

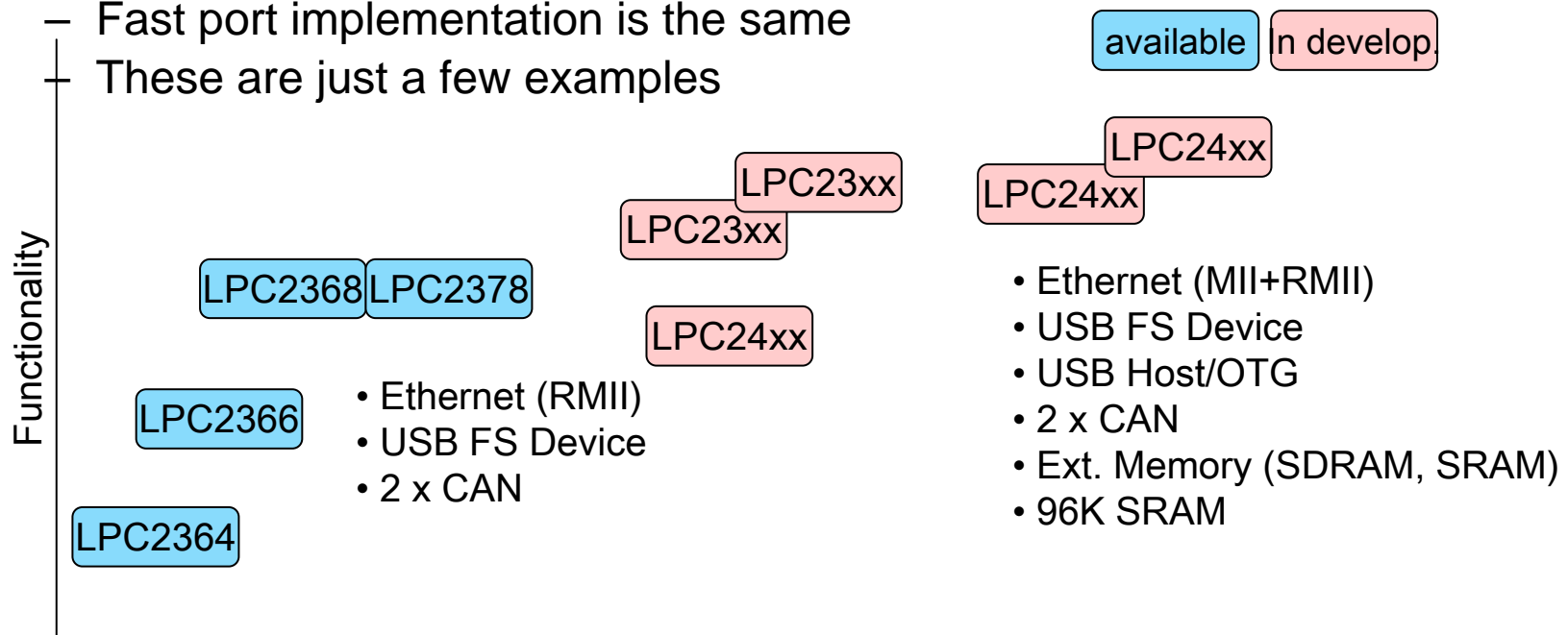


# LPC2300/LPC2400 TCP/IP Overview

TCP/IP and LPC2300/LPC2400 Family  
October 2007

# Introduction to the LPC2300/LPC2400 Family

- ▶ Family is an important word in the title
  - All blocks across the LPC2000 family are similar
  - USB of the LPC2148 is the same as the device part of the LPC2300/LPC2400
  - All SSPs are the same
  - All I2Cs are the same
  - Fast port implementation is the same
  - These are just a few examples



# Identical block memory maps

## LPC2148 SSP Memory Map

Name	Description	Access	Reset value	Address
SSPCR0	Control Register 0. Selects the serial clock rate, bus type, and data size.	R/W	0x0000	0xE006 8000
SSPCR1	Control Register 1. Selects master/slave and other modes.	R/W	0x00	0xE006 8004
SSPDR	Data Register. Writes fill the transmit FIFO, and reads empty the receive FIFO.	R/W	0x0000	0xE006 8008
SSPSR	Status Register	RO	0x03	0xE006 800C
SSPCPSR	Clock Prescale Register	R/W	0x00	0xE006 8010
SSPIMSC	Interrupt Mask Set and Clear Register	R/W	0x00	0xE006 8014
SSPRIS	Raw Interrupt Status Register	R/W	0x04	0xE006 8018
SSPMIS	Masked Interrupt Status Register	RO	0x00	0xE006 801C
SSPICR	SSPICR Interrupt Clear Register	WO	NA	0xE006 8020

## LPC23XX SSP Memory Map

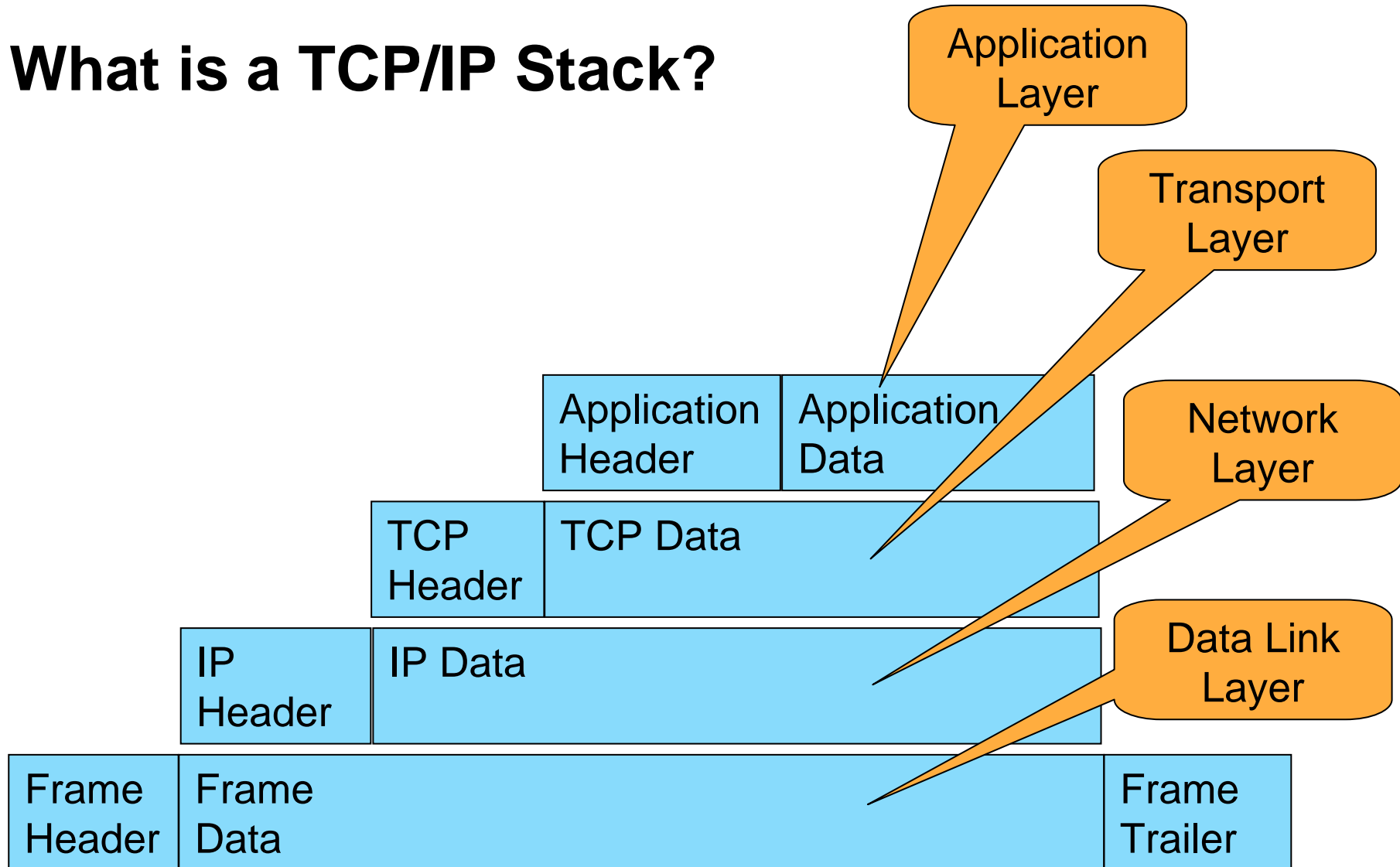
Generic Name	Description	Access	Reset Value	SSPn Register Name & Address
CR0	Control Register 0. Selects the serial clock rate, bus type, and data size.	R/W	0	SSP0CR0 - 0xE006 8000 SSP1CR0 - 0xE003 0000
CR1	Control Register 1. Selects master/slave and other modes.	R/W	0	SSP0CR1 - 0xE006 8004 SSP1CR1 - 0xE003 0004
DR	Data Register. Writes fill the transmit FIFO, and reads empty the receive FIFO.	R/W	0	SSP0DR - 0xE006 8008 SSP1DR - 0xE003 0008
SR	Status Register	RO		SSP0SR - 0xE006 800C SSP1SR - 0xE003 000C
CPSR	Clock Prescale Register	R/W	0	SSP0CPSR - 0xE006 8010 SSP1CPSR - 0xE003 0010
IMSC	Interrupt Mask Set and Clear Register	R/W	0	SSP0IMSC - 0xE006 8014 SSP1IMSC - 0xE003 0014
RIS	Raw Interrupt Status Register	R/W		SSP0RIS - 0xE006 8018 SSP1RIS - 0xE003 0018
MIS	Masked Interrupt Status Register	R/W	0	SSP0MIS - 0xE006 801C SSP1MIS - 0xE003 001C
ICR	SSPICR Interrupt Clear Register	R/W	NA	SSP0ICR - 0xE006 8020 SSP1ICR - 0xE003 0020
DMACR	DMA Control Register	R/W	0	SSP0DMACR - 0xE006 8024 SSP1DMACR - 0xE003 0024

Same functions, same address - LPC23XX/LPC24XX adds DMA

**LPC2300/LPC2400**

**Ethernet**

# What is a TCP/IP Stack?

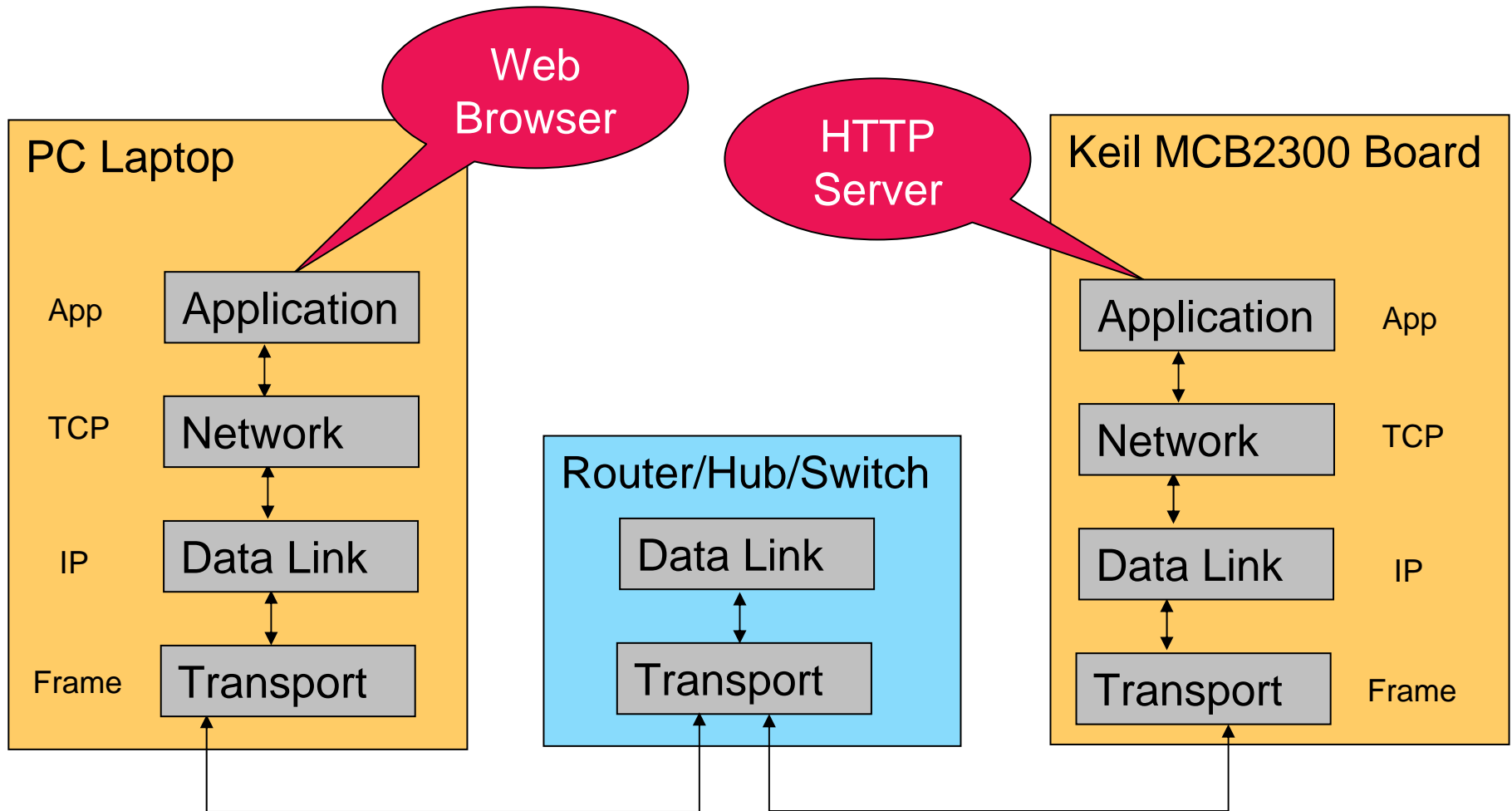


# Basic Ethernet Frame

- ▶ All numbers are in Bytes
- ▶ PRE (preamble) - alternating 1's/0's
- ▶ SOF (Start of Frame) - alternating 1's/0's but last two bits are "11"
- ▶ DA (Destination Address) – who gets it
- ▶ SA (Source Address) – who sent it
- ▶ [NOT SHOWN: optional VLAN header, 4 bytes]
- ▶ Length/Type – interpreted as length when  $\leq 1518$ d or as type when  $> 1536$ d
- ▶ Data payload – minimum of 46, maximum of 1500 bytes
- ▶ FCS (Frame Check Sequence) – is the CRC of DA, SA, VLAN (if present), Length/Type, and Data payload

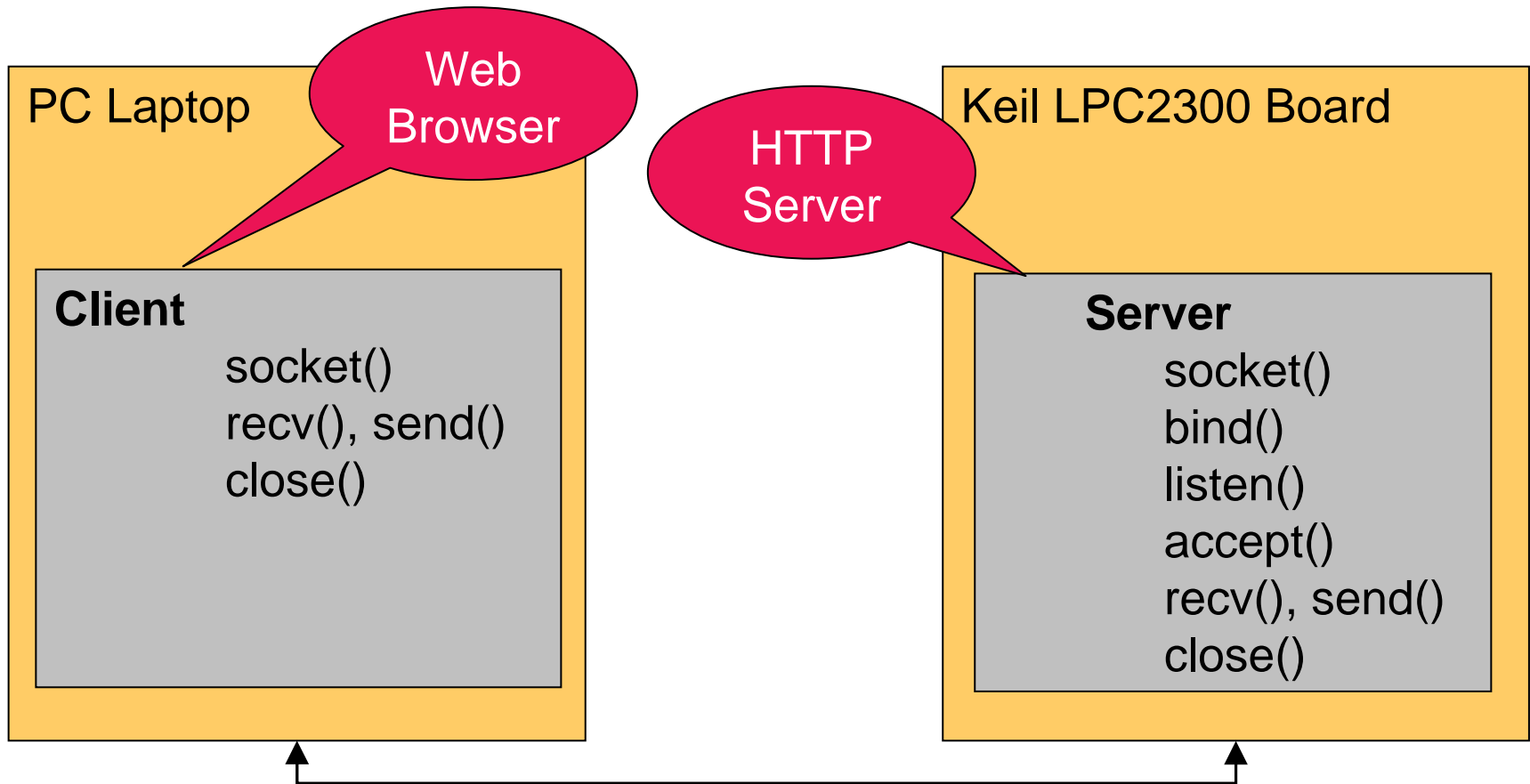
PRE	SOF	DA	SA	Length / Type	Data payload	FCS
7	1	6	6	2	46 - 1500	4

# What is the TCP/IP Stack Demo?



# TCP/IP API

Embedded TCP/IP API will typically be a partial or complete version of the Berkeley Sockets API



# Source Code from NXP Semiconductors

TCP/IP Stack	NXP Sample Code	NicheLite for LPC
Application		<ul style="list-style-type: none"><li>•TFTP client and server</li><li>•DHCP client</li><li>•DNS client</li><li>•Simple Web Server</li><li>•Virtual File System</li><li>•Command-line</li></ul>
----- Transport	-----	----- TCP, mini-sockets API
----- Network	-----	----- IP, ICMP, ARP, UDP
----- Data Link (MAC)	Ethernet Frame	Ethernet Frame

# Software from Interniche



## Nichelite

- Address Resolution Protocol (ARP)
- Internet Protocol (IP)
- Internet Control Message Protocol (ICMP)
- User Datagram Protocol (UDP)
- Transmission Control Protocol (TCP)
- Dynamic Host Configuration Protocol (DHCP) Client
- Bootstrap Protocol (BOOTP)
- Trivial File Transfer Protocol (TFTP)
- Domain Name Service (DNS) client
- Simple Web Server
- Virtual File System
- Command-line

## NicheStack IPv4

### **Feature rich TCP/IP adds:**

- Support for NAT Routing
- Fragmentation and re-assembly
- IP Routing:
  - Routes set via ICMP, SNMP, IGP, etc.
- Loop back test driver
- Optimized checksum routines
- Multi Homed IP support
- Support for Auto-IP Assignment

### **Add**

- BSD Sockets
- Telnet server
- FTP server

## Options available for Nichelite & NicheStack IPv4

### Device Management

- Utilities (FTP, Telnet, FileSystem)
- E-mail (SMTP, POP3)
- Webserver (HTTP server, SSL)
- Simple Network Management Protocol (SNMP)

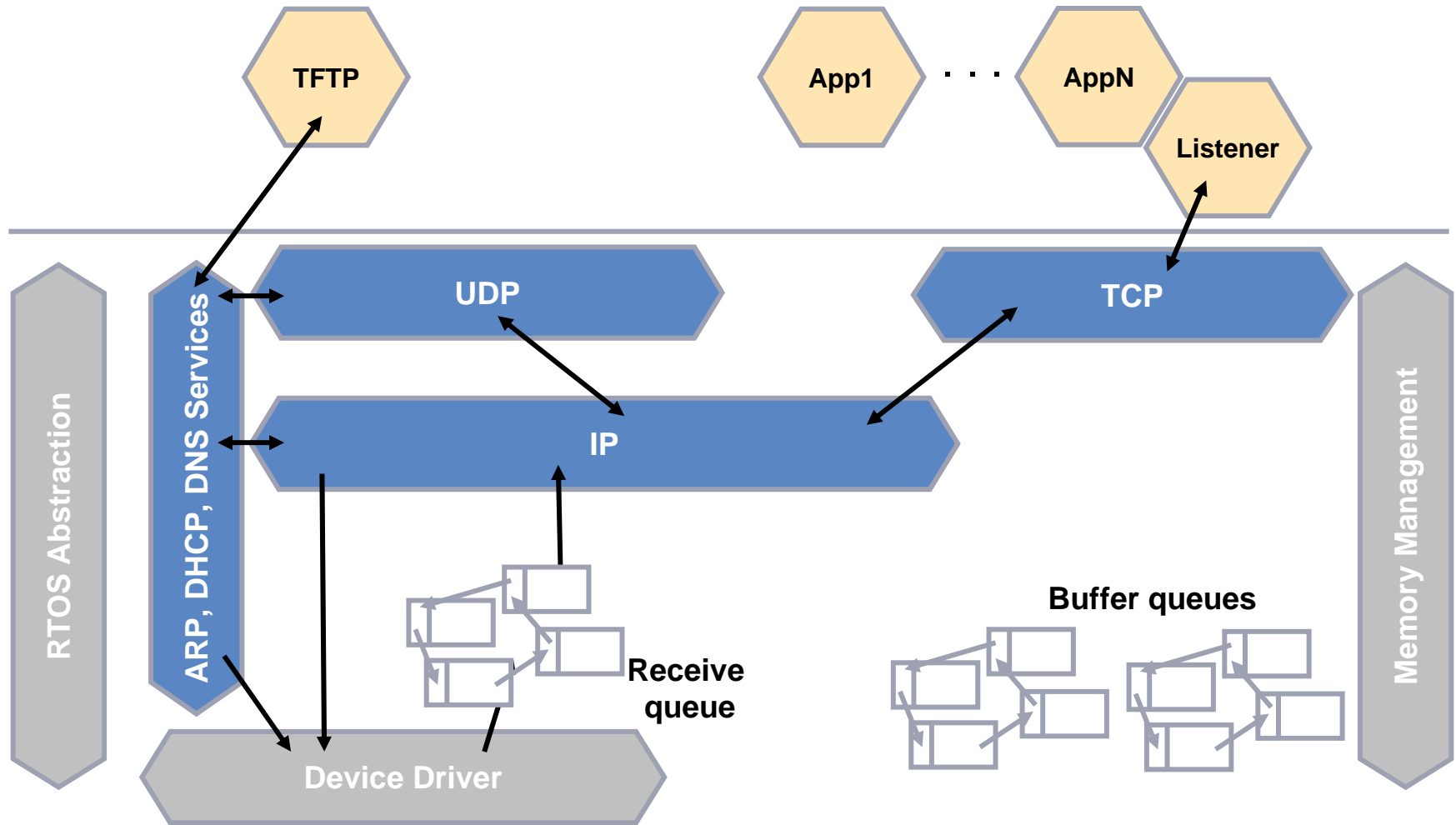
### Gateway Protocols

- Domain Name Server (DNS)
- Network Address Translation Router (NAT)
- Routing Information Protocol (RIP)
- Dynamic Host Configuration Protocol (DHCP)

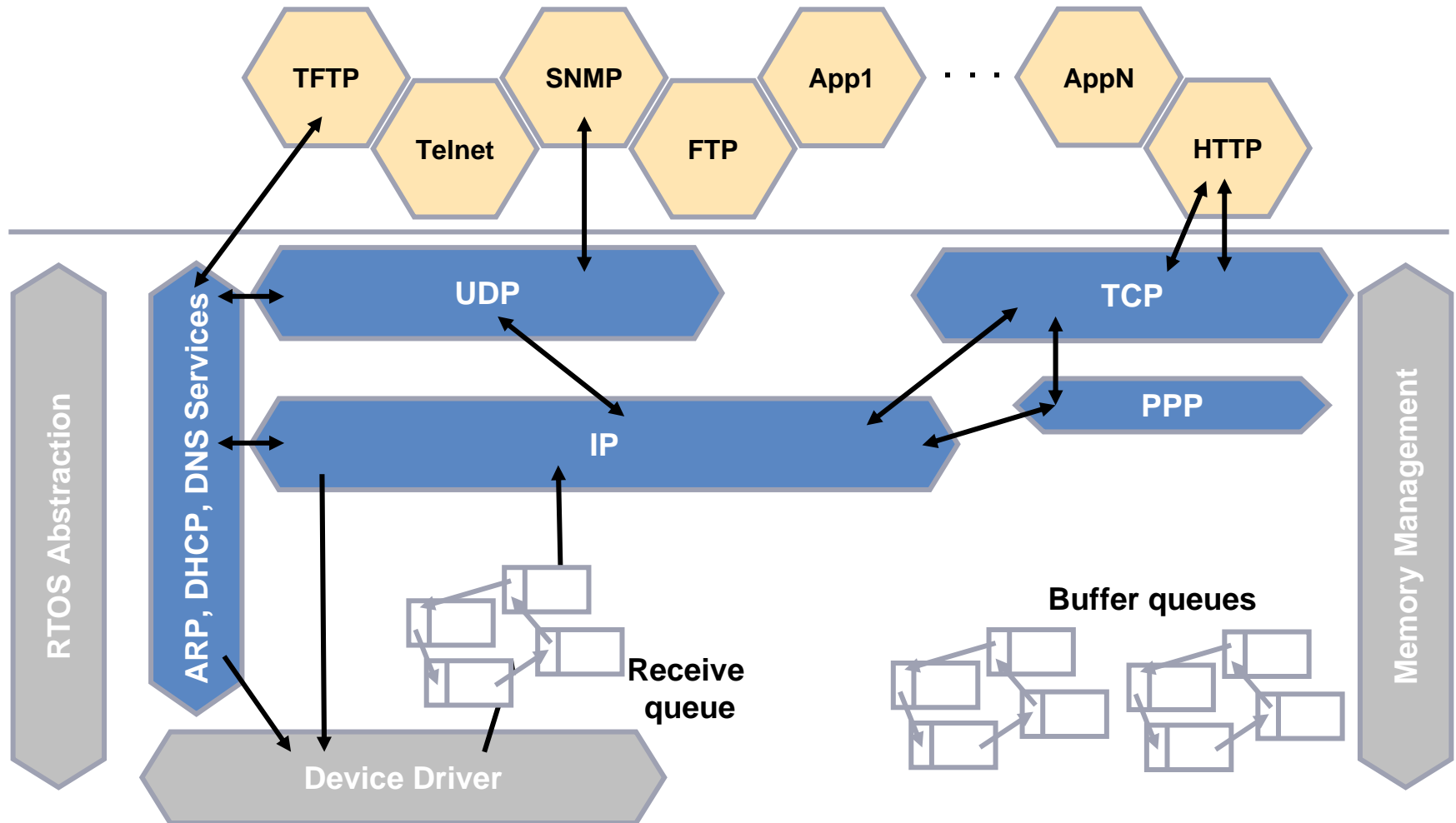
### Security

- Internet Key Exchange (IKE)
- IP Security (IPSec) ,
- Secure Socket Layer (SSL)
- Cryptography (DES, AES, 3DES)

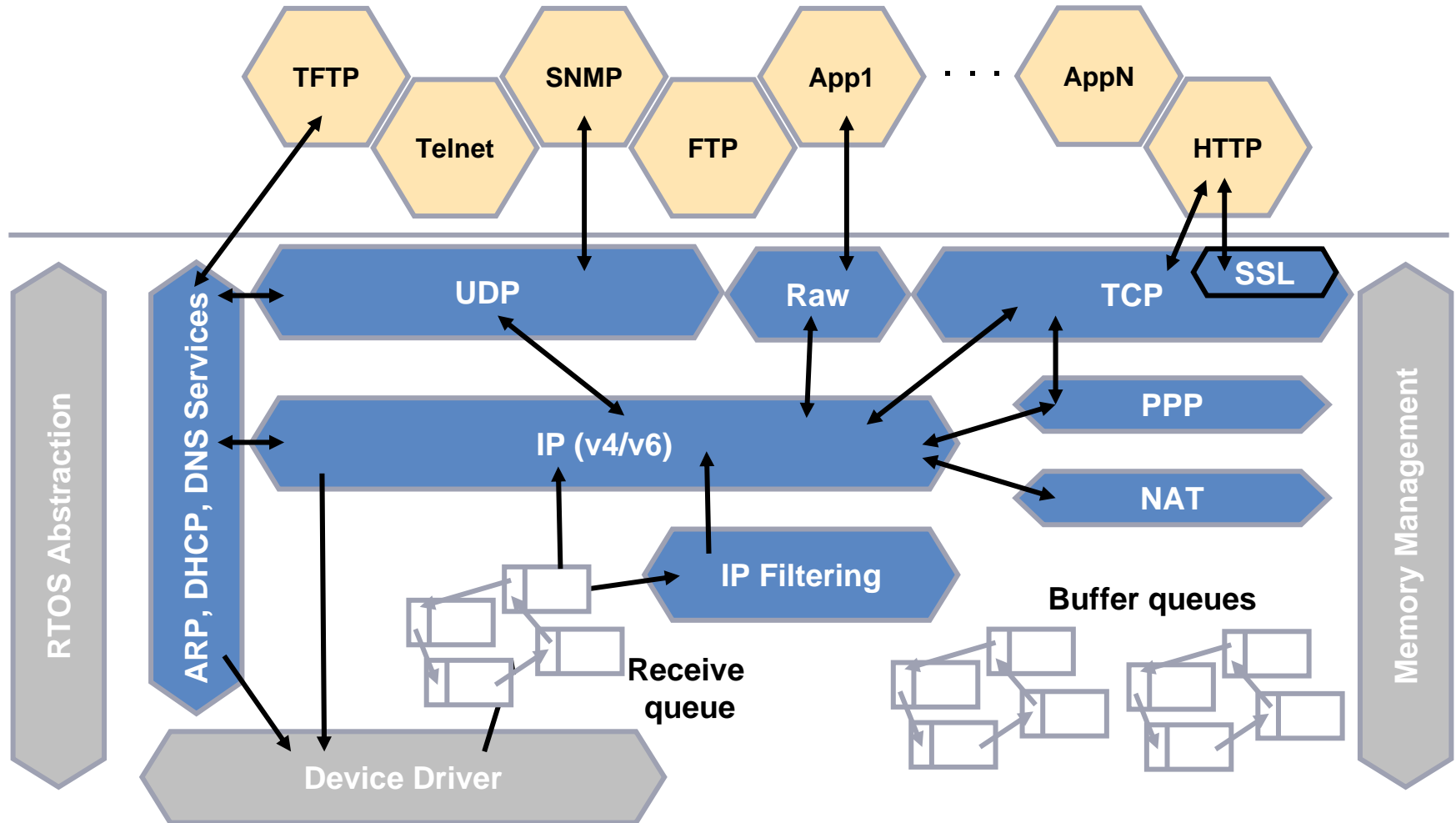
# NicheLite for LPC Architectural Overview



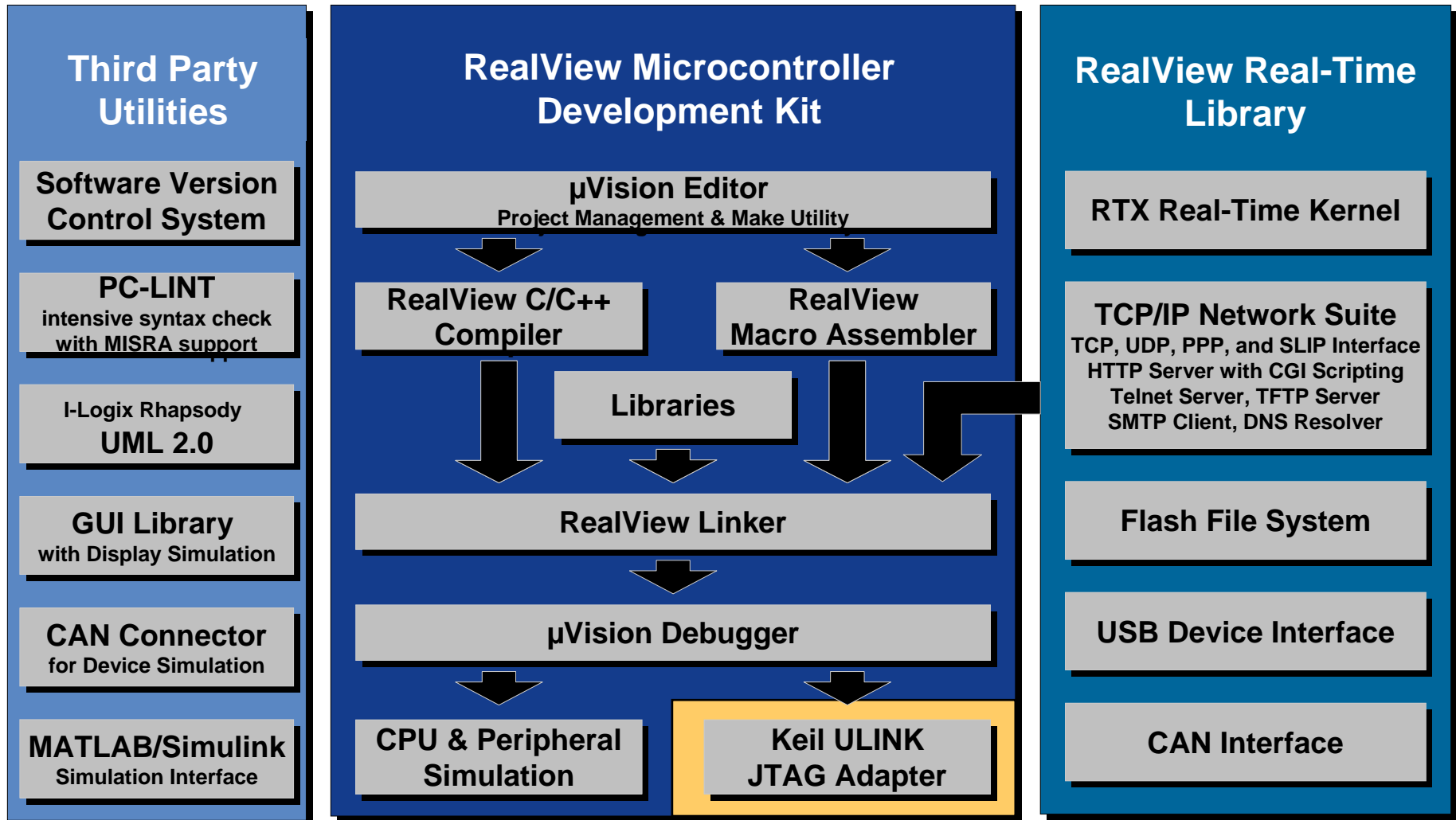
# NicheLite w/Add-ons Architectural Overview



# NicheStack Architectural Overview



# RealView Microcontroller Development Kit



# RealView Real-Time Library

Extensive library of common ready-to-use components, speed software development.

## RealView Real-Time Library

RTX Real-Time Kernel

### TCP/IP Network Suite

TCP, UDP, PPP, and SLIP Interface  
HTTP Server with CGI Scripting  
Telnet Server, TFTP Server  
SMTP Client, DNS Resolver

Flash File System

USB Device Interface

CAN Interface

Examples

## Meets Embedded Developers needs

- Solves Common Embedded Challenges
  - Real-Time Systems
  - Embedded Communication
- Designed for use with MCU Devices
- High level of Re-use

## Extensive Range of Application Examples

- As with all Keil tools.
- Can be used as building blocks.

## Royalty Free

- RTX (Object) included in RealView MDK.

# Software from Keil

---

---

The TCP/IP Network Suite is a TCP/IP stack implemented specifically for embedded applications.

## Application Layer

- DNS
- TFTP & Telnet
- SMTP Client
- Embedded Web server with CGI and password protection.

## Transport Layer

- TCP and UDP sockets

## Data Link Layer

- Ethernet, PPP and SLIP interfaces





**Keil™ includes the TCP/IP stack in their RT-Library**

# Real-Time Library Examples







 C:\Keil\ARM\ARTX\Examples

Example Projects provide a basic configuration and help you to get running quickly.  
All examples run on Evaluation Boards.

## RTOS Kernel Examples

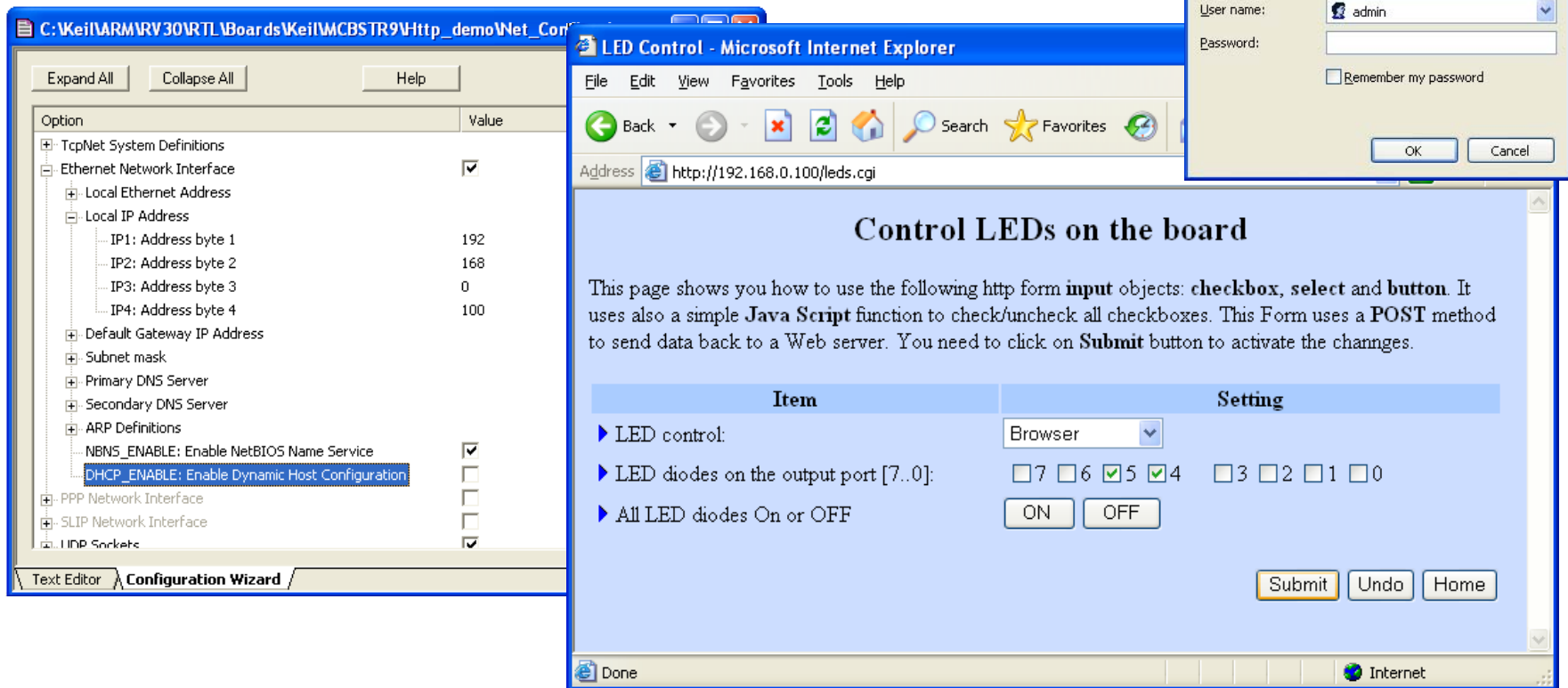
-  Artx\_ex1 Use basic RTOS kernel features: timeouts & signals.
-  Artx\_ex2 Show task priorities and signal passing.
-  Mailbox Using the Mailbox and Memory Allocation functions.
-  Traffic Complete Traffic Light Controller with serial communication.

## TCP/IP Networking Examples

-  easyWeb Simple Web interface
-  Http\_demo HTTP Server with Password Protection and CGI Scripting.
-  Telnet\_demo Telnet Server shows a simple IP based command line interface.
-  DNS\_demo Using the DNS resolver that connects to host names.
-  LEDSwitch Controlling with TCP/IP, UCP via Ethernet, SLIP or PPP Link.
-  SMTP\_demo Shows sending of a dynamic message to an email address.

# Real-Time Library Examples

- ▶ HTTP Server with CGI Interface
  - Server provides authentication and allows multiple sessions
  - A CGI interface allows interaction with MCU hardware.



The image shows a configuration wizard window on the left and a web browser window on the right. The configuration wizard is titled 'Configuration Wizard' and shows a tree view of network options. The web browser window is titled 'LED Control - Microsoft Internet Explorer' and shows a CGI interface for controlling LEDs on a board. A login dialog box is also visible in the top right corner.

**Configuration Wizard Options:**

Option	Value
TcpNet System Definitions	
Ethernet Network Interface	<input checked="" type="checkbox"/>
Local Ethernet Address	
Local IP Address	
IP1: Address byte 1	192
IP2: Address byte 2	168
IP3: Address byte 3	0
IP4: Address byte 4	100
Default Gateway IP Address	
Subnet mask	
Primary DNS Server	
Secondary DNS Server	
ARP Definitions	
NBNS_ENABLE: Enable NetBIOS Name Service	<input checked="" type="checkbox"/>
DHCP_ENABLE: Enable Dynamic Host Configuration	<input type="checkbox"/>
PPP Network Interface	<input type="checkbox"/>
SLIP Network Interface	<input type="checkbox"/>
UDP Sockets	<input checked="" type="checkbox"/>

**LED Control - Microsoft Internet Explorer**

Address: <http://192.168.0.100/leds.cgi>

### Control LEDs on the board

This page shows you how to use the following http form **input** objects: **checkbox**, **select** and **button**. It uses also a simple **Java Script** function to check/uncheck all checkboxes. This Form uses a **POST** method to send data back to a Web server. You need to click on **Submit** button to activate the changes.

Item	Setting
▶ LED control:	<input type="text" value="Browser"/>
▶ LED diodes on the output port [7..0]:	<input type="checkbox"/> 7 <input type="checkbox"/> 6 <input checked="" type="checkbox"/> 5 <input checked="" type="checkbox"/> 4 <input type="checkbox"/> 3 <input type="checkbox"/> 2 <input type="checkbox"/> 1 <input type="checkbox"/> 0
▶ All LED diodes On or OFF	<input type="button" value="ON"/> <input type="button" value="OFF"/>

$\mu$ C/TCP-IP is a compact, reliable, high performance TCP/IP protocol stack from the makers of uCOS.

- Built from the ground up with Micrium's renowned quality
- Highly scalable and reliable,
- Rapid configuration
- Full Berkeley 4.4 socket layer interface
  - TFTP<sub>s</sub> , FTP<sub>c</sub>, FTP<sub>s</sub>
  - HTTP<sub>s</sub>, SNT<sub>Pc</sub>
  - POP3<sub>c</sub>, SMT<sub>Pc</sub>
  - DHCP<sub>c</sub>, DNS<sub>c</sub>
  - TELNET<sub>cs</sub> Q4-2007
  - PPP<sub>c</sub> Q4-2007

# EasyWeb

## Tiny TCP/IP Stack and Web Server

This small TCP/IP stack and web server was published in the extra issue 'Embedded Internet' of the magazine Design & Elektronik.

For a complete German description of this software the extra issue can be ordered at [www.elektroniknet.de/extraheft](http://www.elektroniknet.de/extraheft).

- ▶ Original port to MCB2300 for ARM Compiler (Keil) available as part of Keil ARM software distribution.
- ▶ Port to GNU available on yahoo groups site:
  - <http://tech.groups.yahoo.com/group/lpc2000/>

# EasyWeb

## Tiny TCP/IP Stack and Web Server

This TCP/IP stack handles the ARP, ICMP, IP and TCP protocols. It is optimized for low resource consumption, not for performance. Because of the low resource consumption, the TCP/IP stack has some restrictions:

- ▶ Only one active TCP session at the same time
- ▶ No reassembling of fragmented incoming IP frames
- ▶ No buffering of TCP segments that are received in wrong order
- ▶ No checksum verification of received data
- ▶ no support for 'type of service (TOS)' and security options
- ▶ received TCP options are ignored

# uIP Stack

## GPL Licensed Code

Mr. Adam Dunkels created the uIP stack and makes it available under a BSD-Style License.

- ARP-protocol implemented in the higher layer as provided in the original uIP software.
- Other high layer protocol functions can be added to this basic software.
- The main routine uses polling making it easy to call the routines by a real-time OS.

port available here:

- <http://www.freertos.org/portlpc2368uIP.html> (rowley)

# uIP Stack

## GPL Licensed Code

### Supports

- ▶ Speed support 10 or 100 MB/s
- ▶ Half-, full duplex or loopback mode
- ▶ Receive fragmentation of frames
- ▶ Supports MII or RMII interface
- ▶ Flexible receive and transmit buffer configuration to adapt to frame formats required by application
- ▶ Different receive filter capabilities to reduce the amount of packet processing
- ▶ Uses DMA engine in the Ethernet controller
- ▶ Interrupt driven

### Currently not supported:

- ▶ Imperfect hash filtering
- ▶ Transmit flow control
- ▶ Transmit fragmentation of frames
- ▶ Wake-up on LAN (WoL)

# Recommended Reading

- ▶ Ethernet: The Definitive Guide (O'Reilly)
  - Chapter 1 (Evolution of Ethernet)
  - Chapter 2 (Ethernet System operation)
  - Chapter 3 (MAC Protocol, half-duplex)
  - Chapter 4 (full-duplex)
  - Chapter 5 (Auto-negotiation)
  - Chapter 6 (Media, MII signal definition)
  - Chapter 7 (10BASE-T)
  - Chapter 9 (100BASE-TX)
- ▶ RMII Specification
  - [www.national.com/appinfo/networks/files/rmii\\_1\\_2.pdf](http://www.national.com/appinfo/networks/files/rmii_1_2.pdf)

