

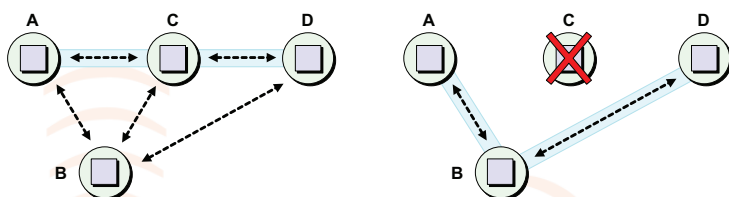


Wireless Technology to Control and Monitor Anything from Anywhere™



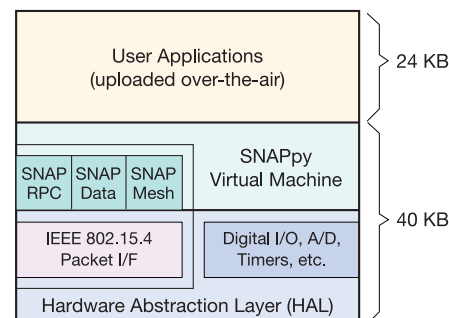
Synapse Network Operating System

SNAP® is Synapse's revolutionary wireless mesh network operating system. Representing a leap-forward in embedded intelligence, SNAP is built on a foundation of peer-to-peer networking and free-form RPC calls. Culminating a wealth of networking protocol design experience, Synapse engineers threw out the rulebooks and created a network operating system with an uncompromising eye toward performance, flexibility, and simplicity. The result is the first system in its class supporting the capability to interactively develop custom embedded applications with a modern, dynamic programming language. Develop your application wirelessly, in minutes, right before your eyes.



Self-Healing: Messages automatically re-route around failed nodes

SNAP networks interface with TCP/IP systems using existing, established Internet standards, such as XML-RPC, for remote procedure calls (RPC). SNAP's native RPC support makes it the clear choice when TCP/IP connectivity is desired.



- Self-healing mesh network.
- Instant-ON: communicate within milliseconds of power-up.
- All devices are peers - every node can bridge, route, and sleep.
- Simple, powerful scripting language (Python based).
- Customize applications and then interactively upload scripts over-the-air.
- Supports full mesh with sleeping (low power) modes.
- Remote Procedure Call (RPC) support, peer-to-peer among all nodes.
- Supports bridging to other networks (TCP/IP, ZigBee®) with transparent RPC calls.
- Scripts have full pin-level access to hardware.
- Flexible "Switchboard" for routing data between logical ports:
 - UARTs
 - Transparent Wireless Data
 - STDIO
 - Packet-mode Async Serial
- Virtually unlimited network size (millions of devices).



Wireless Technology to Control and Monitor Anything from Anywhere™



Synapse Network Operating System

With SNAP, All Devices Are Peers

Keep it simple—there's no need to stock special modules for certain roles in the network. You only have to configure the proper channel and network ID, and you have a working mesh. For connectivity to backend systems (e.g. Synapse Portal® or SNAPconnect), any device with a serial connection can function as a bridge.

Build Your Application

Odds are, your application doesn't fit precisely into a predefined "profile". With SNAP®, you can start building your application immediately—one step at a time. First verify basic connectivity. Then, interactively build the functionality you need for your application. Optimize power consumption and throughput. *SNAP puts you in control.*

Wireless Serial or Remote Serial

SNAP provides transparent wireless serial capability right out of the box. Perfect for cable-replacement applications. Concurrent general-purpose I/O is available in this mode as always. Alternatively, you can handle serial data remotely – in the SNAP device itself. Uploaded application scripts have full access to the serial ports, and the powerful SNAPpy™ language makes handling serial data a SNAP!

Low Power MicroAmp Mesh

SNAP provides the capability to synchronize sleep periods across the mesh network under control of application scripts. A sleepy-mesh reference implementation is provided for immediate use as a drop-in solution. This open, script-based implementation provides the freedom to tailor performance to fit the specific power/latency tradeoffs your application demands.

Tools Required for Assembly... NONE!

Forget hardware probes, compilers, assemblers, etc. With its companion Portal software, SNAP supports interactive application development over-the-air. This allows you to focus on your application functionality, not the wireless infrastructure.

Instant-On

A SNAP device only needs to be configured with the proper channel and network ID, and it's instantly capable of communicating with other SNAP nodes. There's no "join" phase to get on the network—this is a low-latency peer-to-peer system that is built for speed.

SNAPpy Scripting

The native message passing scheme in SNAP is remote procedure call (RPC), and the native language of these procedures is SNAPpy. SNAPpy is a subset of Python™ – a very powerful yet easy to learn general purpose language. The ability to invoke RPC functions universally in the network is key to SNAP's flexibility and performance. With RPC calls flowing effortlessly between SNAP nodes and back-end systems connected to a SNAPconnect, this is a powerful concept indeed.

Availability

SNAP Network Evaluation Kit

Available now to allow easy evaluation of SNAP—includes Portal® wireless application development environment.

SNAP In Production

SNAP is available preinstalled on Synapse, Panasonic and CEL modules for volume production use, and is also available for license to run directly on your embedded products.

SYNAPSE Wireless, Inc.

500 Discovery Drive
Huntsville, Alabama 35806
877 982-7888
synapse-wireless.com