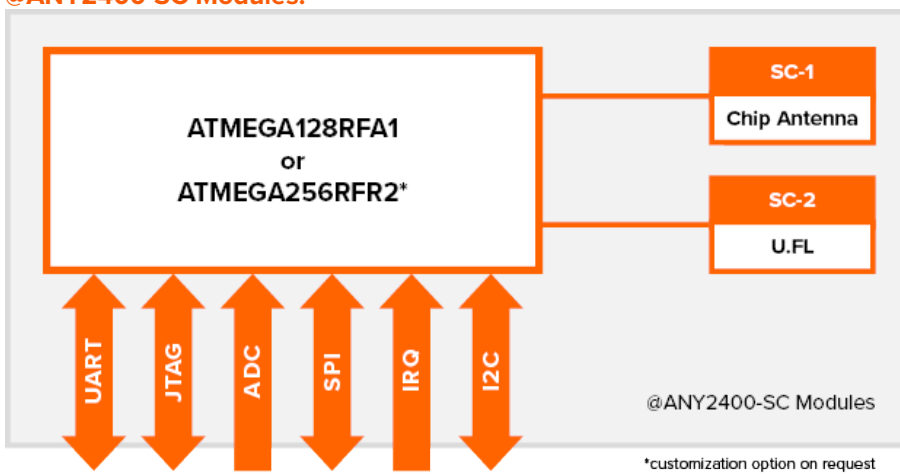


Figure 21 @ANY900ARM-SC Modules

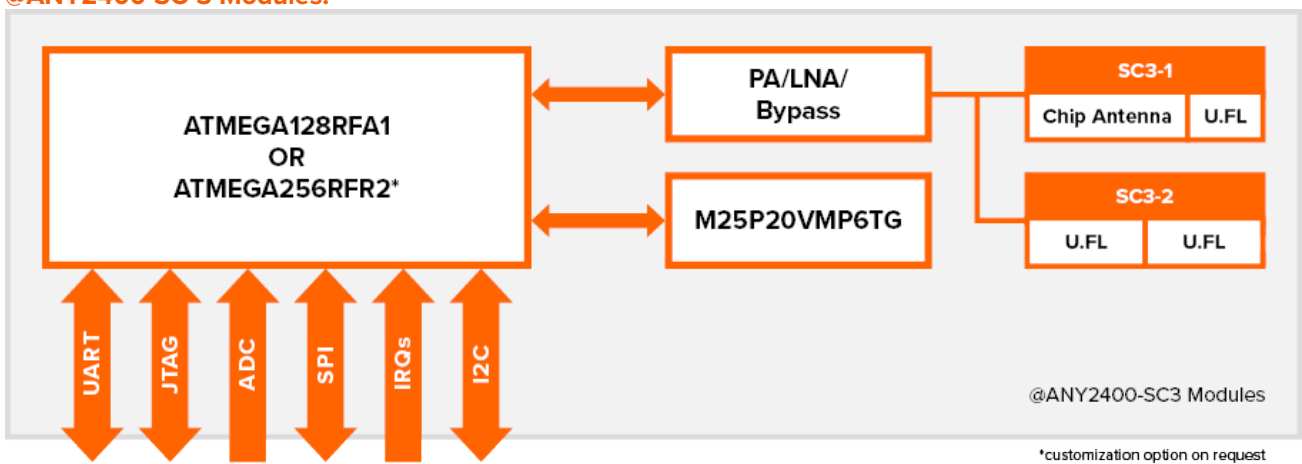
**@ANY2400-SC Modules:**



\*customization option on request

Figure 22 @ANY2400-SC Modules

**@ANY2400-SC-3 Modules:**



\*customization option on request

Figure 23 @ANY2400-SC-3 Modules

## 5.2 JTAG

Eine weit verbreitete Art der Programmierung eines Mikrocontroller ist die Nutzung einer bereitgestellten JTAG/SWD Schnittstelle. Alle angesprochenen Geräte in diesem Dokument sind mit solch einer Schnittstelle ausgestattet. Um die JTAG/SWD Schnittstelle nutzen zu können, ist eine zusätzliche Programmierhardware-Schnittstelle erforderlich. Die Auswahl an möglicher Hardware ist groß und reicht von selbstgebaute Geräten bis hin zu professioneller Hardware, welche viele Funktionen unterstützen weit über die Geräteprogrammierung hinaus. Ein Beispiel für eine professionelle Lösung ist der Atmel ICE (Abbildung 24) welcher kostenpflichtig bei Microchip erworben werden kann.



Figure 24 Atmel ICE Kit

Mit dem Atmel ICE ist es möglich alle unsere Geräte, welche einen 50mil 10Pin JTAG/SWD-Stecker besitzen, zu programmieren. Zur Programmierung des @ANY USB Dongles wird ein Adapter benötigt welcher im Kapitel 1 Übersicht aufgeführt ist. Die Verwendung der JTAG/SWD-Schnittstelle bietet größtmögliche Flexibilität und wird Power-Usern und Embedded-Entwicklern empfohlen, ist allerdings mit einem Kostenaufwand verbunden. Ein kosteneffizienterer Ansatz wird im nächsten Unterabschnitt vorgestellt.

## 5.3 Bootloader

Eine weitere Möglichkeit, die Firmware des @ANY-Modules zu updaten, ist die Verwendung eines Bootloaders. Für die Verwendung müssen zwei Voraussetzungen erfüllt sein. Erstens muss sich ein Code mit Bootloader-Funktionalität in einem bestimmten Bereich des Flash-Speichers im Mikrocontrollers befinden. Zweitens muss die Anwendungssoftware eine Möglichkeit bieten in den Bootloader-Code zu springen, oder die Fuse-Bite des Mikrocontrollers müssen so eingestellt werden, dass der Bootloader-Code bei jedem Einschalten ausgeführt wird. Die frei verfügbare Software Smart MAC Suite (SMS) von A.N. Solutions enthält den erforderlichen Bootloader-Code und bietet eine Möglichkeit, von der Anwendung aus in diesen Code zu springen, in dem ein vordefinierter Satz von AT-Befehlen gesendet wird. Die SMS-Software mit Bootloader-Funktion ist in den Geräten des Development-Kits vorprogrammiert.

Für benutzerdefinierte Lösungen muss hier auf das @ANY BRICK Datenblatt verwiesen werden, welches die Pinbelegung und Anschlussbelegung enthalten.

## 5.4 Firmware flashen/updaten mit dem SMS Updater

Der SMS Updater ist ein kostenlos von A.N. Solutions bereitgestelltes Tool, welches es ermöglicht die Firmware eines @ANY Moduls mit *vorinstallierter Bootloader-SMS* zu upgraden oder zu flashen. Er ermöglicht das flashen und updaten der SMS ohne JTAG/SWD Schnittstelle nur über USB. Voraussetzung dafür ist die korrekte Erkennung der @ANY Module am verwendeten PC, siehe dazu das Kapitel 2.

Der @ANY USB Dongle oder der @ANY BRICK wird per USB mit dem PC verbunden. Im Geräte-Manager taucht ein neuer COM Port auf. Der ATANY-Updater wird geöffnet. Stellen sie sicher das, dass Tool auf der neusten Version ist. Weiter Informationen dazu auf unserer Webseite.



Im Bereich Interface wird der COM gewählt welcher dem @ANY USB Dongle oder @ANY BRICK zugewiesen wurde. Taucht dieser in der Liste nicht auf, kann mit der Schaltfläche „**Refresh port list**“ die Liste aktualisiert werden. Taucht der COM immer noch nicht auf, springen Sie in das Kapitel 2. Die Baud ist auf 38400bd einzustellen und im Feld „**AT-ANY module type**“ ist auto zu belassen. Nun kann im Bereich „**Flash file**“ über open die gewünschte Firmware gewählt werden. Es ist wichtig, dass die zu flashende Firmware keinen Bootloader Bereich besitzt, welches ohne \_BL im Namen gekennzeichnet ist. Wird die Firmware mit Bootloader geflasht, überschreibt dieser den vorhandenen Bootloader. Ist der aktuelle Bootloader überschrieben worden, kann die Firmware nur noch mit dem Kapitel 5.6 oder 5.7 zurück geflasht werden.

Wurde die richtige Firmware gewählt, kann mit „**Start Update**“ der Flashvorgang gestartet werden. Dieser läuft im mittleren Teil des ATANY-Updater durch und kann ein paar Sekunden dauern. Der Flashvorgang wird mit „**programming done**“ erfolgreich abgeschlossen. Dies ist im folgenden Bild dargestellt.

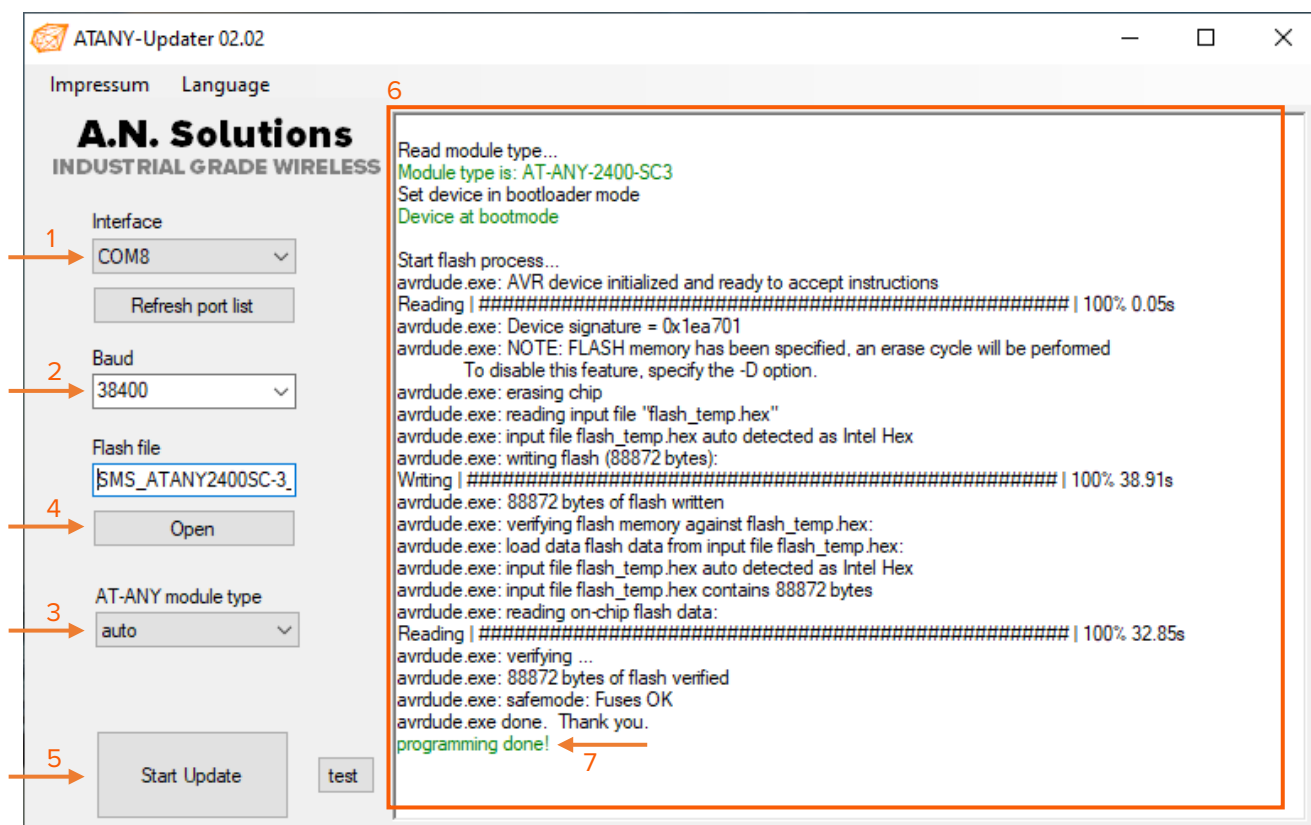


Figure 25 ATANY-Updater

## 5.5 Firmware flashen/updaten über AVRDUde ohne JTAG/SWD

AVRDUde ist eine Free Software welche es ermöglicht über Kommandozeilenparameter Programmcode in den Flash-Speicher vom Atmel AVR Controller zu schreiben.

Zum Flashen das @ANY USB Dongle oder des @ANY BRICK werden diese mit einen freien USB Port verbunden. Der Geräte Manager initialisiert die Geräte und weist ihnen einen COM zum. Ist das nicht der Fall sein, springen sie zum Kapitel 2.

Um über AVRDUde die Firmware zu flashen muss sich ein Bootloader auf dem @ANY USB Dongle und @ANY BRICK befinden. Es werden zwei Konsolen für das Flashen benötigt. Als erstes wird eine Verbindung mit dem @ANY USB Dongle oder @ANY BRICK über ein Kommandozeilen Tool hergestellt. Die bestehende Verbindung wird durch ein AT geprüft und der @ANY USB Dongle oder der @ANY BRICK antwortet mit OK. Die Verbindung bleibt bestehen.



Nun wird eine weitere Konsole geöffnet. In Windows geschieht das durch Drücken der Tastenkombination „**Windows-Logo-Taste + R**“. Es öffnet sich ein Eingabefeld in welches „**cmd**“ eingegeben wird und mit ENTER bestätigt. Die Windows-Konsole ist nun geöffnet. In dieser wird mit Hilfe von „**cd**“ in den Ordner navigiert, in welchem sich die „**AVRDude.exe**“ befindet. Mit der Eingabe von „**AVRDude.exe**“ wird AVRdude ausgeführt. Es ist in dem Ordner zu verbleiben.

Zum Flashen der Firmware muss der @ANY USB Dongle oder @ANY BRICK in den Bootloader springen. Dies wird mit dem Befehl **AT+U** realisiert. Ist dies geschehen, verbleiben 20 Sekunden um die benötigten Kommandozeilenparameter in AVRdude auszuführen. Sind die 20 Sekunden verstrichen springt der @ANY USB Dongle oder der @ANY BRICK zurück in den normalen Programmcode. Es ist sinnvoll sich die benötigte Kommandozeile im AVRdude zurecht zu legen. Im Folgenden wird der Ablauf erklärt.

Kommandozeilen Tool	
<b>AT</b> ↵	Verbindung prüfen
OK	
<b>ATS18=0</b> ↵	Timer S-Register auf 0 setzen
OK	
<b>ATH1</b> ↵	Alle bestehenden Verbindungen trennen
OK	
<b>ATS94=224</b> ↵	Bootloader im S-Register 94 freischalten
OK	
<b>AT+U</b> ↵ → 20 sec Timer läuft	In den Bootloader springen
<b>Terminal Verbindung trennen!</b>	

AVRDude.exe in der Windows Konsole	
<b>AVRDude -p m128RFA1 -c stk500v1 -P\\.\COM8 -b 38400 -U fl:w:SMS_ATANY2400-3_131b_0BB9.hex</b> ↵	
-p	Partnumber / Required. Specify AVR device.
-c	Programmer / Specify programmer type.
-P	Port / Specify connection port.
-b	Baudrate / Override RS-232 baud rate.
-U	<memtype>:write:<filename> / Memory operation specification. Multiple -U options are allowed, each request is performed in the order specified.

Die oben aufgeführt Kommandozeile muss in AVRdude vor dem ausführen des **AT+U** Befehls zurechtgelegt werden, damit diese in den 20 Sekunden ausgeführt werden kann. Nach dem Auslösen der Kommandozeile in AVRdude, schreibt er den Programmcode in den Flash-Speicher des @ANY USB Dongle oder des @ANY BRICK. Das flashen ist mit „**AVRDude done. Thank you.**“ Erfolgreich abgeschlossen.

So ist es möglich auch ohne extra Hardware die Firmware auf dem @ANY USB Dongle und auf dem @ANY BRICK aktuell zu halten. Dies gestaltet sich über den ATANY Updater in Kapitel 5.4 deutlich einfacher.

## 5.6 Firmware flashen/updaten über AVRdude mit JTAG/SWD

Ist ein Atmel ICE Kit oder ähnliche Programmierhardware vorhanden kann dies mit AVRdude genutzt werden. Hierzu muss aber beachtet werden, dass der Atmel ICE erst ab der Version 6.3 von AVRdude unterstützt wird. Außerdem wird für das Flashen des @ANY USB Dongles ein spezial Connector (Kapitel 1 Übersicht) benötigt, weil dieser keinen 10 Pin 50 mil Stecker besitzt.

*Bei dieser Variante muss sich kein Bootloader auf dem @ANY USB Dongle oder dem @ANY BRICK befinden.*

Der @ANY BRICK und der @ANY USB Dongle werden auf zwei unterschiedliche Arten mit dem Atmel ICE und dem PC verbunden.

Der @ANY BRICK wird per USB an einen freien USB Port am PC gesteckt. Dies dient der Stromversorgung. Alternativ kann der @ANY BRICK auch mit zwei 1,5 V AAA Batterien versorgt werden. Das Atmel ICE Kit wird ebenfalls



per USB an einen freien USB Port am PC gesteckt. Der mitgelieferte 10 Pin 50 mil Stecker wird im Atmel ICE auf die AVR Seite oder für SAMR30G18 Module in die SAM Seite gesteckt und am @ANY BRICK auf den vorhandenen 10 Pin 50 mil Stecker. Dabei ist auf die richtige Ausrichtung des Steckers zu achten!

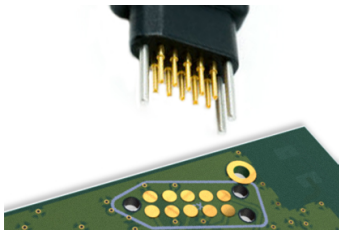


Figure 26 Tag Connect

Mit dem @ANY USB Dongle ist die Verbindung ziemlich umständlich. Der @ANY USB Dongle muss zur Stromversorgung über USB an den PC angeschlossen werden. Es ist vorteilhaft, eine USB-Verlängerung zu verwenden, da sich der Tag Connect (Abbildung 26) auf der Rückseite des @ANY USB Dongle befindet. Als nächstes wird das Tag Connect Kabel mit dem Tag Connect auf der Rückseite des Dongle verbunden. Stellen Sie sicher, dass das Tag Connect immer in Kontakt mit dem Kabel ist. Je nach verwendetem DevKit muss das andere Ende des Tag-Connect-Kabels an den AVR-Steckplatz oder bei ARM an den SAM-Steckplatz des Atmel ICE angeschlossen werden.

Nun wird eine Konsole geöffnet. In Windows geschieht das durch Drücken der Tastenkombination „Windows-Logo-Taste + R“. Es öffnet sich ein Eingabefeld in welches „cmd“ eingegeben wird und mit ENTER bestätigt. Die Windows-Konsole ist nun geöffnet sein. In dieser wird mit Hilfe von „cd“ in den Ordner navigiert, in welchem sich die „AVRDude.exe“ befindet. Mit der Eingabe von „AVRDude.exe“ wird AVRDude ausgeführt. Es ist in dem Ordner zu verbleiben.

Das flashen der Firmware erfolgt nicht wie im Kapitel 5.5 beschrieben, weil hier kein Bootloader benötigt wird. Allerdings muss der Programmierer und der Port im AVRDude geändert werden. Das Flashen des Dongles sowie des BRICKs sind gleich, nur die Anschlussarten unterscheiden sich.

Die folgende Kommandozeile wird mit Hilfe von AVRDude ausgeführt.

AVRDude.exe in der Windows Konsole

```
AVRDude -p m128RFA1 -c atmelice -P usb -b 38400 -U fl:w:SMS_ATANY2400-3_131b_OBB9.hex ↵
```

```
-p Partnumber / Required. Specify AVR device.  
-c Programmer / Specify programmer type.  
-P Port / Specify connection port.  
-b Baudrate / Override RS-232 baud rate.  
-U <memtype>:write:<filename> / Memory operation specification. Multiple -U options are allowed, each request is performed in the order specified.
```

Nach dem Auslösen der Kommandozeile in AVRDude, schreibt er den Programmcode in den Flash-Speicher des @ANY USB Dongle oder des @ANY BRICK. Das Flashen ist mit „**AVRDude done. Thank you.**“ Erfolgreich abgeschlossen. Mit Hilfe des Atmel ICE erfolgt das Flashen deutlich schneller.

Bei Verwendung dieser Methode wird der komplette Flash-Speicher überschrieben und es ist später nicht mehr möglich mit dem Bootloader zu arbeiten.




## 5.7 Firmware flashen/updaten mit JTAG/SWD & Atmel Studio

Eine weitere Möglichkeit die Firmware zu flashen oder upzudaten bietet das Atmel Studio von Microchip. Dafür wird ebenfalls das Atmel ICE Kit benötigt und für den @ANY USB Dongle der Spezial Connector (Kapitel 1 Übersicht).

Bei dieser Variante muss sich kein Bootloader auf dem @ANY USB Dongle oder dem @ANY BRICK befinden.

Der @ANY BRICK wird per USB an einen freien USB Port am PC gesteckt. Dies dient der Stromversorgung. Alternativ kann der @ANY BRICK auch mit zwei 1,5 V AAA Batterien versorgt werden. Das Atmel ICE Kit wird ebenfalls per USB an einen freien USB Port am PC gesteckt. Der mitgelieferte 10 Pin 50 mil Stecker wird im Atmel ICE auf die AVR Seite oder für SAMR30G18 Module in die SAM Seite gesteckt und am @ANY BRICK auf den vorhandenen 10 Pin 50 mil Stecker. Dabei ist auf die richtige Ausrichtung des Steckers zu achten!

Mit dem @ANY USB Dongle ist die Verbindung ziemlich umständlich. Der @ANY USB Dongle muss zur Stromversorgung über USB an den PC angeschlossen werden. Es ist vorteilhaft, eine USB-Verlängerung zu verwenden, da sich der Tag Connect (Abbildung 26) auf der Rückseite des @ANY USB Dongle befindet. Als nächstes wird das Tag Connect Kabel mit dem Tag Connect auf der Rückseite des Dongles verbunden. Stellen Sie sicher, dass das Tag Connect immer in Kontakt mit dem Kabel ist. Je nach verwendetem DevKit muss das andere Ende des Tag-Connect-Kabels an den AVR-Steckplatz oder bei ARM an den SAM-Steckplatz des Atmel ICE angeschlossen werden.

Das Atmel Studio wird bei Microchip heruntergeladen und installiert. Ist Atmel Studio gestartet, öffnet sich drücken der  (**Ctrl+Shift+P**) Schaltfläche ein „Device Programming“ Fenster. Der Kopf dieses Fensters sieht wie folgt aus:

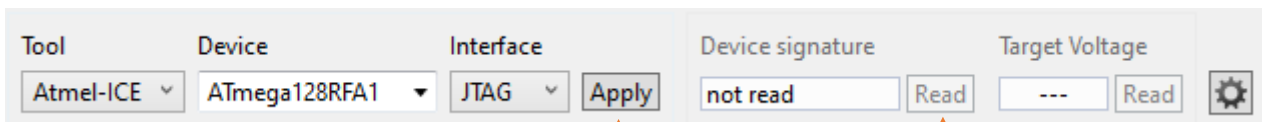


Figure 27 Device programming head

Die Einstellungen werden mit **Apply** bestätigt. Durch Drücken der Schaltfläche **Read**, liest Atmel Studio die „Device signature“ aus und die Verbindung zum Gerät wurde geprüft. Am linken Fensterrand wird in den Reiter **Memories** gewechselt. Im Bereich **Flash** kann durch Drücken der Schaltfläche mit den drei Punkten eine Firmware geladen werden. Mit der Schaltfläche **Programm** wird die Firmware in den Flash-Speicher geschrieben.

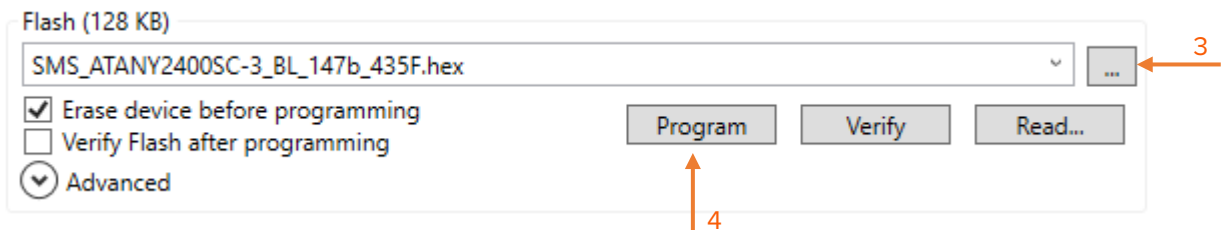


Figure 28 Memories - Flash

Das erfolgreiche Flashen wird mit einem „Programming Flash...Ok“ zurückgegeben.

Bei Verwendung dieser Methode wird der komplette Flash-Speicher überschrieben und es ist später nicht mehr möglich mit dem Bootloader zu arbeiten.

## 6 Software Tools

### 6.1 Sniffer Tool

In der Entwicklungsphase eines Projektes kann es von großem Nutzen sein, jedes gesendete Datenpaket zu analysieren und zu verfolgen. Mit Hilfe der Sniffer Firmware ist es möglich genaue Netzwerkanalysen und zielgerichtetes Debugging durch zu führen.

Als Sniffer wird ein leistungsstarkes Gerät gewählt, mit sehr guten Antennen. In diesem Beispiel wird ein Peer to Peer Netzwerk zwischen einem @ANY USB Dongle und einem @ANY BRICK beobachtet. Ein zweiter @ANY BRICK kommt als Sniffer zum Einsatz.

Der erste Schritt besteht darin ein funktionsfähiges *Peer to Peer* Netzwerk zwischen dem @ANY USB Dongle und einem der zwei @ANY BRICK aufzubauen. In diesem Netzwerk muss es möglich sein, Daten mittels **ATD** Befehl auszutauschen. Hierzu siehe Kapitel 4 „Getting Started“.

Als nächstes wird mit einer der im Kapitel 5 beschriebenen Methoden der andere @ANY BRICK mit der Firmware HalCtrlXXXX\_vXX.hex geflasht. Bei der Auswahl der Firmware ist auf die Kompatibilität mit der einzelnen Hardware zu achten, welche im Namen der Firmware nach „HalCtrl“ angegeben wird.

Mit einem geeigneten Kommandozeilen Tool wird eine Verbindung mit dem @ANY BRICK aufgebaut, welcher die Sniffer Firmware enthält. Dabei ist zu beachten, dass die Baudrate auf 57600 geändert wird. (COMX 57600bd 8 N 1)

Mit dem Befehl **AT?** kann eine Befehlsübersicht aufgerufen werden. Die Übersicht ist nicht so umfangreich wie bei der SMS. Als nächstes wird er Channel mit dem Befehl **AT+C=XX** auf den Channel des Peer to Peer Netzwerkes geändert. Durch Eingabe des Befehls **ATA** startet der BRICK das Sniffing.

```
AT?
HalCtrl12400SC commands:
-----
AT? ..... this
ATB ..... jump to bootloader at 0x1F000
ATZ ..... reset Transceiver
ATA ..... enable RX
ATH ..... disable RX
AT+A ..... enable Transceiver
AT+H1 ..... disable Transceiver
AT+H0 ..... MCU sleep
ATS<r>(?!=<v>) .. read/write Transceiver register
ATD=<data> ..... send ascii
ATDX ..... endless TX (toggle)
AT+C(?!=<ch>) ... read/write channel
ATL<n>=(0/1) ... LED ctrl
AT+F(?!=<f>) ... data output format (RB/RH/FB/FH)
received frame format: +timestamp: ctl sq data crc (rssi)
OK

AT+C=20
OK

ATA
RX enabled
```

Figure 29 Sniffer config

Durch senden mittels **ATD** Befehl, im Peer to Peer Netzwerk, hört der Sniffer das „Gespräch“ mit und bekommt verschiedene Informationen übermittelt. Dies kann wie folgt aussehen:

Koordinator:	End-Gerät
	<b>ATD;</b> ↵
	<b>HALLO</b> ↵
	ID 001: 5 BYTES to 0000
	OK
+ DATA: 5 BYTES FROM 0001 (074) HALLO	+ SENT: ID 001

Sniffer:
+003A.DB47: 8861 9C AB 12 00 00 01 00 48 61 6C 6C 6F 0984 (0A)
+003A.DB4C: 0002 9C EB5D (0B)

Diese Informationen schlüsseln sich wie folgt auf:

Sender:	
+003A.DB47:	Die Zeit ab wann gesendet wurde
8861	Konfiguration-Bits des IEEE 802.15.4 Standard, können dort aufgeschlüsselt werden
9C	Laufende Nummer, dient der Zuordnung von Antworten
AB 12	PANID (Little-Endian-Format*)
00 00	Empfänger (Little-Endian-Format*)
01 00	Sender (Little-Endian-Format*)
48 61 6C 6C 6F	Gesendete Daten Hallo
0984	CRC Checksumme (! = Fehler in der Checksum, Übertragung fehlgeschlagen)
(0A)	Signalstärken-Indikator (RSSI) **
Empfänger:	
+003A.DB4C:	Die Zeit ab wann empfangen wurde
0002	Konfiguration-Bits als Acknowledgement, gesendet als Broadcast
9C	Laufende Nummer zur Zuordnung beim Sender
EB5D	CRC Checksumme (! = Fehler in der Checksum, Übertragung fehlgeschlagen)
(0B)	Signalstärken-Indikator (RSSI) **

\* *Das Little Endian Format ist ein Format für die Übertragung oder Speicherung binärer Daten, in denen das Least Significant Byte (LSB) an erster Stelle kommt und auf der niedrigsten Speicheradresse gespeichert wird.*

\*\* Indicator SMS ED=3\*(RSSI-1)

Ein sinnvolles Anwendungsbeispiel ist die Messung der Paketfehlerrate (PER) auf der Ebene der Hardwareabstraktionsschicht (HAL). Im Gegensatz zu den softwarebasierten Smart MAC Suite-Anwendungen enthält der HAL-basierte Ansatz keine CSMA-Ca, Acknowledgements und Neuübertragungen für eine zuverlässigere Verbindung. Das macht diese Anwendung ideal für Entfernungsmessungen und Robustheitsprüfungen unter verschiedenen Bedingungen. Die Anwendung unterstützt eine Punkt-zu-Punkt-Übertragung, wobei ein Gerät als Sender und das andere als Empfänger fungiert. Die kontinuierliche Paketübertragung kann mit dem Befehl **ATDX** gestartet und gestoppt werden. Der Empfang kann mit dem Befehl **ATA** gestartet werden und wird nach einer definierten Zeit ohne Paketempfang oder nach dem Befehl **ATH** automatisch gestoppt. Die Firmware mit dem Dateinamen "autoTX" führen nach dem Einschalten automatisch den Befehl **ATDX** aus und sendet somit permanente Pakete. Sie eignen sich daher gut für autonome Absender ohne UART-Schnittstelle, z. B. einige BRICK-Boards.



Für den Testaufbau werden zwei @ANY BRICKs verwendet. Der erste wird mit der Firmware Hal-CtrlXXXX\_PER\_test.hex, wie im Kapitel 5 beschrieben, geflasht. Der zweite wird mit der Firmware Hal-CtrlXXXX\_PER\_test\_autoTX.hex beschrieben. Es ist sinnvoll den zweiten BRICK mit 1,5V AAA Batterien zu versorgen um die Mobilität zu erhöhen. Es wird eine Verbindung über ein Kommandozeilen Tool mit dem ersten @ANY BRICK hergestellt und der zweite eingeschaltet. Der zweite @ANY BRICK beginnt direkt mit der Übertragung und der erste empfängt die Daten.

## 6.2 Transparent UART – SMS PRO Nutzer



Dieses Kapitel bezieht sich auf Verfahrensweisen aus dem Kapitel 5, Firmware flashen / updaten. Es werden erweiterte Kenntnisse vorausgesetzt. Diese Verfahrensweisen dienen als Auswahl der Möglichkeiten und lassen sich nicht 1 zu 1 auf das Folgende abbilden. Außerdem kann es sein, dass Zusatz Hard- sowie Software benötigt wird.

Der transparent UART ermöglicht es Ihnen, ohne großen Aufwand Daten zwischen zwei Geräten auszutauschen. Hierbei entsteht ein einfacher bidirektionaler Punkt zu Punkt Datenkanal, mithilfe unserer AT-ANY Module. Die Software wurde eigens dafür abgestimmt und erleichtert so die Einrichtung einer Punkt zu Punkt Verbindung, zwischen zwei Funk Modulen. Anhand des folgenden Beispiels wird dies verdeutlicht.

Derzeit werden folgende AVR Module unterstützt:

- AT-ANY 900-1
- AT-ANY 900-2
- AT-ANY 2400-SC-1
- AT-ANY 2400-SC-2
- AT-ANY 2400-SC-3-1
- AT-ANY 2400-SC-3-2

Zunächst wird die abgestimmte Firmware benötigt. Nach dem Erwerb der @ANY Smart Mac Suite (SMS) Pro, haben Sie von uns ein Softwarepaket erhalten (<https://an-solutions.de/de/any-sms-pro.html>). Neben der SMS Pro befindet sich ein Ordner „Transparent UART“, worin alle wichtigen Files für den transparent UART enthalten sind. Entpacken Sie diesen Ordner an einem Ort Ihrer Wahl auf Ihrem PC.

In diesem Ordner, befinden sich mehrere unterschiedliche HEX Dateien. Dateien, welche mit „eeprom“ beginnen, sind für die Einrichtung des EEPROMs und dessen Einstellungen vorgesehen. Dateien, welche mit „transparent\_uart“ beginnen, sind abgestimmte Firmwarepakete für die AT-ANY Module. Die nachfolgenden Bezeichnungen beziehen sich auf die verwendeten BRICKs und Dongles.

Wählen Sie zwei Ihrer vorliegenden Geräte aus, welche mit dem transparent UART genutzt werden sollen.

Hier im Beispiel werden zwei AT-ANY 2400-SC-3-2 BRICKs verwendet. Die Einrichtung und der Ablauf sind auf jedem Gerät von A.N. Solutions identisch. Zum Flashen der Firmware und des EEPROMs wurde Atmel Studio 7.0 von Microchip mit dem Atmel ICE (Seite 29, Abbildung 24) als Programmierschnittstelle verwendet.

Anhand Ihrer Auswahl werden die richtigen EEPROM-Hex Files benötigt. Achten Sie darauf, dass die Bezeichnung der Hex-Files mit den Ihrer gewählten Geräte übereinstimmt. Für die Einrichtung des EEPROMs werden zwei Files benötigt, „... 1 to 2 ...“ und „... 2 to 1 ...“.

Die hier für das Beispiel benötigten Files sind folgende: *AT-ANY 2400-SC-3-2 BRICKs*

- eeprom\_2400\_ff\_0001\_to\_0002\_XX\_YY.hex\*
- eeprom\_2400\_ff\_0002\_to\_0001\_XX\_YY.hex\*

\* XX – Version, YY – Channel



Die Auswahl der Firmware ist ähnlich. Achten Sie auf die Bezeichnung „transparent\_uart“ und den verwendeten Gerätetyp.

Die hier im Beispiel benötigten Firmware ist folgende: *AT-ANY 2400-SC-3-2 BRICKS*

- transparent\_uart\_2400sc-3.hex

Als Nächstes werden mit einer der im Kapitel 5 aufgeführten Verfahrensweisen die Geräte geflasht. Die dort aufgeführten Verfahrensweisen dienen als Auswahl der Möglichkeiten und lassen sich nicht 1 zu 1 auf das Folgende abbilden. Gehen Sie wie folgt vor:

1. anschließen eines der beiden Geräte an den PC per mini USB (Kapitel 2)
2. 2.4 GHz - ändern der Fuses auf *hfuse: 0x19, lfuse: 0xde und efuse: 0xfe*  
Sub-1 GHz - ändern der Fuses auf *hfuse: 0x19, lfuse: 0xe2 und efuse: 0xfe*
3. Im Atmel Studio muss zuerst ein manueller Erase des Chips erfolgen
4. flashen Sie die benötigte EEPROM-Datei auf das Gerät „*eeeprom\_...\_0001\_to\_0002\_...hex*“
5. flashen Sie die richtige Firmware „*transparent\_uart\_...hex*“ auf das Gerät
6. 2.4 GHz - ändern der Fuses auf *hfuse: 0x91, lfuse: 0x46, efuse: 0xfe*  
Sub-1 GHz - ändern der Fuses auf *hfuse: 0x91, lfuse: 0xe2 und efuse: 0xfe*
7. anschließen des zweiten Gerätes per mini USB (Kapitel 2)
8. 2.4 GHz - ändern der Fuses auf *hfuse: 0x19, lfuse: 0xde und efuse: 0xfe*  
Sub-1 GHz - ändern der Fuses auf *hfuse: 0x19, lfuse: 0xe2 und efuse: 0xfe*
9. Im Atmel Studio muss zuerst ein manueller Erase des Chips erfolgen
10. flashen Sie die benötigte EEPROM-Datei auf das Gerät „*eeeprom\_...\_0002\_to\_0001\_...hex*“
11. flashen Sie die richtige Firmware „*transparent\_uart\_...hex*“ auf das Gerät
12. 2.4 GHz - ändern der Fuses auf *hfuse: 0x91, lfuse: 0x46, efuse: 0xfe*  
Sub-1 GHz - ändern der Fuses auf *hfuse: 0x91, lfuse: 0xe2 und efuse: 0xfe*

Nach dem erfolgreichen Flashen der Geräte sind diese einsatzbereit. Schließen Sie beide Geräte per mini USB an Ihren PC an und öffnen Sie zwei Kommandozeilentools, wir empfehlen die Verwendung von HTerm. Sorgen Sie für ungehinderten Funkempfang. Die angeschlossenen AT-ANY BRICKS / Dongles werden im Geräte Manager als Serial Port aufgeführt.

In Ihrem Kommandozeilentool, kann sich nun mit den zugewiesenen Serial Ports, einer Baudrate von 9600 Baud, 8 Datenbits, einem Stoppsbit und keiner Parität verbunden werden.

Sobald beide Geräte mit jeweils einem Kommandozeilentool verbunden sind, können über die Kommandozeile Zeichen ausgetauscht werden. Diese werden direkt nach dem Senden auf dem anderen Gerät empfangen und ausgegeben. Das ist in beide Richtungen möglich. Somit lassen bestehende Geräte sinnvoll erweitern und umständliche Kabelkommunikation vermeiden.

Herzlichen Glückwunsch, ihr transparent UART ist einsatzbereit.



## Referenz Dokumente

- [1] SMS Command Reference [www.an-solutions.de/SMS\\_Command\\_Reference.html](http://www.an-solutions.de/SMS_Command_Reference.html)
- [2] 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)
- [3] @ANY900 / @ANY900 / @ANY2400 Module Product data sheets (<https://an-solutions.de/de/produkte.html>)
- [4] ITU-T V.250: Serial asynchronous automatic dialling and control

## Änderungsübersicht

Version	Änderungen	Datum
1	Erstellung Grunddokument	20.02.20
2	Review Engineering Bereich und Ergänzungen	24.04.20
3	Freigabe zur Veröffentlichung	25.05.20
4	Kapitel 6.2 Transparent UART hinzugefügt	15.06.20

## 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](mailto: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)

