Adding comm

Getting access to connectivity has never been easier. By **Philip Ling**.

The concept of 'communications' can leave some people cold; it conjures up an image of dusty and verbose standards so convoluted that they flow about as well as treacle on a cold day.

For some standards, this isn't far from the truth. But, in most instances, it's simply down to necessity; for a standard to fulfil its obligations, it often needs to be dictatorial about its implementation and to cover all contingencies.

Reality is that all embedded devices feature some kind of communication – even if it's a proprietary standard confined to the boundaries of the design. So *implementing* a standard isn't really the problem: it's *interpreting* it.

Misinterpreting a standard is worse than not implementing it in the first place, particularly when the device implementing the standard needs to pass independent qualification. It's here that commercial solutions can help and a definite trend in the past few years has been to offer prequalified modules comprising both hardware and software for a particular specification. A good case in point is Bluetooth, which is now available in the form of fully integrated modules and, increasingly, single chip solutions.

A typical application where value is being added by supplying compliant solutions is wireless connectivity. From gsm and gprs, to WiFi and – increasingly – WiMAX, there is a market for plug 'n' play wireless modules. Simply put, interpreting a standard doesn't add as much value to an end product as implementing it, so the trend here is to buy off the shelf, certified solutions for integration into larger products.

Distilling this model, though, will inevitably identify developers who, for one reason or another, are unable to use off the shelf modules and who will, therefore, need to implement the appropriate communication specification themselves.

For companies that weald a lot of purchasing power over silicon vendors, this isn't necessarily a problem, as they are likely to receive all the support they need. For smaller companies, or for production runs with smaller volumes, that support may not be quite so forthcoming.

For users of commercial code bases typically founded on an operating sys-

tem, the picture isn't quite so bleak. The Green Hills Platform for Wireless Devices (PWD), for instance, comprises middleware for WiFi and WiMAX enabled devices. It's available in four configurations, providing all the software technologies required to implement: WiFi end points; WiFi access points; WiMAX base stations; and mobile stations. Configurations are offered as fully integrated and tested packages (for more, see **www.ghs.com**).

The WiFi configuration, for example, supports a number of Conexant chipsets, with support for devices from other vendors available on request. However, WiMAX drivers need to be provided by the respective silicon vendors. Green Hills also offers preloaded reference platforms for development – for both WiFi and WiMAX devices – based on ARM/XScale, Coldfire, PowerPC or Blackfin silicon.

The deployment of WiFi devices is making it even more cost effective for the

s value

embedded domain in general. By way of example, Iosoft (**www.iosoft.co.uk**) offers an 802.11b, solution using a standard wireless PCMCIA that is driven by a PIC micro. The development kit comprises the hardware – including a PIC18F452 and a wireless PCMCIA card – and the software in source code, supporting 64 and 128bit WEP encryption and built on Iosoft's ChipWeb TCP/IP stack, also included.

But what of standards that don't require independent qualification? In this case, commercial demand for prequalified modules isn't as high and, subsequently, neither is the availability of modules. Instead it's often left to the manufacturer to interpret and implement the standard. The complexity of the standard isn't necessarily any lower, however, so what's available for those developers who don't have the time or resources to sit down and digest complex specifications?

For less demanding devices that aren't built on a commercial operating system, but which still need some element of standardised communication, there are some portable implementations of protocol stacks available commercially, most noticeably in wired connectivity.

TCP/IP ubiquity

Most apparent is the ubiquitous TCP/IP, which is nowadays even available for 8051 compliant devices. Developed by Ceibo, it's distributed in the UK by Great Western Microsystems (**www.gwmicros.com**), as are similar products from Kadak. It supplies a range of embedded networking technologies, called KwikNET, which includes a web server, email (POP3/SMTP) clients, FTP, IPsec and others. It also offers Ethernet device drivers for a range of hardware interface devices.

Likewise, Reveal (**www.reveal.co.uk**) is the UK distributor for NexGen Software, which offers a range of embedded networking solutions that claim to be operating system independent. This is thanks to NexGenOS – its platform independent wrapper which can interface either to an existing operating system or directly with the underlying silicon. It currently supports nine different processors, 12 commercial operating systems and 14 ready to use boards.

It's inescapable that the current trend is towards wireless connectivity and there is a plethora of technologies available. There is considerable activity in the 'personal area networking' arena at the moment, with Bluetooth being the most mature technology in this sector. Targeting machines as opposed to people, ZigBee holds a lot of promise and we are starting to see the inevitable modules appear. But for those wishing to develop their own solution, help is still at hand in the form of embedded wireless networking specialist, Ember (**www.ember.com**).

Ember's product offering includes ZigBee ready SoCs which come with or without an embedded microcontroller. Customers may implement their preferred microcontroller and port the EmberNet stack to it.

The EmberNet protocol stack resides on the microprocessor and interfaces to the supported Ember products over the serial peripheral interface. A high level API supports the development of self healing, peer to peer networks targeted to specific customer requirements. It supports mesh, star and hybrid networks, as well as ZigBee – allowing multiple network types within the same application.

For ZigBee specific implementations, there's EmberZNet and EmberZNet 2.0. The latter offers full ZigBee device type support, including: ZigBee Coordinator; ZigBee Router; and ZigBee End Device. It also has fully integrated MAC and NWK layers, providing what Ember claims are the smallest flash and ram footprints in the industry. Support for portable end devices, such as a remote control, is also a feature – as is support for the full ZigBee APS – to allow easy application of ZigBee specified public profiles.

> "It's inescapable that the current trend is towards wireless connectivity and there is a plethora of technologies available."

Support for ZigBee defined home automation network profiles is provided, along with inherent flexibility to support other network profiles that may be defined by ZigBee in the future.

With wired and wireless connectivity almost a prerequisite in today's products, it's encouraging to see the availability of so many commercial solutions targeting low cost, deeply embedded devices. With the increased adoption of the technologies, however, we can expect to see even more emerging tomorrow.