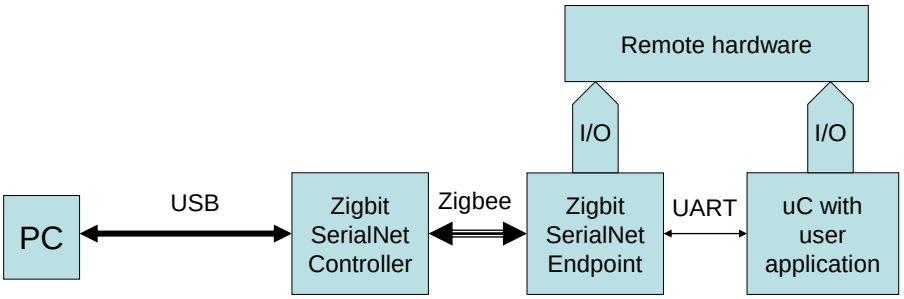
Using SerialNet

Two models of SerialNet usage are possible, see Figure 1. Both require Zigbit modules programmed with fixed SerialNet Firmware, and then configured (once only) via their serial host interface. Host-to-Host uses an additional microcontroller and therefore two host interfaces each using SerialNet commands to access the Zigbee network. Host-to-Zigbit uses a standalone module which must previously have been configured via its own host interface and a PC etc.

Host-to-Host Model



Host-to-Zigbit Model

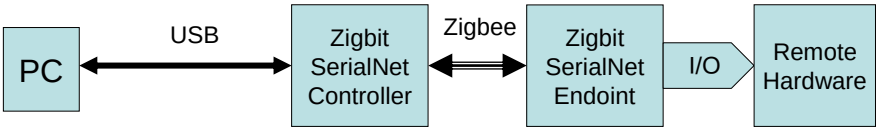


Figure 1

Zigbit IO limitations

Nine Zigbit pins (GPIO0-8) may be configured as GPIO (separately inputs or outputs) Four different pins may be configured as analog inputs: BAT (normally used to measure battery) and ADC_INPUT1-3.

The analog pins can be used as digital inputs if suitable voltage levels are provided. In fact if high input data rate is required a 3 bit D->A converter could be implemented with 4 resistors and used to input 3 digital values in parallel on one pin. Note that the 200us conversion time is much less than the time needed to transmit data back using SerialNet.

Speed

Data rate is limited by the host interface baud rate of 38.4kbit. In the Host-to-Zigbit model, changing the value of a single Zigbit output, or interrogating a single Zigbit input, takes roughly 5 bytes, or 1.2ms. Selecting a particular

remote node takes another 5 bytes so the data rate to GPIO pins is < 800 bit/s if all to the same node, or <400 bit/s if just one bit is sent or received from each node.

A/D channels are input at roughly the same speed.

This means that local processing will be much faster than controller-driven SerialNet , but this requires application code to be written under BitCloud running on the Zigbit modules, or host-to-host mode (see below).

Host-to-Host Mode

If remote node Zigbit modules have UART connected to a host microcontroller additional functionality is possible. The remote host microcontroller can use SerialNet commands to control its Zigbit module I/O, or alternatively can use its own I/O, with local processing, so that Zigbit I/O is not needed. The host controller PC and the remote microcontroller can exchange data by sending packets of data (D or DU commands) to each other. Data is received asynchronously and presented to the other host computer as a response sequence followed by the transmitted data which can be up to ~80 characters long.

The code on the remote microcontroller (uC) needs only to set up a UART (if possible choose a uP with a built-in hardware UART) and receive and transmit SerialNet text commands, performing any necessary local processing - this is straightforward. Data for the uC inputs & outputs can be sent to or from the PC in a suitably compressed form - up to 30 times faster than that possible using Zigbit I/O. In a network with many nodes even Zigbit I/O, although slow, can be locally processed before transmission to/from the controller making network throughput much higher.

The hardware cost of adding a uC to a Zigbit module is very small – a 28 pin 0.3” DIP package costing £1 with 4 interconnects to the Zigbit, and a 6 pin IDC header to allow programming. See AVR page for information on AVR hardware. The software cost is small, except for the UART serial interface, sample code for which is available. See the AVR Howto for more information on AVR tool chain & programming.