ETRX2 and ETRX35x 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 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.

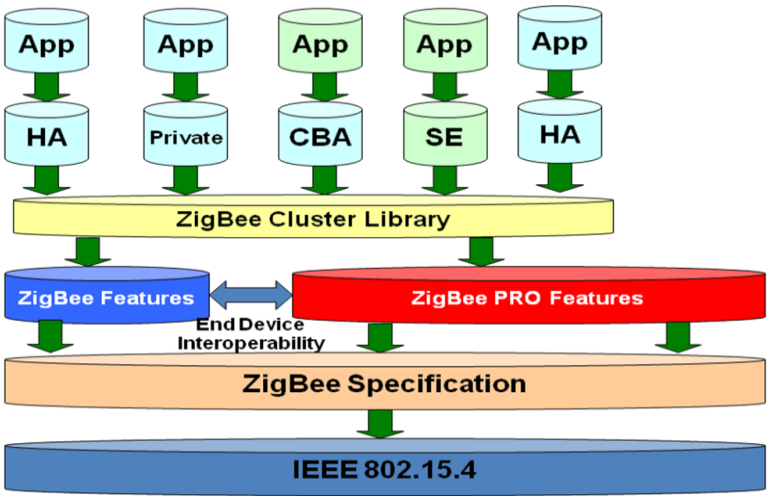


Figure 1: ZigBee Application Architecture

On top of the two stack profiles, the ZigBee Alliance specifies several application profiles. An application profile uses a defined sub-set of the messages defined in the ZigBee cluster library [3] plus some message clusters specific to that profile, and defines various classes of device for the purpose of service discovery. The current public profiles include:

- ZigBee Building Automation
- ZigBee Health Care
- ZigBee Home Automation
- ZigBee Input Device
- ZigBee Network Devices
- ZigBee Remote Control
- ZigBee Retail Services
- ZigBee Smart Energy
- ZigBee Telecom Services
- ZigBee 3D Sync

Also, many custom application profiles are in use today and it is possible for a custom application profile to be declared public by the ZigBee Alliance if three member companies of the alliance use it and publicise it.

To prevent every profile task group from re-inventing the wheel the ZigBee cluster library [3] 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.

The basic ZigBee specification, the ZigBee cluster library and the various public profile specifications can all be downloaded from the ZigBee Alliance's website.

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 is assigned a random 16-bit ID when joining a ZigBee PAN (Personal Area Network). This 16-bit ID is also referred to as the 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 Object)
1	Dimmer
5	Temperature Sensor

Table 1: Example endpoint list

2.4 Clusters

In a ZigBee command message, the command itself is a single one-byte code often followed by some parameters. This only allows for 256 different commands whereas ZigBee systems have far more functions than that, so each message also includes a cluster ID which defines the context in which the command is to be interpreted. A cluster is a set of related commands plus a set of attributes that describe the state of the device (more specifically, the state of the virtual device residing on the relevant endpoint). For example, in cluster 0x0006 (the On/Off cluster), attribute 0x0000 is the on/off status of the controlled device and command 0x01 is “switch on”. In cluster 0x0009 (the Alarm cluster), on the other hand, attribute 0x0000 is the alarm information and command 0x01 is “reset all alarms”.

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. The specification of a profile defines which clusters each class of device shall and may support, so that an actual device like a HA light switch 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 when a remote end is capable of interacting with this custom cluster.

3 The ZDO

On all ZigBee devices endpoint 0 has to implement the ZigBee Device Object (ZDO). 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 many implementers.

The ZDO contains information about the hardware platform and the firmware capabilities, which can be interrogated remotely using ZigBee Device Profile messages from the ZigBee Device Cluster (defined in the ZigBee specification [2]). Several of these requests can be made using AT commands, which in the R308 firmware [1] are:

AT command	Function	ZDP command
AT+IDREQ	Request Node's NodeID	NWK_addr_req
AT+EUIREQ	Request Node's EUI	IEEE_addr_req
AT+NODEDESC	Request Node's Descriptor	Node_Desc_req
AT+POWERDESC	Request Node's Power Descriptor	Power_Desc_req
AT+ACTEPDESC	Request Node's Active Endpoint List	Active_EP_req
AT+SIMPLEDESC	Request Endpoint's Simple Descriptor	Simple_Desc_req
AT+MATCHREQ	Find Nodes which Match a Specific Descriptor	Match_Desc_req
AT+ANNCE	Broadcast a device's presence	Device_annce
AT+CCHANGE	Change the network's channel	Mgmt_NWK_Update_req
AT+NTABLE	Read neighbour table	Mgmt_Lqi_req
AT+RTABLE	Read routing table	Mgmt_Rtg_req
AT+DASSR	Disassociate remote device	Mgmt_Leave_req

Table 2. ZDO commands

For example, the Active Endpoint List returns the numbers of the endpoints supported in a device's firmware; for a device with R308 these will be 0 (the ZDO), 1 (the Telegesis message interpreter) and possibly 2 if activated. Interrogating each endpoint in turn will reveal:

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 → Clusters supported on the server side of the device

Output Clusters → Clusters supported on the client side of the 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.

4 Interacting with a ZigBee PRO device

4.1 Introduction

From the information given in the previous chapter it is clear that in order to interact with a ZigBee PRO compliant device the following information must be known:

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 or more 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
- S0F8=0 Show unhandled messages received by Endpoints 3 and above

(These are the settings required by the firmware in current use, namely R308. Full details can be found in the AT Command Manual.)

4.2 Network creation

A device with R3xx firmware can easily be commanded to establish a new network by sending the AT+EN command, or it can be commanded to join a network with the AT+JN or AT+JPAN commands, and details of all the AT commands can be found in the appropriate R3xx AT Command Manual. However, the network security settings must be consistent with the settings of the other devices in the network. There are three levels of security commonly found in ZigBee networks:

1. Residential security. Messages are encrypted but the network key is sent to new devices unencrypted. Set register S0A=0000 for this security level.
2. Commercial security. The coordinator sends the network key to new devices but encrypts it with the link key, therefore all devices must have the link key stored in register S09. The simplest setting for this level is S0A=0114, but refer to appendix A of the AT command manual for more details.
3. Use of an application layer link key. This is required in ZigBee Smart Energy networks but is not supported by the R3xx firmware.

4.3 Home Automation networks

The ZigBee Alliance has defined a link key value that is used in all Home Automation devices, and by some of the other public profiles as well. Its value is 0x5A6967426565416C6C69616E63653039 which is the hexadecimal equivalent of "ZigBeeAlliance09". In addition they recommend that a subset of the 16 ZigBee channels be used to avoid the commonly-used WiFi bands; register S00 contains the channel mask. Therefore to use the ETRX357 in an HA network use the following commands:

4.3.1 To create a network

ATS00=6314

ATS0A=0914;password (or "ATS0A=0114;password" if endpoint 2 is not needed)

ATS09=5A6967426565416C6C69616E63653039;password

AT+EN

4.3.2 To join a network as a router

ATS0A=0914;password (or "ATS0A=0114;password" if endpoint 2 is not needed)

ATS09=5A6967426565416C6C69616E63653039;password

AT+JN or AT+JPAN:<channel>,<PAN ID>

4.3.3 To join a network as a Sleepy End Device

ATS0A=4914;password (or "ATS0A=4114;password" if endpoint 2 is not needed)

ATS09=5A6967426565416C6C69616E63653039;password

AT+JN or AT+JPAN:<channel>,<PAN ID>

4.4 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
3+	Transparent

Table 2: Endpoint List

Endpoint 1 supports the Telegesis AT-Commandset as a custom application profile, whereas endpoints 2 and above are transparent in that every message arriving at these endpoints will be displayed to the host microcontroller transparently (if enabled). Only endpoint 2 can have its own profile ID and cluster lists, and respond to ZDO queries from other devices. Bit B of register S0A activates endpoint 2, and values can be written into registers S48 through to S4C to define the profile ID, device ID, device version, input clusters and output clusters.

4.5 Example 1: 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 0x01 indicating the command is Cluster Specific

ZCL Sequence No is set to a value between 0 and 255. Sequence numbers allow a response from the remote device to be matched up with the command, and it is not normally essential to use them incrementally

Command is set to 0x00 for ZCL_OFF_COMMAND, 0x01 for ZCL_ON_COMMAND and 0x02 for ZCL_TOGGLE_COMMAND

Send ZCL_ON_COMMAND to the HA light to set the ZCL_ON_OFF_ATTRIBUTE to ON by using the AT command:

AT+UCASTB:03,<address>
<0x01 0x00 0x01>
Send ZCL_OFF_COMMAND to the HA light to set the ZCL_ON_OFF_ATTRIBUTE to OFF by using the AT command:AT+UCASTB:03,<address>
<0x01 0x00 0x00>

Send ZCL_TOGGLE_COMMAND to the HA light to toggle the ZCL_ON_OFF_ATTRIBUTE by using the AT command:

AT+UCASTB:03,<address>
<0x01 0x00 0x02>

In each case the payload within <…> must be three binary characters, not ASCII text, and the user can choose a suitable sequence number. It is often possible to use the command AT+UCAST instead of AT+UCASTB, but the former cannot send the character 0x0D.

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. When using the latest version of Telegesis Terminal, you can switch to Tools→Use Nonprinting Chars and enter a typical payload with the format <00><01><02>. Alternatively 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". You can also use a terminal application such as HyperTerminal, PuTTY or Docklight which can handle binary characters easily.

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.

4.6 Example 2: Sending a Discover Attributes command

Example 1 was a command specific to a particular cluster, but the ZigBee Cluster Library also has a set of commands that operate across all the clusters; these are used to read and write the various attributes and control their reporting, and here the cluster ID is used to indicate which set of attributes are being addressed.

As an example, we can use the Discover Attributes command 0x0C. Many attributes are optional so this command causes a device to reply with a list of all the attributes that it actually supports. From the ZigBee Cluster Library specification:

2.4.13.1 Discover Attributes Command Frame Format

The discover attributes command frame shall be formatted as illustrated in Figure 2.26.

Octets: Variable	2	1
ZCL header	Start attribute identifier	Maximum attribute identifiers

Figure 2.26 Format of the Discover Attributes Command Frame

In this example we enquire the attributes of a Home Automation device that supports the metering cluster. As before you need to set some S-registers first:

```
ATS0F=0104      ; reveal messages on endpoints 0, 2 and above
ATS40=020A      ; source and destination endpoints, 0A to suit the device being questioned
ATS42=0702      ; cluster ID - metering
ATS44=0104      ; profile ID - Home Automation
```

Endpoint 2 is used for convenience. We will request a maximum of 8 attributes starting at zero, so with an arbitrary sequence number the frame is:

ZCL header			ZCL payload	
Frame control	Sequence number	Command	Start attribute identifier	Maximum attribute identifiers
0x00	0x04	0x0C	0x0000	0x08

Table 3. Example Discover Attributes

A suitable AT command is then

AT+UCASTB:06,<destination address>

followed by the data

00 04 0C 00 00 08

To repeat what we have already said, the data must be sent to the R3xx firmware as binary characters - you must not send ASCII text “00 04 0C …”. When this message was sent to an actual device, an RX reply was received followed by a sequence of binary characters:

RX:313E,0104,02,0A,1C:

18 04 0D 00 00 02 18 00 03 01 03 22 02 03 22 03 03 18 00 04 2A

They start with the usual header 0x18040D, where 0x18 is the frame control, 0x04 the sequence number and 0x0D the command “Discover Attributes Response”, and the rest describes a series of attributes.

5 Conclusion

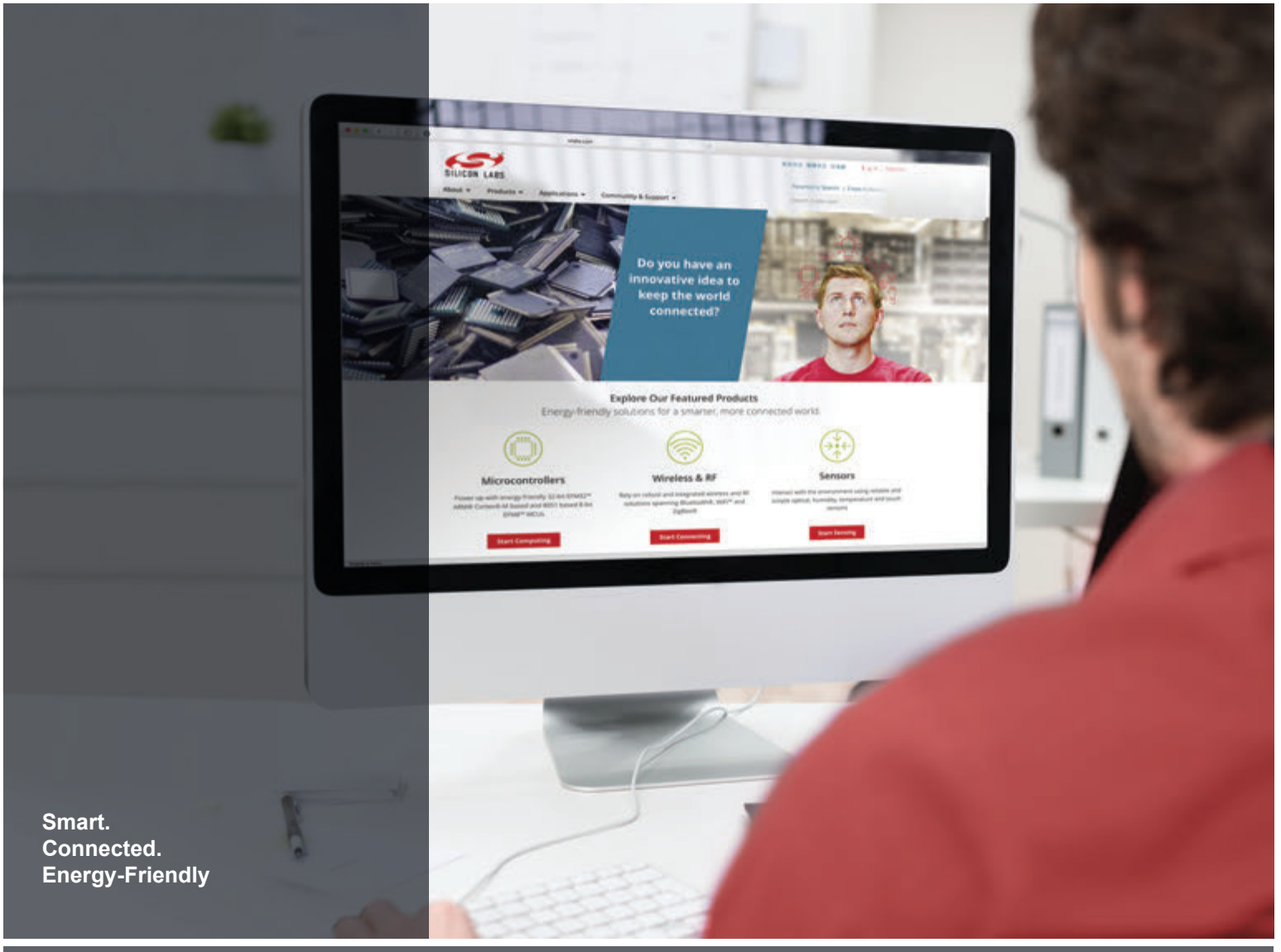
The transparency build into the Telegesis AT-Command firmware allows easy interaction with 3rd party ZigBee compliant devices.

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 InSight ZigBee development toolchain
- Commissioning Telegesis to do the above for your project.

6 References

1. R308 AT Command Manual
2. ZigBee specification (ZigBee Document 053474r17)
3. ZigBee cluster library specification(ZigBee Document 075123r03ZB)



Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS®

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com