ETRX2 and ETRX357 ZigBee® MODULES

Application Note – R3xx Interoperability

Table of Contents

1 Introduction

As more and more ZigBee compliant end products conforming to public ZigBee application profiles like HA (Home Automation) or SE (Smart Energy) appear on the marketplace it becomes desirable to be able to interact with these 3rd party ZigBee compliant devices.

By default the Telegesis AT-Command firmware represents a non-public application profile (an MSP, or Manufacturer Specific Profile) defining the functionality triggered by AT commands. For example when sending a unicast to a Telegesis node it is the custom application profile which defines that on the receiving side "UCAST:…" is displayed.

This document describes how to use the Telegesis AT-Command firmware to interact with 3rd party ZigBee PRO compliant devices, including another manufacturer's MSP should the message format be known.

2 Background

To get a better understanding on how ZigBee messages are addressed and get interpreted by ZigBee PRO compliant nodes this section gives a bit of background information on how ZigBee Messages are composed and interpreted.

2.1 Stack Profiles

The Telegesis R3xx AT-Command firmware is based on a stack supporting the latest ZigBee PRO featureset (also known as the ZigBee PRO stack profile, or ZigBee 2007). In order to interact with ZigBee PRO compliant devices any remote device should also be ZigBee PRO compliant.

The only exception are end devices, in which case ZigBee PRO compliant end devices can interact with a ZigBee network and also ZigBee compliant end devices can interact with a ZigBee PRO network.

Only the ZigBee PRO featureset can offer fully self-healing mesh networking, whereas the ZigBee featureset (a.k.a. ZigBee 2006) is based on a tree topology.

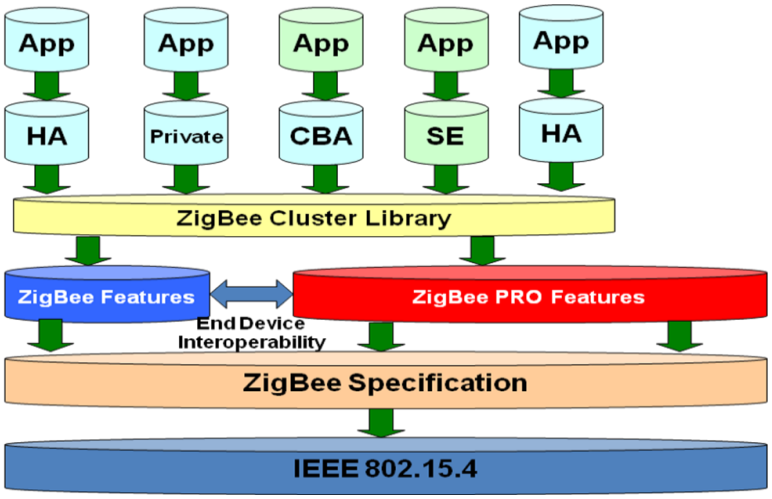

Figure 1: ZigBee Application Architecture

On top of the two stack featuresets (or 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.

Also, many custom application profiles are in use today and it is possible for a custom application profile to become public given that three member companies of the ZigBee Alliance use it and publicise it.

To prevent every profile task group from ending 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.

2.2 Device Addressing

Each ZigBee node has a unique 64-bit EUI64, which is effectively the device's "MAC Address". These unique numbers are assigned by the IEEE and consist of a 24-bit manufacturer ID, followed by a 40-bit device ID. Telegesis nodes have an EUI64 starting with either "0021ED…" (which is Telegesis's IEEE assigned manufacturer ID) or "000D6F…" (which is Ember's).

In order to keep the ZigBee protocol headers small and also to minimize the memory requirement of the ZigBee stacks' internal addressing tables each ZigBee node gets assigned a random 16-bit ID when joining a ZigBee PAN (Personal Area Network). This 16-bit ID is also referred to as NodeID. As the NodeIDs are assigned randomly there is a potential for conflicts, which are detected and resolved automatically by the stack.

It is also this 16-bit NodeID which sets the theoretical limit of 64000 nodes per ZigBee network.

2.3 Endpoints

Now that we know that an individual ZigBee node can be addressed with either its 64-bit EUI64, or its NodeID we can go one level deeper.

Every physical ZigBee radio can potentially be used for more than just a single application. For example it is possible to have a dimmer, which also acts as a temperature sensor at the same time. In this case it must be possible when interacting with this ZigBee node to address either the dimmer, or the temperature sensor.

This is achieved by assigning the two functionalities to two separate endpoints. In our example the endpoint list could look as follows:

Endpoint	Functionality
0	ZDO (ZigBee Device Objects)
1	Dimmer
5	Temperature Sensor

Table 1: Endpoint List

On all ZigBee Devices endpoint 0 has to implement the ZigBee ZDO (ZigBee Device Objects). The ZDO offers functionality used to identify the node and its neighbours in the network, to identify its endpoints and capabilities as well as to commission the node in a network. Following this additional addressable endpoints can be added. There are no rules regarding the numbering scheme of the endpoints, but it can be seen that most implementations prefer to count upwards

linearly, so in the example above the Temperature Sensor would be assigned to endpoint 2 by most implementers.

When using the ZDO to get additional information about an endpoint, the endpoint description contains the following information:

Application Profile ID → This could be HA, CBA, SE, or a non-public application profile ID

Device ID → E.g. if the application profile was HA, this could identify a light switch

Input Clusters → Inbound requests the device can understand

Output Clusters → Outbound messages device can send

2.4 Clusters

As mentioned before the ZigBee Alliance has defined a cluster library (which can be downloaded from www.zigbee.org) containing clusters defining all the functionality needed by any of the public application profiles.

An application profile basically consists of a subset of clusters picked from this cluster library. An in depth description of the two released application profiles (HA and SE) can also be downloaded from (www.zigbee.org)

Finally an actual device like a HA light switch again has to implement a subset of the clusters used by the HA application profile.

In addition to this, any device may implement additional custom clusters allowing for functionalities beyond what has been defined for this device in the specification, however this additional functionality can only be used in case a remote end is capable of interacting with this custom cluster.

3 The ZDO

The ZDO is used to commission and maintain a ZigBee network. To date the Telegesis AT-Commandset already supports a number of commands utilising the ZDO, including

AT+NTABLE → List Neighbour Table

AT+DASSR → Remove Remote Device from the PAN

AT+FNDSR → Fins a source route to the remote device

For example to get a list of all devices on the network (including 3rd party devices) in a ZigBee compliant way most ZigBee Network analyzer tools as well as commissioning tools start up by requesting the coordinator's neighbour table, followed by the coordinator's neighbours' neighbour table and so on, thus forming a map of the entire network.

Support for additional ZDO specific AT-Commands will be expanded in future releases of the firmware as upcoming releases of the Ember Compiler will allow a higher code efficiency.

In addition to using ZDO specific AT commands the ZDO of both the local as well as a remote node can be accessed as described in the next chapter.

4 Interacting with a ZigBee PRO device

From the background information given in the previous chapter it has become clear that in order to interact with a ZigBee PRO compliant device the following has to be undertaken (assuming we are already in the same PAN etc., see the user guide for more info on network establishment and security).

1. The Address of the remote node (EUI64 or 16-bit NodeID)
2. The remote Endpoint Number
3. The Profile ID
4. The Cluster ID

All these parameters can be set using the Telegesis AT-Command firmware, therefore allowing communication with every 3rd party ZigBee PRO compliant device.

Furthermore, steps must be taken to handle messages that arrive from a node using a non-Telegesis profile. If these conform to the format of our normal R3xx firmware then they can be directed to endpoint 1 and handled by our normal interpreter, in which case no changes to the local configuration are needed. It is more likely, though, that the incoming messages will conform to a different profile, in which case the Telegesis firmware will be unable to interpret them. Instead they can be directed to the serial port without modification for interpretation by a host processor. To enable this it is necessary to configure two S-register bits:

- S0AB=1 Allows Endpoint 2 to reply to ZDO endpoint queries
- S0F1=0 Enable showing unhandled messages received by Endpoints 0 and 2

(These are the settings required by the firmware in current use, namely R305. Full details can be found in the AT Command Manual.)

4.1 Endpoints of the AT-Command firmware

The Telegesis AT-Command firmware has the following endpoints:

Endpoint	Functionality
0	ZDO (ZigBee Device Objects)
1	Telegesis AT-Commandset
2	transparent

Table 2: Endpoint List

Endpoint 1 supports the Telegesis AT-Commandset as a custom application profile, whereas endpoint 2 is transparent in that every message arriving at endpoint 2 will be displayed to the host microcontroller transparently (if enabled).

In future releases of the firmware it is planned to make endpoint 2 configurable to a point in which the profile ID and cluster lists can be defined by the host processor.

4.2 Example: Interacting with a HA On/Off Light

The HA On/Off Light is part of the ZigBee Home Automation Profile (Profile ID 0x0104, DeviceID 0x0100) and it implements the server side of the following clusters –

- Basic Cluster – Cluster ID 0x0000
- Identify Cluster – Cluster ID 0x0003
- Groups Cluster – Cluster ID 0x0004
- Scenes Cluster – Cluster ID 0x0005
- On/Off Cluster – Cluster ID 0x0006

The AT commands listed in this document are for setting the attributes of the On/Off cluster.

The On/Off cluster has the following attributes:

- ZCL_ON_OFF_ATTRIBUTE - Attribute ID 0x0000 defines the current state of the light and supports the following commands –
- ZCL_OFF_COMMAND – Command Id 0x00 defines the command to switch off the light
- ZCL_ON_COMMAND – Command Id 0x01 defines the command to switch on the light
- ZCL_TOGGLE_COMMAND – Command Id 0x02 defines the command to toggle the light

In order to interact with the HA compliant On/Off light we need to do the following:

1. Find the node's address using ZDO commands (either EUI64 or NodeID)
2. Set the profile ID for outgoing messages to the HA application profile ID 0x0104 by using the AT command: **ATS44=0104**. By default this register is set to 0xC091, which is the custom profile ID of the AT command firmware
3. Set the source and destination endpoints for the transmission in register S40. The source endpoint should be selected to be endpoint 2 as any potential response will then be displayed transparently
4. Set the target cluster ID in S42 to 0x0006 (On/Off Cluster) by using the AT command: ATS42=0006
5. Send the commands (On, Off, Toggle) to the HA On/Off light to set the ZCL_ON_OFF_ATTRIBUTE

This is achieved by sending a unicast to the target node with three bytes of data –

<ZCL Frame Control> <ZCL Sequence No><Command>
ZCL Frame Control is set to 1 indicating the command is Cluster Specific
ZCL Sequence No is set to a value between 0 and 255

Command is set to 0 for ZCL_OFF_COMMAND, 1 for ZCL_ON_COMMAND and 2 for ZCL_TOGGLE_COMMAND

(i) Send ZCL_ON_COMMAND to the HA light to set the ZCL_ON_OFF_ATTRIBUTE to ON by using the AT command – **AT+UCAST: <address>,<b1 b0 b1>**

(ii) Send ZCL_OFF_COMMAND to the HA light to set the ZCL_ON_OFF_ATTRIBUTE to OFF by using the AT command – **AT+UCAST: <address>,<b1 b0 b0>**

(iii) Send ZCL_TOGGLE_COMMAND to the HA light to toggle the ZCL_ON_OFF_ATTRIBUTE by using the AT command – **AT+UCAST: <address>,<b1 b0 b2>**

The difficulty in preparing this document is that the binary payload cannot be represented in a printable form. It may be easier to reverse the process and exemplify the ZCL_TOGGLE_COMMAND in hexadecimal. Assume (for brevity) that we want to send to the device identified by node ID 5A6B. The command then becomes

```
41 54 2B 55 43 41 53 54 3A 35 41 36 42 3D 01 00 02 0D

 A  T  +  U  C  A  S  T  :  5  A  6  B  =  □  □  □ <cr>
```

Please note that the address of the target node is entered using ASCII characters, whereas the message payload needs to be entered in binary. Telegesis Terminal currently does not support entering binary data; the best way to work around this is to prepare text files in an editor, which contain the desired commands. These can be sent to the command line by right clicking on the terminal and selecting "send text".

For more information on the individual clusters and their attributes please refer to the ZigBee cluster library and the application profile descriptions obtainable from www.zigbee.org.

5 Conclusion

The transparency build into the Telegesis AT-Command firmware allows easy interaction with 3rd party ZigBee compliant devices.

Support for transparency and ZDO specific AT-Commands will be expanded in future releases of the firmware as upcoming releases of the Ember Compiler will allow a higher code efficiency.

In addition to using the Telegesis AT-Command firmware there are additional options for using the Telegesis range of ZigBee modules to interact with 3rd party ZigBee compliant devices:

- Using the Ember EZSP (Ember ZigBee Serial Protocol) on the Telegesis Hardware. This binary protocol allows a lower level access to the Ember stack, but is not human readable and takes more time to implement.
- Developing custom firmware using the Ember ZigBee development toolchain
- Commissioning Telegesis to do the above for your project.

6 Trademarks

All trademarks, registered trademarks and products names are the sole property of their respective owners.

7 Disclaimer

Product and Company names and logos referenced may either be trademarks or registered trademarks of their respective companies. We reserve the right to make modifications and/or improvements without prior notification. All information is correct at time of issue. Telegesis (UK) Ltd does not convey any license under its patent rights or assume any responsibility for the use of the described product

8 Contact Information

Website: www.telegesis.com

E-mail sales@telegesis.com

Telegesis (UK) Limited
Abbey Barn Business Centre
Abbey Barn Lane
High Wycombe
Bucks
HP10 9QQ
UK

Tel: +44 (0)1494 510199
Fax: +44 (0)5603 436999

9 References

Telegesis - www.telegesis.com

Ember - www.ember.com