

Telegesis		TG-R2xx to R3xx Migration Guide
Migration from firmware R2xx to R3xx		Firmware Migration Guide 1.02

TG-R2xx to R3xx Migration

ETRX2 ZigBee® MODULES

Migration Guide



Table of Contents

1	INTRODUCTION	3
1.1	A Note on Interoperability and ZigBee® Compliance.....	3
1.2	Firmware Customization.....	4
1.3	Updated command-line interaction.....	4
2	NETWORK SETUP	5
2.1	Starting a PAN	5
2.2	Joining a PAN	6
2.3	Managing a PAN.....	7
3	MESSAGING	8
4	S-REGISTERS	9
5	SINK HANDLING	10
6	SOURCE ROUTING	11
7	BUILD IN FUNCTIONALITY	13
8	END DEVICE HANDLING	13
9	MISCELLANEOUS NEW FEATURES	14
9.1	Changes between R301 and R302	14
10	SECURITY	15
10.1	Join Decisions.....	16
10.2	Changing Network keys	17
11	INTERACTION WITH OTHER ZIGBEE PRO DEVICES	18
11.1	Features of R302 and later versions.....	19
12	GLOSSARY	20
13	APPENDIX A: AT COMMAND COMPARISON	21
14	APPENDIX B: S-REGISTER COMPARISON	23
15	APPENDIX C: BUILT IN FUNCTIONALITY COMPARISON	26

1 Introduction

This document is intended for customers who are familiar with the R2xx versions of the Telegesis AT-Commandset based on EmberZNet2.5 and want to upgrade to the ZigBee PRO compliant R3xx AT-Commandset based on EmberZNet 4.3.

The new ZigBee PRO based firmware offers a number of new and exciting features like:

- Messages can now travel up to 30 hops
- Networks can have more than a single sink
- Source-Routing
- Improved security including Trust-Centre link keys
- Support for up to 4 external interrupts and 4 analogue inputs
- Nodes can be addressed by their EUI as well as their 16 bit NodeID
- New message types and options
- Some level of interoperability with 3rd party ZigBee PRO compliant nodes

For more information please also refer to the AT-Command dictionary for the R3xx firmware which is available from www.telegesis.com. This document was written when the current version was R305. It does not contain any references to the ETRX357 family of modules because there is no version of the R2xx firmware for the ETRX357, so the ETRX357 does not really have a migration path. Nevertheless, statements about the R3xx firmware generally apply to both the ETRX2 and the ETRX357.

1.1 A Note on Interoperability and ZigBee® Compliance

The R3xx versions of the Telegesis firmware are not over the air compatible to previous releases of the Telegesis AT Commandset (R2xx and R1xx).

The Telegesis R300 firmware has been tested and certified for MSP (manufacturer specific profile) compliance by a test house appointed by the ZigBee Alliance.

This certification includes tests guaranteeing that:

- Modules running the Telegesis AT-Commandset will not interfere with existing ZigBee Networks in a malicious way
- Modules running the Telegesis AT-Commandset can join a 3rd party ZigBee PRO network and use its routing capabilities
- Modules running the Telegesis AT-Command set can allow third party nodes to join into a network consisting of Telegesis nodes and use its routing capabilities

In addition to implementing a manufacturer specific application profile, the AT-Command set allows for a certain level of transparency allowing to communication with third party nodes running any public or manufacturer specific application profile.

If you want to use the term ZigBee or the ZigBee Logo in your product documentation the current regulations state that you have to be at least an adopting member of the ZigBee Alliance.

A “ZigBee Certified Product” must, among other things, use a public profile and be submitted for compliance testing; it can then carry the “ZigBee Certified Product” logo. R3xx firmware uses our manufacturer specific profile and so cannot meet these requirements. If you intend to build a

product compliant to a public application profile (e.g. Home Automation, Smart Energy) feel free to contact us to discuss your options. For up-to-date details of the requirements relating to the use of logos and the term “ZigBee” you should refer to the ZigBee Alliance’s website.

1.2 Firmware Customization

Please note that our customers have the option of using Ember’s suite of development tools to develop custom firmware to run on the ETRX2 range of modules and overwrite the Telegesis AT Commandset.

For Profile compliant development (e.g. Home Automation Light Switch) Ember provides an application builder tool, which greatly simplifies the development of customized firmware.

If you are interested in this please contact Ember at www.ember.com.

In addition to this Telegesis is happy to help with firmware customizations for high volume projects and finally it is also possible to use Ember’s low level EZSP over UART interface on the ETRX2 range of modules. Information on the EZSP over UART interface again can be found at www.ember.com.

1.3 Updated command-line interaction

To ease the interaction with a host microcontroller the command-line has been altered as follows:

- The <CR> which terminates a command is echoed
- Every reply is encapsulated by <CR><LF><response><CR><LF> including the “OK” and “ERROR:XX” prompts
- In addition to this when setting bit 7 of S12 every reply or prompt is framed by the STX and ETX characters.

These changes have been introduced due to popular demand and should improve the interpretation of the data provided by the ETRX2 range of modules on host microcontrollers with limited processing power.

2 Network setup

To setup and maintain a network ZigBee PRO has brought a lot of improvements and new features, which are reflected in the new Telegesis R3xx AT Commandset. Please note that the new security features are covered in more detail towards the end of this document.

2.1 Starting a PAN

As with the previous version the command **AT+EN** forms a new network and makes the local node the coordinator and trust centre. In order to successfully start a network it must be made sure that the local node is not part of a network already.

As before **AT+EN** first makes the module scan all channels not masked out in S00 in order to start a new PAN on the quietest channel. When starting a new PAN the following parameters are set randomly unless defined otherwise in the respective S-Registers.

- PAN ID
- Network key
- Extended PAN ID
- Trust Centre Link key

We already know the PAN ID (PID) and Network key from R2xx, but now we have also generated a 64-bit extended PAN ID (EPID). The purpose of the EPID is to detect PAN ID conflicts, so when two PANs with the same PID, but a different EPID will see each other a conflict is assumed and one of the PANs will automatically change its PID on all nodes.

Due to this it is highly recommended to always use a random EPID (default) since forcing a specific EPID may cause conflicts in which two networks with identical PID and EPID may come to existence.

The prompt indicating the PAN to which membership has been altered:

JPAN:<channel>,<PID>,<EPID>

The new trust centre link key allows nodes to communicate securely with the trust centre (coordinator). This will be explained further in the security section towards the end of this document.

Note: Due to ZigBee PRO features the coordinator/trust centre must remain part of the network and cannot leave the network or move out of range once it has started the network.

2.2 Joining a PAN

When a node wants to join an existing PAN there are various ways to do that. First of all a node may wish to check which PANs are around before trying to join one of them. This can be done using the command **AT+PANSCAN**.

For each PAN found the following prompt will be shown:

+PANSCAN:<channel>,<PID>,<EPID>,<ZigBee stack profile>,<join status>

where the ZigBee stack profile and the join status are explained in the tables below:

ZigBee Stack Profile number	Interpretation
00	Custom
01	ZigBee
02	ZigBee PRO

Join status	Interpretation
0	Joining allowed
1	Joining permitted

Notes

- The command AT+PANSCAN in the same way as AT+ESCAN may now also be executed if the node is already part of a PAN.
- Even if AT+PANSCAN indicates that joining is allowed there may still be reasons why a node is not able to join a specific network. See the chapter on security towards the end of this document for more information.
- Routers can only join ZigBee PRO networks (stack profile number 02), whereas end devices can join ZigBee as well as ZigBee PRO networks (stack profile numbers 01 and 02 respectively)

Now that we have hopefully found a network which we want to join we can do so using the command **AT+JPAN:<channel>,<PID or EPID>**. As you can see it is now possible to specify either the PAN ID or the new extended PAN ID when joining a PAN.

Alternatively there is the option of joining the next best PAN, which will allow us to join using the command **AT+JN**. This command is also the basis of the build in functionality 0015, which by default is executed every minute if the node is not part of the network. The functionality behind this has also been improved so that when a node has not been able to join the first best network for whatever reason it will continue scanning and will try to join the next best one.

2.3 Managing a PAN

The command to tell the local node to leave the network (**AT+DASSL**) has remained unchanged in the same way as the equivalent for a remote node (**AT+DASSR**) has remained unchanged. The only exception is that a remote node can now be addressed in three possible ways, which are described in more detail in the messaging section.

Note: **AT+DASSR** is using standard ZigBee messaging (ZDO), so it can also be used to throw non-Telegesis nodes out of the network.

To find all nodes on the network there are now two ways. First of all it is possible to use the known command **AT+SN** to get all recipients to report in within a five-second window. This works well when executed on a sink or the coordinator because all nodes in the network know a route to the sink and/or the coordinator, but only nodes running the Telegesis firmware will ever respond to this command as this functionality is part of Telegesis's manufacturer specific profile.

To find all nodes in the network including third party nodes the ZigBee standard allows to ask any device on the network for its neighbours in a fully ZigBee compliant way (ZDO). This has been implemented using the command **AT+NTABLE:<index>,<address>**.

When sending this command to either a remote node or the local node the reply could look like this:

```

NTable: 03
No. | Type |      EUI          | ID | LQI
0.  | FFD | 000D6F000015896B | BC04 | FF
1.  | FFD | 000D6F00000B3E77 | 739D | FF
2.  | FFD | 000D6F0000AAD11  | 75E3 | FF
    
```

This shows that the node has three neighbours (NTable: 03) in its neighbour table with the listed device type, EUI64 and 16 bit NodeID. In case the node has more than three neighbours in its neighbour table it may be required to call **AT+NTABLE** repeatedly with incremented index counts as only three entries can be displayed at the same time.

So a ZigBee compliant way of finding all nodes on the network as well as the structure of the network is to extract the neighbour table from a known node (e.g. the coordinator which always has NodeID 0000, or the local node). From there we get to know additional nodes whose neighbour table we could then interrogate in the same way and thus build up the topology of the network.

This is more effort than using the AT+SN command which Telegesis provides, but it works with all ZigBee PRO compliant devices in networks of any size, rather than with Telegesis nodes only.

Finally to get information on the network status of the local node the command **AT+N** can be used as before with the only exception that now the EPID is displayed in addition to the PID in the response:

```
+N=<devicetype>=<channel>,<power>,<PID>,<EPID>
```

3 Messaging

As previously mentioned all commands which target a remote node in the past could only be addressed using the remote node's 64 bit EUI.

With the R3xx series of the Telegesis AT Commandset it is now possible to address remote nodes in any of the following three ways

1. By its 64-bit EUI
2. By its 16-bit NodeID
3. By its address table entry

When joining the network every node is assigned a random 16-bit NodeID, which is used to address that device. This shortens the ZigBee headers used for addressing purposes and makes ZigBee messaging more efficient.

In R2xx this feature has been hidden under the bonnet, but can now be accessed, however the user should note that a 16-bit NodeID can change due to a device leaving and re-entering the network or due to the detection of address conflicts, whereas its EUI64 will never change.

Finally it is possible to add nodes which are repeatedly the target of a transmission to an address table, which can be referenced using an 8-bit index number.

To display the 6 entries of the address table the command **AT+ATABLE** can be used. The last entry (05) contains the address of the sink (if known) and can be overwritten by the command **AT+ASET:xx,<NodeID><EUI64>**, which is used to set an address table entry as described in the AT-Command dictionary, should you wish to override the normal sink selection.

Note: If the NodeID of a device is unknown when creating an address table entry the NodeID must be substituted with FFFF

The commands **AT+BCAST**, **AT+BCASTB** have not changed at all, apart from the fact that the maximum allowable payload for all message types is now 82 bytes (in case the EUI gets attached only 74 bytes) and the maximum number of hops has been extended to 30.

AT+SCAST and **AT+SCASTB** have not changed either apart from the fact that they now share the same UCAST prompt with **AT+UCAST** and **AT+UCASTB** since under the bonnet **AT+SCAST** is identical to **AT+UCAST:05** (a unicast to address table entry 05).

Note: Every node can send a message to itself, for example **AT+IDENT:<own EUI64 or NodeID>** to beep the local beeper, given it is part of a network. To aid this, addressing address table entry FF automatically sends the message to the local node, e.g. **AT+DASSR:FF** is identical to **AT+DASSL**.

In addition to these known message types, a new message type has been introduced: the Multicast. Multicasts are distributed similar to a broadcast, but only nodes which have subscribed to the correct 16 bit multicast group will actually interpret the multicast. A typical application scenario would be to switch all lights in the living room, but not the kitchen.

To check which multicast groups a node has subscribed to the command **AT+MTABLE** can be used. To alter an entry of the multicast table the command **AT+MSET** can be used. In analogy to a broadcast, a multicast allows you to specify the number of hops a message is allowed to travel in addition to the multicast ID.

Note: The multicast table is volatile, so we have added S-Registers allowing setting of the initial value for the first two entries of the multicast table in S3E and S3F.

4 S-Registers

The list of S-Registers has expanded rapidly. Going into detail on each and every bit is beyond the scope of this document, however it is worth flagging up a few important add-ons:

- The local node's EUI and NodeID are now shown in S-Registers as well as the parent's EUI and NodeID if the device is an end device.
- Pull-ups as well as Pull-downs can be enabled for all I/Os
- 4 x A/D registers of which two are enabled by default
- 4 x registers defining the functionality of the 4 interrupts
- 2 new registers which can hold a functionality which is executed directly after bootup
- Source and destination endpoints for xCASTs (see chapter 10)
- Profile IDs for xCASTs (see chapter 10)
- Cluster IDs for xCASTs (see chapter 10)

Note: The S-Register numbers have changed compared to the R2xx series of the Telegesis AT commandset. See Appendix B for a comparison list.

For accessing remote S-Registers the command has changed to

ATREMS:<address>,xx[x]? (read access)

and

ATREMS:<address>,xx[x]=<data> (write access)

to make the command more consistent with other addressed commands. Like with all messages the address for this command can be a EUI64, a NodeID as well as an address table entry.

Finally an additional parameter has been added to the **ATSALL** command allowing to address only devices subscribed to a particular multicast group, all non-sleepy devices (routers and the coordinator), or all devices in the network. (**ATSALL:<group ID>,xx[x]=<data>**)

5 Sink Handling

One of the great new improvements is that it is now possible to have more than a single sink in the network. Every node in the network will automatically send its data to the sink nearest to it.

When a sink advertises itself using functionality 02xx (where xx indicates the number of hops it advertises for) it does two things:

1. Create a many-to-one routing entry on all remote nodes (ZigBee PRO feature)
2. Advertise itself as a sink using a broadcast (for the Telegesis MSP to identify the sink)

Many-to-one routing means that every node in the network gets to know a route to the sink or concentrator automatically, so there is no need for route discoveries in case a node wants to send data to a sink. This mechanism is a ZigBee PRO feature.

As a many to one routing table entry obviously takes up one entry in the routing table (although it will eventually age out) it is recommended not to use an excessive amount of sinks in a network as this will fill up the routing tables and leave less space for regular routing entries.

As before **AT+SSINK** can be used to actively search for a sink in the same way as the node will by default search actively for a sink if a transmission to a sink is attempted and no sink is known.

By default a sink will be deleted once three consecutive transmissions to the sink fail.

Important Note: The coordinator will also create many to one routing entries on all devices in the network as this is required for it to function as a trust centre. Because of this it is important that functionality 02xx is not disabled by stopping the corresponding timer/counter on a coordinator even if the coordinator is not a sink.

6 Source Routing

Every message sent to a sink (and the coordinator) will be preceded by a route record message (ZigBee PRO feature). A route record message travels from node to node towards the sink or coordinator following the many-to-one route and at each hop the local node's NodeID is added to the payload of the route record message. This causes each message sent to sink or coordinator to be preceded by a **SR:xx,<EUI>,<NodeID>,...** prompt. In addition to displaying the prompt, the path is stored in a temporary routing cache, allowing acknowledgements sent back to utilize the routing information. To do this the recorded path is attached to the outgoing message's payload and no route discovery is needed -> the message gets source routed.

When using source routing it is important to note that for every hop which is attached to the message's payload the application payload reduced by two bytes.

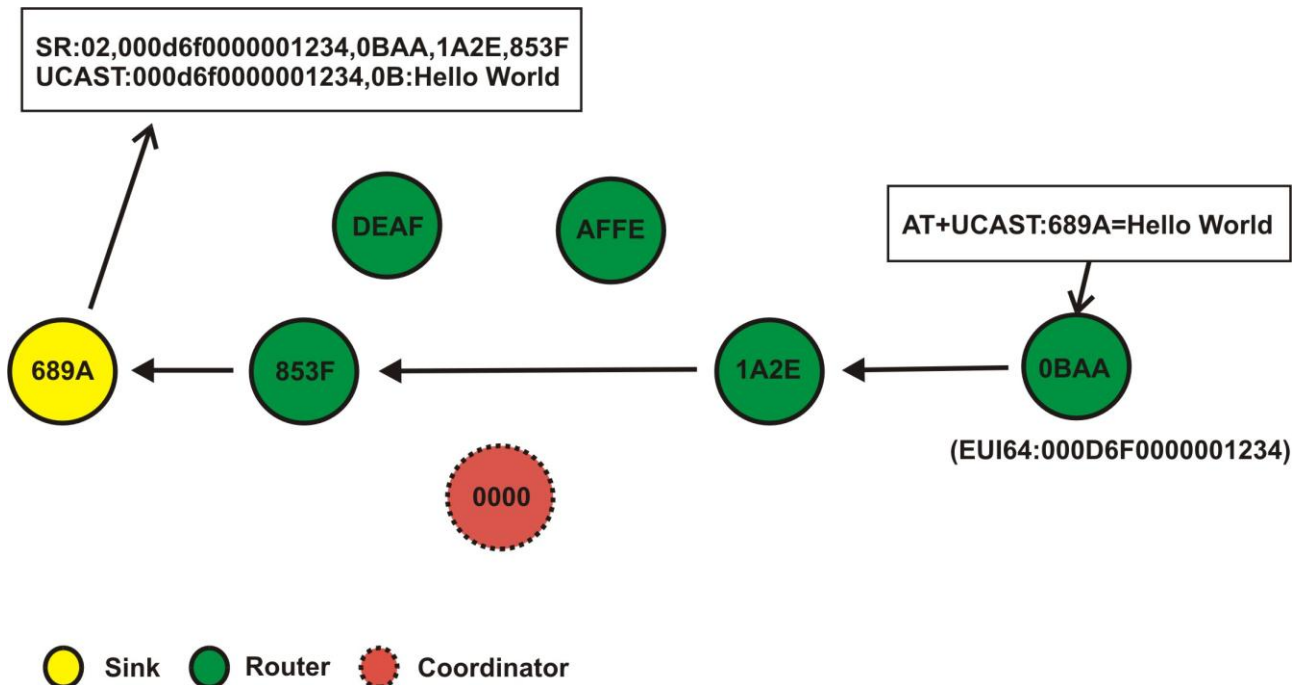


Figure 1: A Source record message is preceding the unicast to the sink

The new command **AT+FNDSR**, which only works on sinks and coordinators, does nothing but send a dummy UCAST to the remote node, triggering a route record message to be assembled to precede the acknowledgement and thus result in a **SR:** prompt. **AT+FNDSR** can also be used to find the number of hops along a route and thereby calculate the maximum message payload.

The scenario in which the central node wishes to communicate with many (hundreds or even thousands) of outstations at any given time in the past has always been a problem in ZigBee as the routing tables are too small to hold the next hops for all the potential addressees. This used to cause the need for excessive route discoveries which eventually used up all available bandwidth.

This can now be overcome by connecting a powerful processor to the sink which has enough memory to save all the recorded routing information and tell the sink how to route a specific message prior to sending it.

This can be done using the **AT+SR** command. The **AT+SR** command forces all messages to the specified destination to be routed through exactly the defined path, unless overwritten.

For example executing on node with the Node ID 689A:

```
AT+SR:0BAA,1A2E,0000,853F
AT+UCAST:0BAA,hello world
```

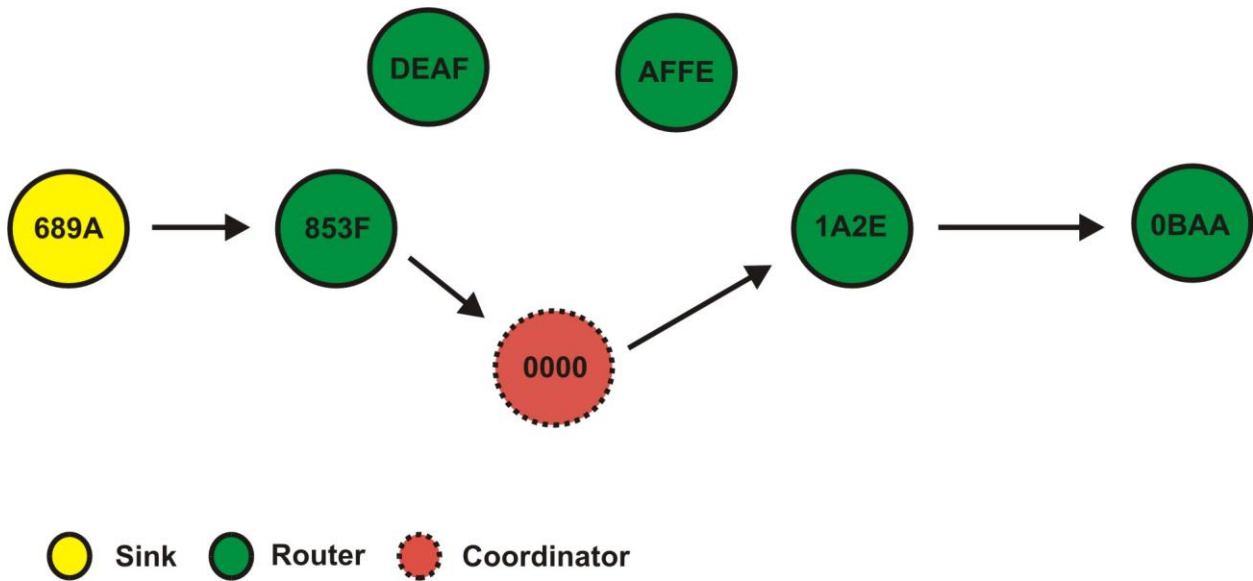


Figure 2: Specifying a source route

OR

```
AT+SR:0BAA,1A2E,AFFE,DEAF,853F
AT+UCAST:0BAA,hello world
```

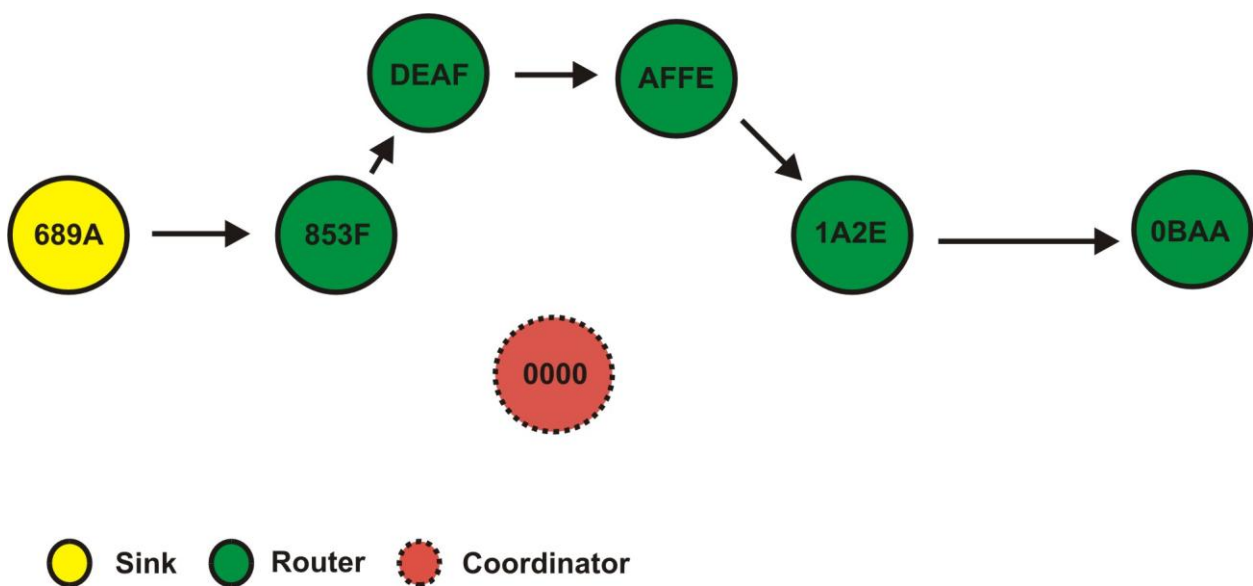


Figure 3: Specifying a different source route

Notes

- If the path which has been found for a source route breaks, the remote node will appear unresponsive. To fix the source route you can issue **AT+FNDSR** again to find a broken source route and restore its information.
- AT+SR can be used to force messages to take a specific path through the network, even if this path is not the one with the lowest cost

7 Build in Functionality

Most of the build in functionality has remained the same apart for the fact that whether a timer triggering a functionality is restarted or not now depends on the most significant bit. See Appendix C for a comparison of R212 and R3xx.

For example

- 0001 will be executed once and the timer will be disabled
- 8001 will be executed and the timer will be restarted

In addition to this some basic RSSI based tracking functionality has been added with functionalities 0111 and 0112. This is based on an 802.15.4 message being sent by the node which is to be tracked and on reception all routers hearing this message will notify the nearest sink together with their RSSI readings. (This is a Telegesis add-on, not a ZigBee-compliant function.)

Finally functionality 0012 to 0014 has been extended to give the user a broader range of choices what to do in case a device has orphaned.

8 End Device Handling

There is also some news around end devices and end device handling:

- It is now possible to have 16 end devices per parent (32 with an ETRX357), which can be any mix of mobile end devices as well as sleepy end devices
- Sleepy end devices also time out of the parent's child table in case they have not polled for 5 minutes (mobile ones do so after 5 seconds)
- Finding a new parent is quicker and also the best parent (based on RSSI) is automatically chosen
- End devices will indicate "LeftPAN" in case they have not polled for long enough (requires power mode < 3)
- There is a new command for an end device which has timed out and wants to re-join the network: **AT+REJOIN** with the option to join encrypted or unencrypted. It should be tried to re-join unencrypted first and only if this fails, which could be the case because the network key has been changed, it is recommended to try re-joining unsecured. In the later case the node re-joins without knowing the network key and is passed the network key from the trust centre encrypted with the trust centre link key.
- When polling and the parent has timed out a secured re-join is attempted automatically. As mentioned before an unsecured re-join is very effective.
- When searching for a parent to join the network and multiple parents reply the parent with the highest RSSI is used.

9 Miscellaneous New Features

- The response to the ATI command now draws the unit name directly from manufacturing information. On some older modules manufactured before the beginning of 2007 the ATI command may now show unreadable characters. This does not affect the functionality of the module.
- A method for over the air upgrading from R2xx to R3xx and vice versa is given in the R300 command dictionary
- The additional ADCs and IRQs can be enabled in S11
- Unbalanced link detection and optimizations for dense networks have been added with introduction of the new Ember ZNet3.1

9.1 Changes between R301 and R302

In the R302 release we were able to bring the R212 AT+OPCHAN functionality back, which had to be omitted because of memory constraints. (A new revision of the XAP2b compiler allowed us to bring back this popular feature.) To avoid ambiguity over the use of the term “channel”, the new command is AT+DMODE (data mode). There is no equivalent of the old AT+OPLCHAN because the new command permits the use of the sequence “+++”, provided it is not preceded by a pause longer than about 250ms.

10 Security

As previously mentioned the security features have been improved with the addition of the new ZigBee PRO feature set to the ZigBee standard.

The Telegesis AT command set implements what is described as “standard security” in the ZigBee specification (security level 5).

As with the previous version of the firmware all messages are AES encrypted at network level with a 128-bit network key. When a device becomes the coordinator and forms a PAN it will generate a random network key unless a valid (non-zero) key is provided in S-Register 08.

Nodes joining the network will always have to receive the valid network key directly from the coordinator (trust centre) over the air. It is not possible for nodes to already know the network key at the time of joining the PAN. Unlike with previous versions of the firmware, specifying a network key on S-Register 08 on any other node than the one becoming the coordinator has no effect.

This means the coordinator is now in charge that every node in the network (which is allowed to join) will receive the network key over the air, therefore the coordinator is also operating as the trust centre.

Notes:

- The Telegesis AT Commandset does not allow a device other than the coordinator to become the network’s trust centre.
- Only the trust centre can allow new nodes onto the network. It is not possible any more that any router makes this decision without interrogating the trust centre.
- The coordinator is no longer allowed to leave the PAN and rejoin as an ordinary router, since the PAN will not have a trust centre.
- (In R3xx it is possible to use a distributed Trust Centre mode, but this is not truly ZigBee compliant)

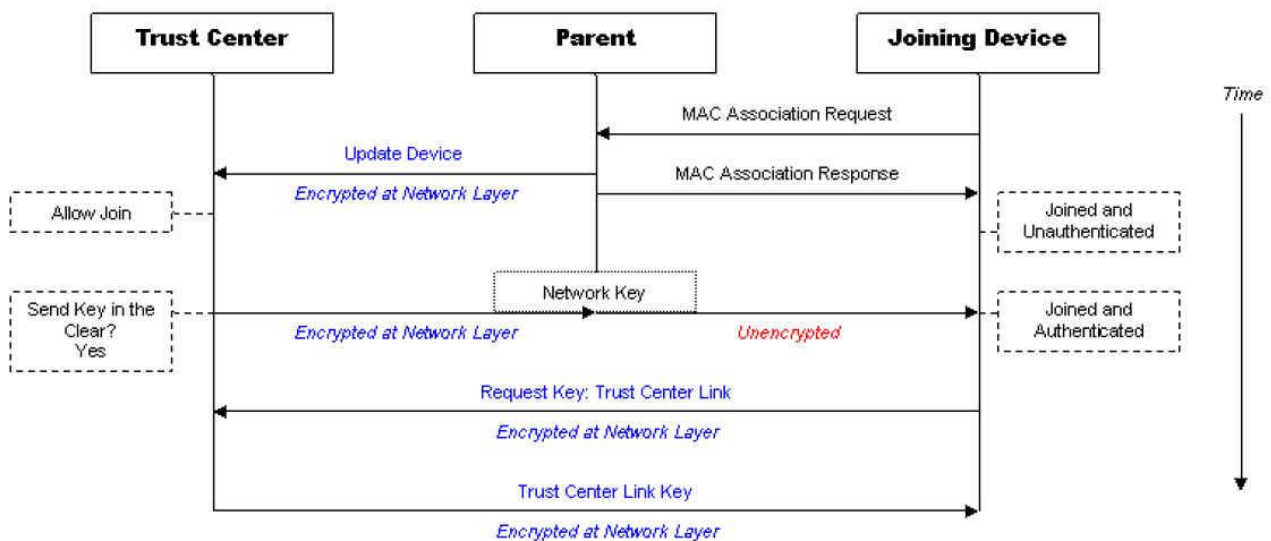


Figure 4: A Device joining the network via a parent gets the network key sent in the clear because it does not know the trust centre link key. After joining, the joined device will automatically request the trust centre link key and receives it encrypted by the network key.

Sending the network key over the air to newly joined nodes at the moment of joining could be captured and is therefore a security risk as shown in figure 4. To avoid this the trust centre can send the network key to the remote node encrypted using the trust centre link key. For this to work there are two options:

1. The Trust Centre and the new device both know the same global trust centre link key (in S-Register 09). For a joining device to use the trust centre link key stored in S09 bit 8 of S0A needs to be set.
2. The new device had joined the network before (secure rejoin) and the trust centre link key was passed to it on request after joining.

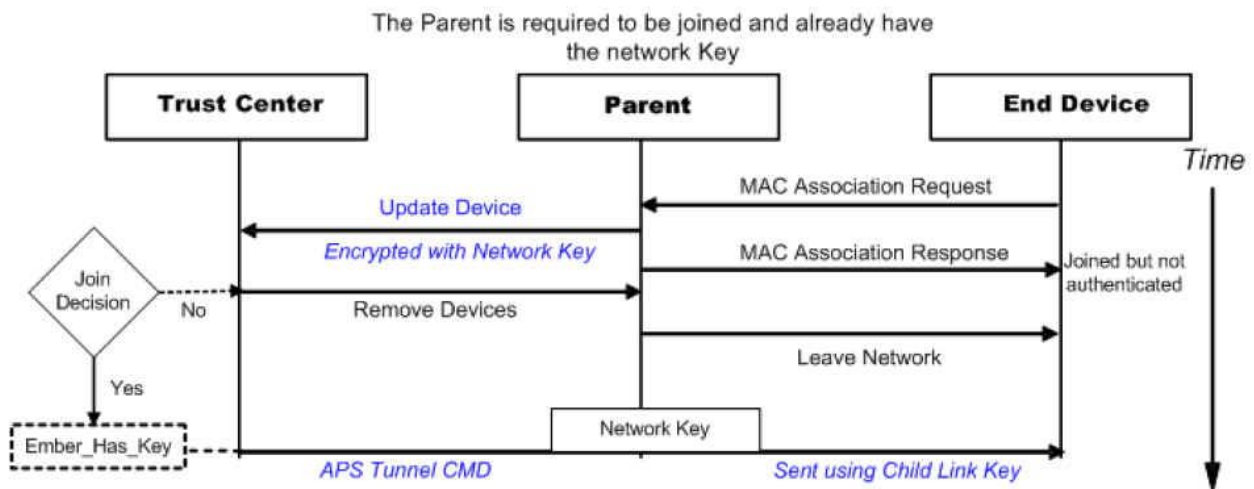


Figure 5: Joining node knows the trust centre link key

The trust centre link key can be one global key known by all devices wishing to communicate with the trust centre, or the trust centre can issue individual hashed link keys to each and every joining device (enabled by setting bit 7 of S-Register 0A on the coordinator before starting a network).

10.1 Join Decisions

The trust centre is now the key for setting the joining and security policies of the network.

There are a few simple decisions, which can be made to do this.

By default the trust centre will allow any node to join and re-join the network and happily send the network key in the clear to everybody. This is the default state in which the Telegesis ZigBee modules are shipped as this allows for a very easy network setup.

The Trust centre can now decide to:

1. Don't allow any new nodes to join
2. Send the network key encrypted with the link key to every node trying to join
3. Send the network key unencrypted to every node trying to join (default)

and in the same way for re-joining nodes:

1. Don't allow any nodes to re-join the network
2. Send the network key encrypted with the link key to every node trying to rejoin
3. Send the network key unencrypted to every node trying to rejoin (default)

When selecting the security settings for your network please note that a malicious node could also pretend to be re-joining although it is actually joining.

A common example would be to send the network key unencrypted to all joining nodes during the initial setup of the network, then once the network is completed don't allow any new nodes to join and at the same time allow re-joining using trust centre link key encrypted network keys (secured re-join). This setup is very secure unless during the initial setup phase the network key is compromised.

Another way would be to only allow secured joining as well as re-joining with a global trust centre link key programmed into S-Register 09 at production time.

Setting bit 0 of S-Register 0A to 1 simply does not allow nodes to join via this particular node as a parent. This is also the bit which is overwritten by functionality 0017, so unlike with the R2xx series of firmware this feature cannot be used to control joining on a global scale.

10.2 Changing Network keys

Another improvement to ZigBee security is a protection against replay attacks in which any packet is simply captured and replayed without being able to decrypt its contents.

To prevent this type of attacks every node which passes on a message will decrypt it at network level, check its integrity and re-encrypt the message with the link's frame counter which is incremented after each transmission. All devices maintain a list of their neighbour's and children's frame counters. Every time a device sends a packet, it increments the frame counter. A receiving device verifies that the frame counter of the sending device has increased from the last value that it saw. If it has not increased, the packet is silently discarded. If the receiving device is not the intended network destination, the packet is decrypted and modified to include the routing device's frame counter. The packet is then re-encrypted and sent along to the next hop.

The frame counter is 32 bits and **may not wrap to zero**. Prior to the frame counter hitting its maximum value, the Network Key must be updated. This is done using functionality 0011 or AT+KEYUPD. However, this is unlikely to be a major issue as it takes a very long time to overflow a 32-bit counter.

The new network key sent out to all nodes is encrypted using the old network key. In case a device misses the network key update because it was asleep or temporarily unavailable it can use the trust centre link key to obtain the new network key (secured rejoin).

In case the device does not know the trust centre link key and has missed a network key update it will have to rejoin in an unsecured way, which is comparable to an unsecured join.

11 Interaction with other ZigBee PRO devices

To explain the interaction with other third party ZigBee devices we should start with a brief explanation on how ZigBee is structured.

The ZigBee Specification is based on the IEEE 802.15.4 standard. On the lower level ZigBee defines stack profiles of which there are two at the moment: ZigBee and ZigBee PRO. ZigBee devices and ZigBee PRO devices cannot interact, with the only exception of end devices, which can connect to networks of both types.

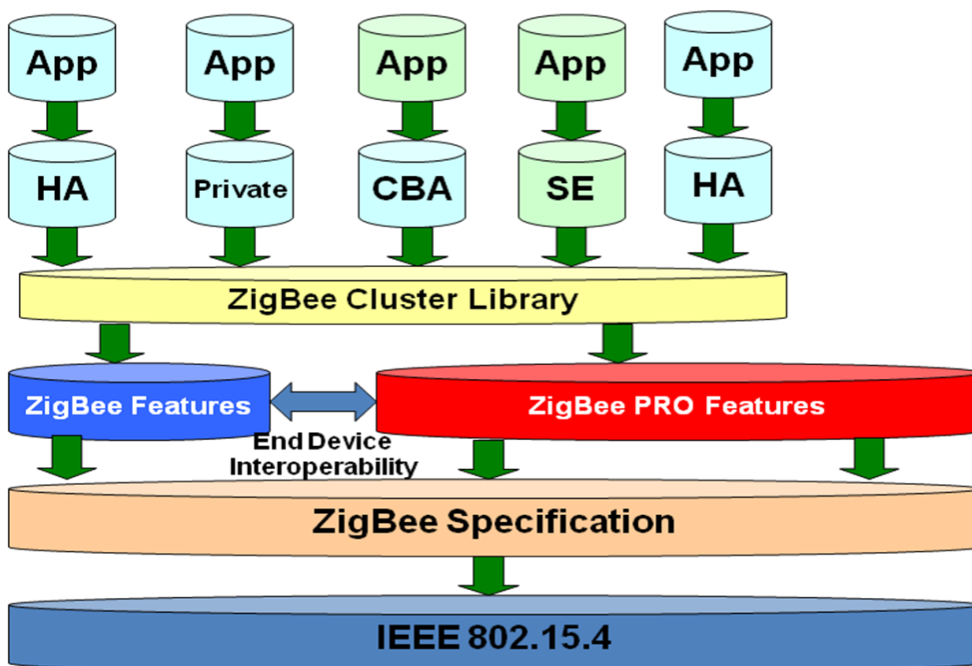


Figure 6: The ZigBee Standard

On top of the stack profiles ZigBee defines application profiles like HA (Home Automation), SE (Smart Energy, previously known as AMI). There are additional application profiles being worked on by various ZigBee Alliance profile task groups (PTG) like CBA (commercial Building Automation) and many more.

To prevent that every profile task group ends up re-inventing the wheel the ZigBee cluster library has been formed. The cluster library is a collection of functionalities, which can be used by the individual profile task groups. For example the HA application profile can now utilize the same on/off cluster as the CBA application profile, thus preventing the individual PTG duplicating message definitions which have already been defined differently by other PTGs.

To make things even more confusing from an outside point of view every ZigBee device can have multiple endpoints. For example there could be a light switch with an integrated temperature sensor representing two individual devices connected to a single ZigBee radio. This device would have the following endpoints:

1. Endpoint 0: The ZDO (ZigBee Device Objects). This endpoint has to be offered by all ZigBee compliant devices. It is used to commission and identify devices on a network. For example the commands **AT+NTABLE** or **AT+DASSR** use this endpoint.
2. Endpoint x: The light switch (x doesn't have to be 1, but can be)
3. Endpoint y: The temperature sensor (y doesn't have to be 2, but could be)

A message being sent to this particular device could now be addressed as follows:

1. Address of the ZigBee radio (64-bit EUI64 or 16-bit NodeID)
2. Endpoint no. y
3. Cluster xxxx -> give me your temperature reading

or

1. Address of the ZigBee radio (64-bit EUI64 or 16-bit NodeID)
2. Endpoint 0 (ZDO)
3. Cluster xxxx -> "What is attached to your endpoint 1" or "leave the network" etc.

To offer the known functionality of the Telegesis AT commandset it was required to define a manufacturer specific (or private) application profile (MSP) to allow for remote S-Register accesses, accessing the I/O capabilities of the module etc.

The Telegesis AT command firmware has three endpoints:

1. Endpoint 0: The ZDO (ZigBee Device Objects). This endpoint has to be offered by all ZigBee compliant devices. It is used to commission and identify devices on a network. For example the commands AT+NTABLE or AT+DASSR use this endpoint.
2. Endpoint 1: The Telegesis Commandset (MSP). This endpoint communicates with the manufacturer specific functionality of the ETRX2 module.
3. Endpoint 2: User Endpoint. Any data sent to this endpoint is displayed to the user transparently.

When sending messages using UCAST[B], BCAST[B] and MCAST[B] the user can now define the source and destination endpoints as well as the cluster ID, which allows addressing of any third party ZigBee compliant device. By default all messages will be sent from endpoint 1 to endpoint 1 with cluster ID 0002. This cluster of the Telegesis MSP is displaying all incoming messages in the known format (UCAST:... BCAST:... MCAST:...).

11.1 Features of R302 and later versions

In the R302 release the description of endpoint 2 is more configurable and also there is AT command support for additional ZDO functionalities to aid interaction with third party ZigBee PRO compliant devices. In R305, any endpoint (apart from 0 and 1) can receive messages in the same way as endpoint 2, but only endpoint 2 is listed in the response to a ZDO request for active endpoints.

12 Glossary

BCAST	Broadcast
COO	Coordinator
FFD	Full Function Device
MCAST	Multicast
MED	Mobile End Device
MSP	Manufacturer specific profile
PAN	Persona Area Network
PTG	Profile Task Group (ZigBee Alliance)
RSSI	Received Signal Strength Indication
SCAST	Unicast directed to the nearest sink/concentrator
SED	Sleepy End Device
TC	Trust Centre
UCAST	Unicast
ZDO	ZigBee Device Object

13 Appendix A: AT Command Comparison

R2xx	R3xx	Description
ATI	ATI	Display Product Identification Information
ATZ	ATZ	Software Reset
AT&F	AT&F	Restore Factory Defaults
AT+BLOAD	AT+BLOAD	Enter The Bootloader Menu
AT+CLONE	AT+CLONE	Clone Local Node To Remote Node
AT+RECOVER	AT+RECOVER	Recover From A Failed Clone Attempt
AT+PASSTHROUGH		Enter pass-through bootloading mode (ETRX1 only)
ATS	ATS	S-Register Access
ATSALL	ATSALL	Remote S-Register Access
AT+TOKDUMP	AT+TOKDUMP	Display All S-Registers
ATSREM	ATREMS	Remote S-Register Access
AT+ESCAN	AT+ESCAN	Scan The Energy Of All Channels
AT+EN	AT+EN	Establish Personal Area Network
AT+JN	AT+JN	Join Network
AT+PANSCAN	AT+PANSCAN	Scan For Active PANs
AT+JPAN	AT+JPAN	Join Specific PAN
AT+DASSL	AT+DASSL	Disassociate Local Device From PAN
AT+DASSR	AT+DASSR	Disassociate Remote Node from PAN
AT+NTABLE	AT+NTABLE	Display Neighbour Table
AT+N	AT+N	Display Network Information
AT+CTABLE		Display list of local children
AT+PARENT		Display Parent's ID
AT+POLL	AT+POLL	Poll for Data from Parent
AT+SN	AT+SN	Scan Network
	AT+ATABLE	Display Address Table
	AT+ASET	Set Address Table Entry
	AT+MTABLE	Display Multicast Table
AT+REMSN		Scan for remote device's direct neighbours
	AT+MSET	Set Multicast Table Entry
AT+LINKCHECK		Check link parameters with a neighbour
AT+PING	AT+ANNCE	Indicate Presence In The Network
AT+BCAST	AT+BCAST	Transmit A Broadcast
AT+BCASTB	AT+BCASTB	Transmit A Broadcast Of Binary Data
AT+UCAST	AT+UCAST	Transmit A Unicast
AT+UCASTB	AT+UCASTB	Transmit A Unicast Of Binary Data
AT+SCAST	AT+SCAST	Transmit Data To The Sink
AT+SCASTB	AT+SCASTB	Transmit Binary Data To A Sink

R2xx	R3xx	Description
AT+SSINK	AT+SSINK	Search for a Sink
AT+RDATAB	AT+RDATAB	Send Binary Raw Data
	AT+MCAST	Transmit A Multicast
	AT+MCASTB	Transmit A Multicast Of Binary Data
	AT+FNDSR	Find the Source Route to a remote device
	AT+SR	Set Source Route to Remote Device
AT+SINK		Display the local Node's sink
AT+OPCHAN		Opens a channel to a remote node
+++		Close channel
AT+OPLCHAN	AT+DMODE	Opens a limited channel to a remote node
AT+ACKCHAN		Accept channel
AT+IDENT	AT+IDENT	Play A Tune On Remote Devboard

14 Appendix B: S-Register Comparison

R2xx		R3xx	
S00	Channel Mask	S00	Channel Mask
S01	Preferred PAN ID	S02	Preferred PAN ID
S02	Transmit Power Level	S01	Transmit Power Level
S03	Encryption key	S08	Network Key
		S03	Preferred Extended PAN ID
S04	User Definable name	S0B	User Readable Name
		S04	Local EUI
		S07	Parent's NodeID
S05	OEM Word		
S06	Main Function	S0A	Main Function
S07	Extended Function1	S0E	Prompt Enable 1
S08	Extended Function2	S0E	Prompt Enable 1
		S0F	Prompt Enable 2
		S09	Link Key
		S10	Extended Function
S09	Password	S0C	Password
S0A	Revision Number	S0D	Device Information
S0B	UART Setup	S12	UART Setup
S0C	ETRX2: Pull-up enable ETRX1: Reserved	S13	Pull-up enable
S0D	Data Direction of I/O Port (DDR) (volatile)	S16	Data Direction of I/O Port (volatile)
S0E	Initial value of S0D	S17	Initial Value of S16
S0F	Output Buffer of I/O Port (PORT) (volatile)	S18	Output Buffer of I/O Port (volatile)
S10	Initial value of S0F	S19	Initial Value of S18
S11	Input Buffer of I/O Port (PIN) (volatile)	S1A	Input Buffer of I/O Port (volatile)
S12	A/D1	S1F	A/D1
S13	A/D2	S20	A/D2

R2xx		R3xx	
S14	ETRX2: A/D3 (Reserved) ETRX1: Reserved	S21	A/D3
S15	Immediate functionality at IRQ0	S23	Immediate functionality at IRQ0
S16	Immediate functionality at IRQ1	S24	Immediate functionality at IRQ1
		S25	Immediate functionality at IRQ2
		S26	Immediate functionality at IRQ3
		S27	Functionality 1 at Boot-up
		S28	Functionality 2 at Boot-up
S17	Timer/Counter 0	S29	Timer/Counter 0
S18	Functionality for Timer/Counter 0	S2A	Functionality for Timer/Counter 0
S19	Timer/Counter 1	S2B	Timer/Counter 1
S1A	Functionality for Timer/Counter 1	S2C	Functionality for Timer/Counter 1
S1B	Timer/Counter 2	S2D	Timer/Counter 2
S1C	Functionality for Timer/Counter 2	S2E	Functionality for Timer/Counter 2
S1D	Timer/Counter 3	S2F	Timer/Counter 3
S1E	Functionality for Timer/Counter 3	S30	Functionality for Timer/Counter 3
S1F	Timer/Counter 4	S31	Timer/Counter 4
S20	Functionality for Timer/Counter 4	S32	Functionality for Timer/Counter 4
S21	Timer/Counter 5	S33	Timer/Counter 5
S22	Functionality for Timer/Counter 5	S34	Functionality for Timer/Counter 5
S23	Timer/Counter 6	S35	Timer/Counter 6
S24	Functionality for Timer/Counter 6 (volatile)	S36	Functionality for Timer/Counter 6
S25	Initial Functionality for Timer/Counter 6		
S26	Timer/Counter 7	S37	Timer/Counter 7
S27	Functionality for Timer/Counter 7 (volatile)	S38	Functionality for Timer/Counter 7
S28	Initial Functionality for Timer/Counter 7		
S29	Power mode (volatile)	S39	Power mode (volatile)
S2A	Initial Power Mode	S3A	Initial Power Mode
S2B	Start-up Functionality Plaintext A	S3B	Start-up Functionality Plaintext A

R2xx		R3xx	
S2C	Start-up Functionality Plaintext B	S3C	Start-up Functionality Plaintext B
S2D	Parent's EUI	S06	Parent's EUI
S2E	Device Specific	S11	Device Specific
S2F	Special Function Pin 1 (volatile)	S1B	Special Function pin 1 (volatile)
S30	Initial value of S2F	S1C	Initial Value of S1B
S31	Special Function Pin 2 (volatile) (ETRX2 only)	S1D	Special Function Pin 2 (volatile)
S32	Initial value of S31 (ETRX2 only)	S1E	Initial Value of S1D
S33	Supply Voltage (ETRX2 only)	S3D	Supply Voltage
S34	16-bit Network ID (volatile)	S05	Local NodeID
S35	Start-up Functionality 16-bit register (volatile)		
		S3E	Multicast Table Entry 00
		S3F	Multicast Table Entry 01
		S40	Source and Destination Endpoints for xCASTs (volatile)
		S41	Initial Value of S40
		S42	Cluster ID for xCASTs (volatile)
		S43	Initial Value of S42
		S44	Profile ID for xCASTs (volatile)
		S45	Initial Value of S44
		S46	Start-up Functionality 32 bit number (volatile)
		S47	Power Descriptor
		S48	Endpoint 2 Profile ID
		S49	Endpoint 2 Device ID
		S4A	Endpoint 2 Device Version
		S4B	Endpoint 2 Input Cluster List
		S4C	Endpoint 2 Output Cluster List
		S4D	Mobile End Device Poll Timeout
		S4E	End Device Poll Timeout
		S4F	MAC Timeout

15 Appendix C: Built in Functionality Comparison

R2xx		R3xx	
0000	No operation of the corresponding interrupt/timer/counter	0000	No operation of the corresponding interrupt/timer/counter
0001 0005	Change to power mode 0.	0001	Change to power mode 0.
0002 0006	Change to power mode 1.	0002	Change to power mode 1.
0003 0007	Change to power mode 2.	0003	Change to power mode 2.
0004 0008	Change to power mode 3.	0004	Change to power mode 3.
0010 0011	If I am a Mobile/Sleepy end device Poll Parent for data.	0010	If I am a Mobile/Sleepy end device Poll Parent for data.
0012 0013	If I am a Sink advertise and stop timer (if applicable)	02xx	If I am a Sink advertise me for xx hops (max. no. of hops: 30)
		0011	Update the Network key with new random key.
0014	Check for neighbours in local neighbour table. If no neighbours are present for 5 consecutive times leave the PAN.	0012	Check for other devices on the network. If no other devices could be found for three consecutive tries, attempt a rejoin using the current network key each time this functionality is triggered.
		0013	Check for other devices on the network. If no other devices could be found for three consecutive tries, attempt a rejoin using the current network key. If this is unsuccessful try an unsecured rejoin each time this functionality is triggered from there on. Note: No functionality on COOs.
		0014	Check for other devices on the network. If no other devices could be found for three consecutive tries, attempt a rejoin using the current network key. If this is unsuccessful try a rejoin using the current link key the next time this functionality is triggered. If this is unsuccessful leave the current network the next time this action is triggered. Note: No functionality on COOs.
0015 0016	In case I am not joined to a network scan for and join the next best network	0015	In case I am not joined to a network scan for and join the next best
0017	Allow joining for 60 Seconds (in case it is disabled in S06)	0017	Allow joining for 60 Seconds (in case it is disabled in S0A)

R2xx		R3xx	
0018 0019	Copy local inputs to remote outputs: Read the local S11 and if changed since the previous time, write the reading to the remote S0F, whose address is given in S2B.	0018	Copy local inputs to remote outputs: Read the local S1A and if changed since the previous time, write the reading to the remote S18, whose address is given in S3B.
001A 001B	Copy remote inputs to local outputs: Read the remote unit's S11, whose address is given in S2C and write the reading to the local S0F.		
001C	Check for children which have not reported in and can be erased from the child table, and clean up table.		
001D	Close channel (if open)		
002x 003x	Toggle I/Ox	003x	Toggle I/Ox
004x	Flash I/Ox (pull low) for 250ms	004x	Flash I/Ox (pull low) for 250ms
		005x	Set I/Ox to 0
		006x	Set I/Ox to 1
0100 0101	Sends the reading of the I/O and the two analogue ports to the network's sink and if no sink is known the unit will search for a sink instead.		
0102 0103	Same as 0100/0101, but to charge an external RC timer I/O7 is pulled high whilst sending the data and left high impedance the rest of the time.		
0108 0109	The unit sends the contents of S2B to the networks sink.	0108	The unit sends the contents of S3B to the networks sink.
010A 010B	The unit sends the contents of S2C to the networks sink.	0109	The unit sends the contents of S3C to the networks sink.
0110 0111	Sends the reading of the I/O and the two analogue ports as well as an 8-bit transmission counter which increments with every transmission to the network's sink and if no sink is known the unit will search for a sink instead.	0110	Sends the reading of the I/O and the two analogue ports as well as an 8-bit transmission counter which increments with every transmission to the network's sink and if no sink is known the unit will search for a sink instead.
0112 0113	Same as 0110/0111, but to charge an external RC timer I/O7 is pulled high whilst sending the data and left high impedance the rest of the time.	0111	Same as 0110, but to charge an external RC timer I/O7 is pulled high whilst sending the data and left high impedance the rest of the time.
0120 0121	Sends the contents of S2B as a RAW transmission.	0120	Sends the contents of S3B as a RAW transmission.
0122 0123	Sends the contents of S2C as a RAW transmission.	0121	Sends the contents of S3C as a RAW transmission.
		0112	Send a Tracking Message to all nearby routers which will forward this message and the RSSI reading to their nearest sink.

R2xx		R3xx	
		0113	Same as 0112, but to charge an external RC timer I/O7 is pulled high whilst sending the data and left high impedance the rest of the time.
0200	Show status on I/O10. LED on (pin driven low) = no connection. Blinking fast = Auto-searching for PAN. Blinking slow = connected to PAN.		
0201	Show AT Command line's error status on I/O11. LED off no error. LED blinking = error. Reset by 'OK' prompt.		
2000	When triggered the number of times listed in the accompanying counter a message is sent to the sink containing a transmission counter and the reading of the analogue and digital inputs. Note: Can only be triggered by setting S15 or S16 to 400x.	2000	When triggered the number of times listed in the accompanying counter a message is sent to the sink containing a transmission counter and the reading of the analogue and digital inputs. Note: Can only be triggered by setting S23, S24, S25 or S26 to 230x.
2001	When enabling this action the command line is disabled and as soon as a number of bytes in excess of the number N specified in the accompanying timer/counter register is received on the serial port, a SCAST containing these characters is sent to the network's sink. If no sink is known a sink is searched instead. After 3 unsuccessful transmissions the sink is assume unavailable and a new sink is searched. Notes: This event is triggered by receiving a character on the serial port. $N \leq 64$.	2001	When enabling this action the command line is disabled and as soon as a number of bytes in excess of the number N specified in the accompanying timer/counter register is received on the serial port, a SCAST containing these characters is sent to the network's sink. Notes: This event is triggered by receiving a character on the serial port. $N \leq 64$.
0300 0301	Increment S35.		
0310 0311	Decrement S35.		
0320 0321	Clear S35.		
3000 3001	The contents of S2B is sent to the local command line followed by carriage return.	2100	The contents of S3B is sent to the local command line followed by carriage return.
3002 3003	The contents of S2C is sent to the local command line followed by carriage return.	2101	The contents of S3C is sent to the local command line followed by carriage return.
400x 401x	Start timer x.		
402x 403x	Toggle timer x.		
404x 405x	Stop timer x.		

R2xx		R3xx	
50xx 51xx	Start timers defined by the bitmask xx.	24xx	Start timers masked in xx.
		25xx	Toggle timers masked in xx.
		26xx	Stop timers masked in xx.
8xxx 9xxx	Change I/O port to the LSBs	3xxx	Change I/O port to the LSBs.
Axxx Bxxx	Change data direction of the I/O port to the LSBs	4xxx	Change data direction of the I/O port to the LSBs.