



embeddedssoftware
solutions

embOS/IP[®]

Can be used on any CPU

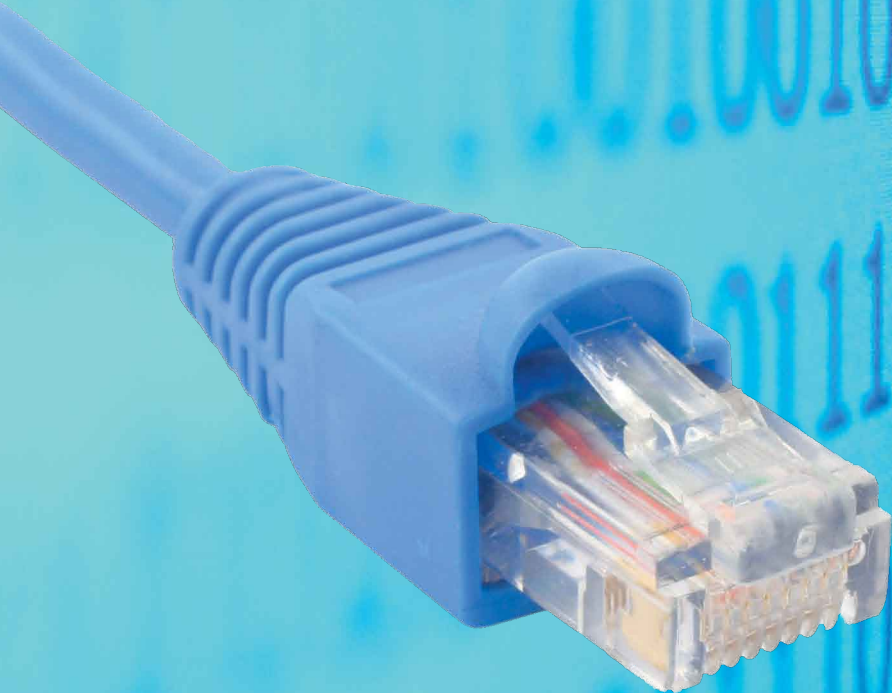
No configuration needed

Runs “out-of-the-box”

High performance

Small footprint

Zero data copy



TCP/IP stack

+++ Fast – Efficient – Reliable +++



embOS/IP is a TCP/IP stack that provides a small memory footprint, high performance solution for embedded networking solutions. The stack has been optimized for use in real-time, memory-constrained embedded systems. It offers RFC-compliant TCP/IP and a standard socket API. embOS/IP works seamlessly with the embOS operating system. Several higher-level protocols are also available.

embOS/IP structure

embOS/IP is organized in different layers, as shown in the following illustration:

Application Layer	DHCP, DNS, FTP, HTTP, Telnet
Transport Layer	TCP, UDP
Network Layer	IP, ICMP, ARP, RARP, ...
Link Layer	Ethernet (IEEE 802.3), PPP, ...

Layer Structure

Portability

embOS/IP is written in ANSI-C and apart from the Link-Layer-Driver hardware-independent.

Performance

The stack has been optimized for both, performance and code size. The standard socket interface is complemented by the Zero-copy API, which allows reading and writing of data without a single memcpy, if the target hardware and driver support DMA.

Configuration free

The entire stack can be compiled into a library. Setup is reduced to a minimum, performed at run-time. This, along with a wealth of sample programs gets you up and running quickly. Since inter-module dependencies are limited to the parts required for the functionality of the stack, unused parts of the code are automatically excluded by the linker.

Easy to use

The stack comes with a variety of confidence tests and example applications. It runs out of the box. For most microcontrollers, sample projects are available. All modules can output debug messages and warnings in debug builds. The modules to output this information can be selected at run-time, allowing the developer to focus on the aspect he is analyzing.

Multi-task support

embOS/IP allows any number of tasks to call

API functions concurrently. The stack itself uses typically 2 tasks (one for housekeeping and one for packet reception), but can also be used with just a single or even no task (polled mode). The 2 task model allows minimum interrupt latency on systems without nested interrupts.

Basic package Components

The following protocols are part of the basic package:

- ARP (Address resolution protocol)
- IP (Internet protocol)
- ICMP (Internet control message protocol)
- UDP (User datagram protocol)
- TCP (Transmission control protocol)
- Standard Socket API
- Zero-Copy API
- DNS client
- DHCP client
- Telnet server (sample)

Available Add-ons

Web server:

The embOS/IP web server allows embedded systems to present web pages with dynamically generated content. It has all features typically required by embedded systems: multiple connections, authentication, forms. The RAM usage of the web server has been kept to a minimum by smart buffer handling.

Using the sockets interface, it can be used with any TCP/IP stack.

FTP server and FTP client:

The embOS/IP FTP server can use the same file system as the web server. It can be used in r/o or in r/w mode and allows reading and modifying of configuration data or web content.

With the FTP client add-on data can be exchanged with any FTP server.

SMTP client:

The SMTP client allows you to send e-Mails

Features

- Standard socket Interface
- High performance
- Small footprint
- No configuration required
- Runs "out-of-the-box"
- Very simple network interface driver structure.
- Works seamlessly with embOS in multitasking environment
- Zero data copy for ultra fast performance
- Non-blocking versions of all functions
- Connections limited only by memory availability
- Delayed ACKs
- BSD style "Keepalive" option
- Re-assembly of fragmented packets
- Support for messages and warnings in debug build
- Drivers for most common ethernet controllers available.
- Royalty-free

from your embedded system via an e-Mail-server.

SSL:

Secure socket layer support for encrypted communication.

PPP Link layer:

As an alternative to Ethernet PPP allows the use of IP via modem or GSM.

Further add-ons are available.

Memory requirements*

The memory requirements depend on the used CPU, compiler, memory model, as well as on various other factors.

A typical ROM size for a system using ARP, IP, ICMP, UDP, TCP and sockets is about 18kB (on typical 32-bit microcontroller with size optimization). Minimum RAM usage is about 6KB for simple applications.

* Precise values depend on the functionality used. Values are measured on a specific target system and will be different on other systems.

SEGGER offers ready-to-use trial versions for common evaluation boards



Öä^&Q•ã@ŠcaZKed ID, Èää&ã•ã@B|È\Ð
 V@Pæ||-EÖ:^ac,||cOPañEÖ:^ac,||cOcaã|^ÈUc:|ã•@^UYFIAGÖPÉNS
 V^|KÉ|AGDFG|Á|||€AWCæKÉ|AGDFG|Á|GJJMæ^•Oää^&ã•ã@B|È\