FlexiPanel

ZigBee for Applications Developers

Bluetooth is fine for getting rid of that junk of cables between personal devices. Wi-Fi manages the wireless ethernet around our homes quite nicely. WiMax promises to do the same for entire cities. So do we really need another wireless protocol? The ZigBee Alliance seems to think so, and here's why: wireless should mean wireless. Mains leads and power cables are not invited.

No strings attached

The *wireless-should-mean-wireless* attitude has had a profound effect on every layer of the wireless architecture. Most noticeably, mesh and tree networking are allowed in addition to the star networks used by Bluetooth and Wi-Fi. This means that if you're not in range of the *node* (i.e. the ZigBee radio transceiver) you want to communicate with, you can ask other nodes in between to pass a message along. Since halving the range of a radio reduces radio power consumption by 75%, this is vital to achieving dramatic power reductions without compromising range. It has the side-effect of adding fault tolerance, too, since a message can be routed ad-hoc. (Figure 1.)

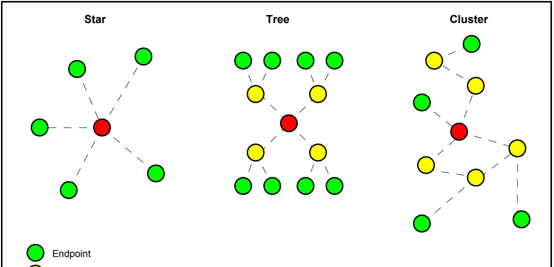

Figure 1. Bluetooth and Wi-Fi communicate in star networks. Using routers, ZigBee also allows tree and cluster networks, reducing power consumption and increasing range

The second noticeable difference is that there are three distinct classes of node. At the bottom are the *end devices* which probably represent the vast majority of nodes. The ZigBee architecture ensures that these are as dumb as possible and can spend most of their time in sleep mode. This ensures they don't cost much and they don't need their batteries replaced for years. Each end device can have up to 240 *end points* which are separate applications sharing the same radio. For example, a three-gang light switch would have three distinct end points sharing the same radio electronics and battery.

At mid-level are *routers*. They must have the ability to stack up messages for, and respond to general enquiries about, end devices in their vicinity that are asleep. They must also figure out the best way to pass on a message to a nodes that is not in range.

At the top of the pile is the *coordinator*. It has to be always on, so it is the one node that always needs a good power source. In addition to being a router, it sets the rules of the network, finds a free frequency channel to operate on, etc. Practically speaking, routers and coordinators will most likely be the same physical devices and a jumper switch will allow you to switch between the two.

Binding, KVPs and messages

The coordinator also manages the *Key Value Pair* service. In ZigBee networks, data is abstracted as much as possible into *Key-Value Pairs* (KVPs). The coordinator has a master look-up table known as a *binding table* that lists which nodes are interested in a particular KVP. This not only keeps the message passing simple, it saves the end points from knowing who they need to send messages to. For example, a light switch only has to tell the coordinator it wants to modify a KVP from off to on. It is the coordinator and/or router's job to work out whether to then send an 'on' message to the bedside light or the runway lights at Heathrow. (Figure 2).

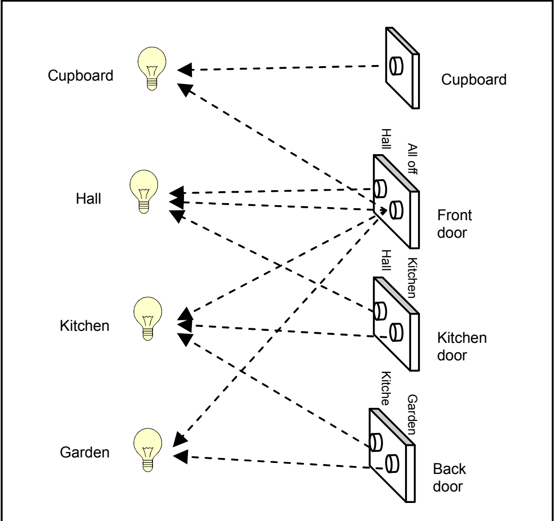

Figure 2. Binding connects compatible ZigBee endpoints to each other. The coordinator stores the *binding table.*

The process of securely connecting different end points to each other is known as binding. Typically, an end-user will 'bind' two end points with compatible KVPs by pressing a recessed button on each unit at the same time. In secure networks authentication and encryption processes will additionally be taking place under the hood, but in a well designed ZigBee network the installer would not be aware of this.

Once two nodes are bound, the coordinator will ensure that any KVP message generated by one unit is forwarded correctly. Nodes should be able to know how to send and receive four basic types of KVP message: *Get* to enquire a KVP value, *Get Response* to reply to a *Get*, *Set* to modify a value and *Event* to signal that a particular event has happened. Each type of message also has a corresponding acknowledge message, and they all have XML schemas defined to provide a seamless bridge to the internet.

Some applications may not suit the KVP service, so a free-form message passing service is also provided. End points do not need to be bound in order to send messages to each other.

ZigBee, Bluetooth or Wi-Fi?

ZigBee, Bluetooth and Wi-Fi are all relatively new wireless technologies. The key comparison points are shown in table 1. Essentially, ZigBee is best suited for low-cost, low data rate and battery powered applications. Point-to-point range is limited, but mesh networking provides an alternative stepping-stone approach to communicating over large distances.

Where a PC or cellphone is involved, there is some overlap between Bluetooth and ZigBee. For example, either technology could be used for activating your garden lights from your cellphone. ZigBee can do the job more cheaply, but Bluetooth is more likely to be compatible with the phone you have in your pocket today. Where you don't want to go with ZigBee is time-critical, high data rate applications such as audio and video links.

Table 1 – Comparison of wireless technologies

	Wi-Fi	Bluetooth	ZigBee
Frequency bands	2.4GHz	2.4GHz	2.4GHz, 868 / 915 MHz
Stack size	~1Mb	~1Mb	~20kb
Raw data rate	11Mbps	1Mbps	250kbps (2.4GHz) 40kbps (915MHz) 20kbps (868MHz)
Number of channels	11 – 14	79	16 (2.4GHz) 10 (915MHz) 1 (868MHz)
Data types	Digital	Digital, Audio	Digital, Key-Value Pairs
Inter-node range	100m	10m – 100m	10m – 100m
# of devices	32	8	255 / 65535
Power requirements	Medium – hours on one battery	Medium - days on one battery	Very low – years on one battery
Current market penetration	High	Medium	None
Architectures	Star	Star	Star, Tree, Cluster
Best applications	Internet inside buildings	Computer & phone peripherals	Low-cost control and monitoring

Applications and profiles

For each type of ZigBee application, *clusters* of KVPs are defined to achieve particular task. These are grouped together to form a *profile* for the application. For common applications, "public" profiles are available so what you make will immediately interoperate with other manufacturers' products. For example, the Home Control – Lighting profile includes clusters for turning lights on an off, and for setting dimming level.

At time of writing, profiles exist for Home Control (e.g. lighting, heating), Building Automation (e.g. utilities, energy monitoring within building) and Plant Control (e.g. configuring, controlling and monitoring shop-floor machinery). You can define your own profiles if the existing ones do not suit your needs, and choose whether to keep them proprietary or make them public so other developers can make compatible devices.

The ZigBee Alliance is keen for people to develop profiles public in their specialist areas. For example at FlexiPanel Ltd we see a clear demand for RS232 and RS485 cable replacement applications and are in the process of ensuring public profiles are made available for them. However, to propose a public profile requires paying participant-level membership fees of $9500/year. This would be prohibitive for developers of a 'long tail' of applications, from model railway controls to pot-plant moisture monitoring, and seems a little short sighted of the ZigBee Alliance.

The ZigBee stack

ZigBee, like other communications networks such as Ethernet, IrDA and Bluetooth, divide up the communications tasks into layers on a stack loosely based on the 7-layer Open Systems Interconnection model. At the top is the application that uses the data; at the bottom is the actual transmission medium such as coax, infrared or microwaves. In the middle is all the glue so that applications on different nodes can communicate securely and reliably without needing to know anything's going on in between, or how the data got from place to place. The ZigBee standard is mostly concerned with the higher layers of the stack; lower layers adopt the IEEE 802.15.4 protocol.

Even if you intend to use an off-the-shelf ZigBee module, it's worth knowing a little about what-goes-on-where in the stack so you can troubleshoot when having difficulties and to understand all the tricks it has hiding up its sleeve.

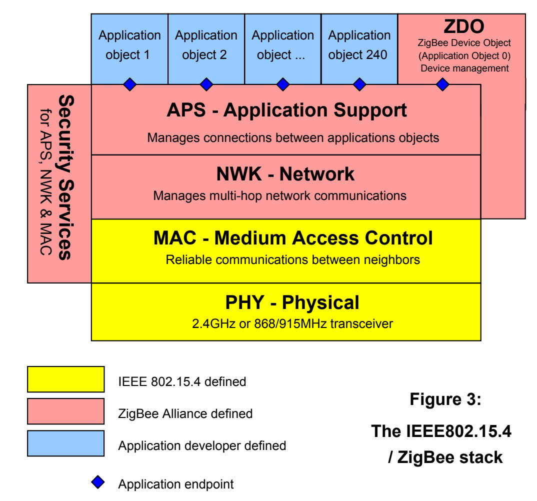
Application object 1
Application object 2
Application object ...
Application object 240
ZDO
ZigBee Device Object (Application Object 0) Device management
Security Services
for APS, NWK & MAC
APS - Application Support
Manages connections between applications objects
NWK - Network
Manages multi-hop network communications
MAC - Medium Access Control
Reliable communications between neighbors
PHY - Physical
2.4GHz or 868/915MHz transceiver
IEEE 802.15.4 defined
ZigBee Alliance defined
Application developer defined
Application endpoint

Figure 3: The IEEE802.15.4 / ZigBee stack

Referring to figure 3, the basic tasks of each layer in the stack are as follows:

Physical Layer (PHY): The PHY layer consists of a half duplex radio transceiver operating at 868MHz, 915MHz or 2.4GHz. 868MHz is license-free in Europe, whereas 915MHz is license-free in North America and Australia. 2.4GHz may be used practically worldwide, and so is expected to dominate in future. It also has the greatest number of channels available.

PCB design for 2.4GHz is definitely not for beginners, so you really should consider buying a radio module, complete with integral antenna, even if you feel confident enough to develop the rest yourself. Most radio modules come with FCC and CE certification, include the MAC layer, and are based around chips such as the CC2420 (also marketed as EM2420) or the ZMD44101.

Medium Access Control (MAC): The MAC layer provides reliable communications between a node and its immediate neighbors. One of its main tasks, particularly on a shared frequency channel, is to listen for when the channel is clear before transmitting. This is known as Carrier Sense Multiple Access – Collision Avoidance communication, or CSMA-CA. In addition, MAC can provide beacons and synchronization to improve communications efficiency.

The MAC layer also manages packing data into frames prior to transmission, and then unpacking received packets and checking them for errors.

Network Layer (NWK): The NWK layer provides the routing and multi-hop capability required to turn MAC-level communications into a full star, tree or mesh network. For end devices, this amounts to little more than joining and leaving the network. Routers also have to be able to forward messages, discover neighboring devices and build up a map of the routes to other nodes.

In the coordinator, the NWK layer can start a new network and assign network addresses to new devices when they join the network for the first time.

Application Support Layer (APS): The APS layer routes messages on the network to the different application end points running on the node. This includes maintaining the binding tables and forwarding messages between bound devices. If you develop a ZigBee product by adding application objects to a ZigBee stack, you will mostly be making function calls to the APS layer.

Applications Objects: An application object is the software at an end point which achieves what the device is designed to do. You develop a ZigBee product by creating application end points on top of the ZigBee stack.

ZigBee Device Object (ZDO): The ZDO is responsible for overall device management, and security keys and policies. If you develop a ZigBee product by adding application end points to a ZigBee stack, you may well make calls to the ZDO in order to discover other ZigBee devices on the network and the services they offer, to manage binding and to specify security and network settings.

The ZDO is like a special application object that is resident on all ZigBee nodes. It is always end point zero, and your application end points are then numbered 1 to 240. It has its own profile, known as the ZigBee Device Profile (ZDP), which your application end points and other ZigBee nodes can access. It is the ZDP that contains the services for device discovery, etc.

Security Services: The ZigBee architecture provides security services for establishing and exchanging security keys, and using these keys to secure the communications. The security services don't fit neatly into a single layer of their own; they are services used by the MAC, NWK and APS layers to provide security at each level. The general rule is that the layer responsible for generating a frame of data is responsible for encrypting it when it is generated and authenticating it when it is received.

The ZDO dictates the security policies and configurations implemented by the security services. If you develop a ZigBee product by adding application end points to a ZigBee stack, you will negotiate with the ZDO to specify the security settings required; security will then be more or less transparent to you.

Buy or build?

If you want to go ahead and develop your own ZigBee projects and products, should you do it all yourself, or should you buy some of the technology in the form of ready made modules?

The short answer to this is it depends on how standard your application is. For many applications such as switching and serial communications, you can or soon will be able to buy products off-the-shelf with very simple electrical specifications and you won't need to understand anything about ZigBee. If you think you will need to develop your own application profile, or if you have other tasks which you would like the microcontroller running the stack to be doing, you will want a module that provides some or all of the lower layers of the stack.

FlexiPanel Ltd, amongst others, is developing FCC- and CE-qualified modules for developers, complete with integral antennae. Its EZBee module offers just the PHY/MAC layers, whereas the PICZee product contains the entire stack up to APS/ZDO. (Figure 4.)

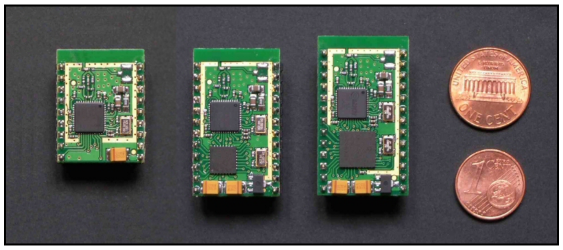

Figure 4. EZBee and PICZee products

PICZee is based on the ZigBee stack Microchip Technology, so full source code and evaluation kit are available, and FlexiPanel is developing ZigBee-qualified application-specific versions including home control profile and serial / USB communications.

The weird wireless web

A few killer applications such as lighting and power control are likely to ensure that ZigBee networks soon permeate our homes, offices and factories. With a ZigBee backbone in place, a wide variety of low cost battery powered applications can hitch a free ride on the network. I regularly get technical support calls from people who want to try things that they would never have considered before, such as a light switch in the hallway that turns off all the lights in the house, or a paging network in restaurants to summon a waiter. Here's my prediction: one day your kids are going to ask you how you lived without ZigBee.