



# HTTP Demonstration

**MCB2300**

RealView Real-Time Library and  
RealView MDK



## Overview

### Aim

To demonstrate easy networking implementation using RealView Real-Time Library and RealView MDK, highlighting;

- Networking Components of RealView Real-Time Library.
- Integration of Real-Time Library with RealView MDK.
- Configuration wizards for Networking and RTX real-time kernel
- Control of hardware across TCP/IP network.

### Requirements

- RealView MDK v3.01 or higher
- HTTP example file (MBC2300 version)
  - Open project...Keil\ARM\Boards\Keil\MCB2300\RL\TCPnet\Http\_demo
  - Downloaded from <http://www.keil.com/download/docs/328.asp>
  - Atmel and STMicroelectronics versions are also available from [www.keil.com](http://www.keil.com)
- MCB2300 Evaluation board.
- Ethernet cross-over cable.
- Two USB cables – one for ULINK2 (or ULINK) and one for board power.
- ULINK2 or ULINK

### Conventions



Box shows actions for demonstrator or user of demo.  
Icon represents the button controls to be used.

### Information.

Further information on RealView Real-Time Library and TCP/IP is available in:

#### **Application Note 192**

Using TCP/IP Examples on ARM-powered Evaluation Boards

[http://www.keil.com/appnotes/docs/apnt\\_192.asp](http://www.keil.com/appnotes/docs/apnt_192.asp)

## RealView Real-Time Library

RTL-ARM offers a library of commonly used components to aid developers in developing their applications. RTL-ARM includes:

- The **RTX Real-Time Kernel**, a full-featured Real-Time Operating System that provides functions for: time management (for timeout or delay), semaphore management (for resource sharing), event management (for task synchronization), and mailbox management (for task inter-communication). The Kernel is easily retargeted using a single configuration file.
- The **TCP/IP Networking Suite (TcpNet)**; a ground-up TCP/IP stack implemented specifically for embedded applications. It includes TCP and UDP sockets, PPP and SLIP interfaces, DNS, Telnet, TFTP, SMTP, and an Embedded Webserver with CGI and password protection. The stack works with Ethernet or Serial (Modem) interfaces. Preconfigured examples are provided for several standard evaluation boards.
- The **Flash File System** which allows you to save files in Flash or ROM (RAM support is currently in development). These are useful in systems that require large amounts of data storage or for systems with internet interfaces (HTTP or FTP).
- The **USB Device Interface** which implements standard USB devices such as HID or Mass Storage.
- The **CAN Interface** which provides a classic CAN driver that interfaces to the RTX Real-Time Kernel.

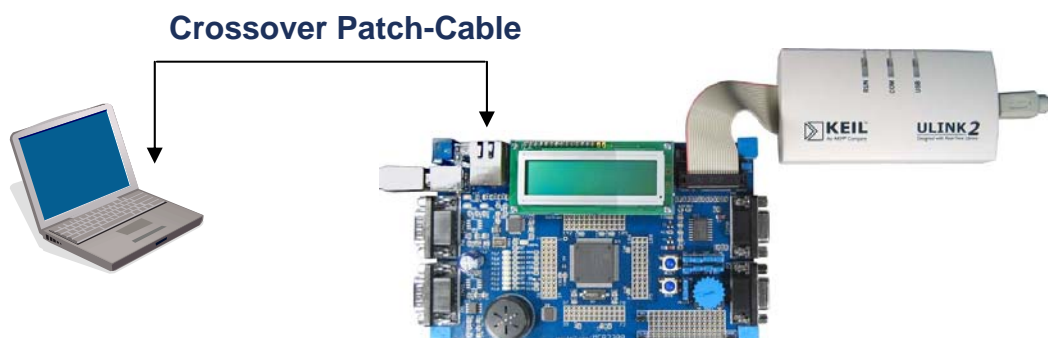
While it is possible to implement an embedded program without using a real-time kernel, a proven kernel like Keil RTX enables developers to save time, produce a reliable, expandable system and makes software development easier.

More information regarding RealView Real-time Library is available at <http://www.keil.com/rl-arm>

The HTTP example uses RTX and TcpNet components.

### Set-up – Direct PC Connection

You may use a cross-over patch cable to connect a single PC to the MCB2300. In this case the network card of the PC needs to be configured for fixed IP addresses (see appendix on page 11).

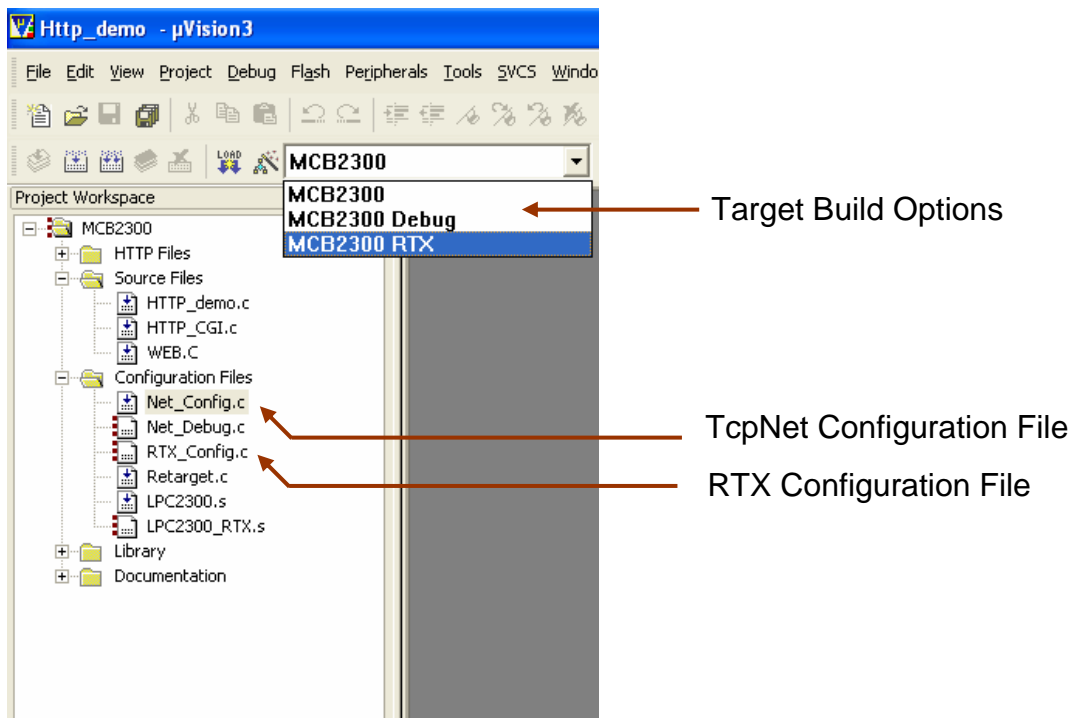


## Configure Program

### Open HTTP Project



When the HTTP project is open you will see the following files and options in  $\mu$ Vision.



Full support of RTL-ARM is available through  $\mu$ Vision in RealView MDK, making it easy to configure various parameters of RealView Real-Time Library, we will show configuration for:

- **TcpNet** - Networking Suite
- **RTX Real** -Time Kernel

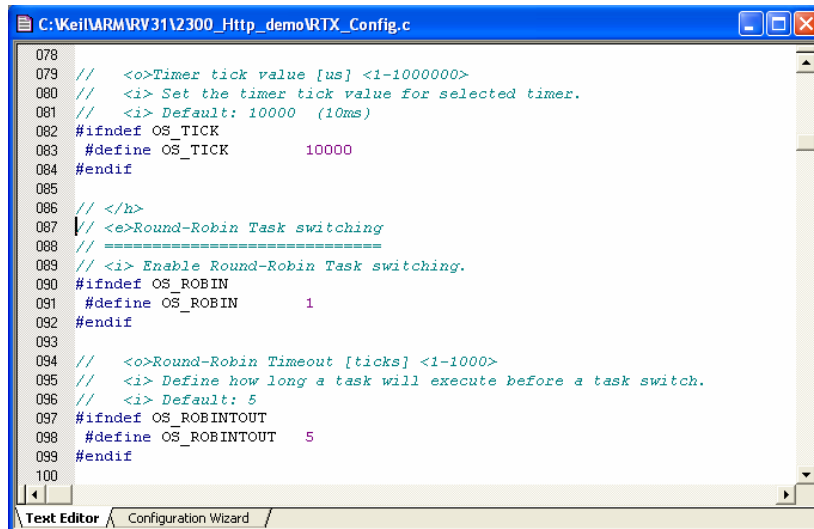
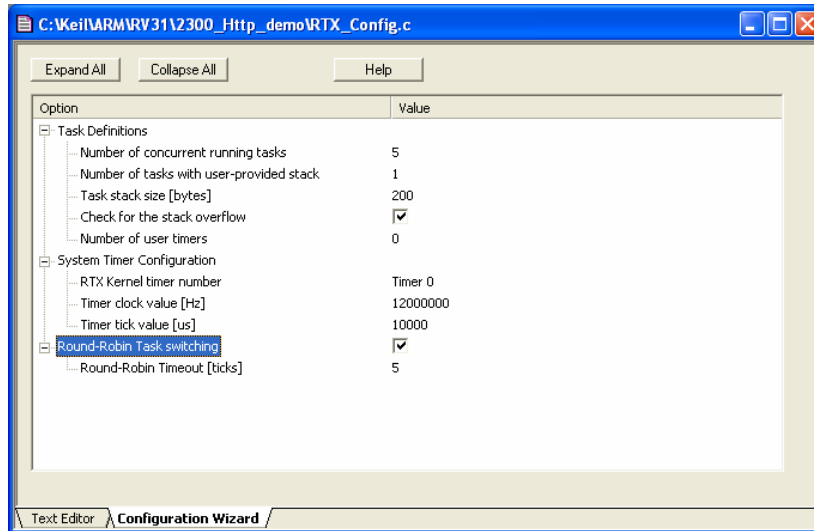


## RTX Configuration

The same style configuration wizard is available for RTX, enabling easy setup of the Real-Time Kernel.



Double Click RTX\_Config  
Click 'Configuration Wizard'



# Build and Download Program

## IMPORTANT:

**DO NOT RECOMPILE THE EXAMPLE WITHOUT RL-ARM LICENSE!**



Compile and link the project with **Project – Rebuild all target files.**



Connect MCB2300 using ULINK  
Download application using **Flash - Download**

The program will download to the on-chip Flash within a few seconds;  $\mu$ Vision will confirm that it has loaded correctly.

Reset MCB2300 using Reset button

With the application running, the LED's on the board will start blinking, when connected to a LAN.

## Test LAN Connection

The LAN connection to the board can be checked using a simple PING command.

Open a **Command Prompt** window and type:

```
PING 192.168.0.100
```

An output similar to the one below should be displayed:

```
Command Prompt
Pinging 192.168.0.100 with 32 bytes of data:
Reply from 192.168.0.100: bytes=32 time=1ms TTL=128
Reply from 192.168.0.100: bytes=32 time<1ms TTL=128
Reply from 192.168.0.100: bytes=32 time=1ms TTL=128
Reply from 192.168.0.100: bytes=32 time=1ms TTL=128
Ping statistics for 192.168.0.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

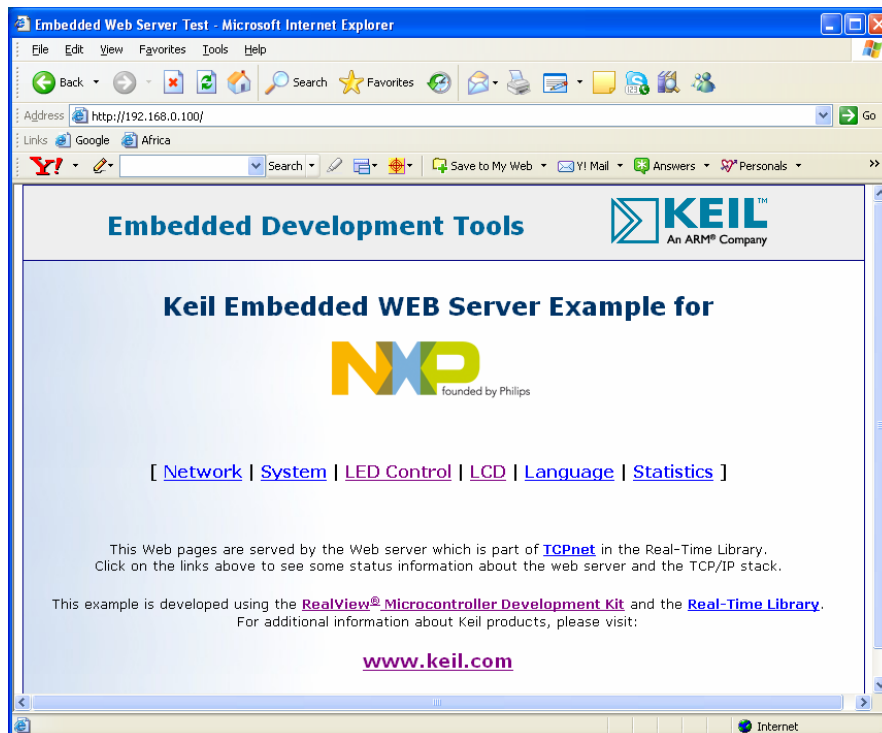
## Control of board using HTTP server

Open an **Internet Browser** and type: <http://192.168.0.100>

The HTTP server running on the MCB2300 has password protection, you will see:



A control window will now open allowing control or viewing of information from the board, this is the Home page for the HTTP server.



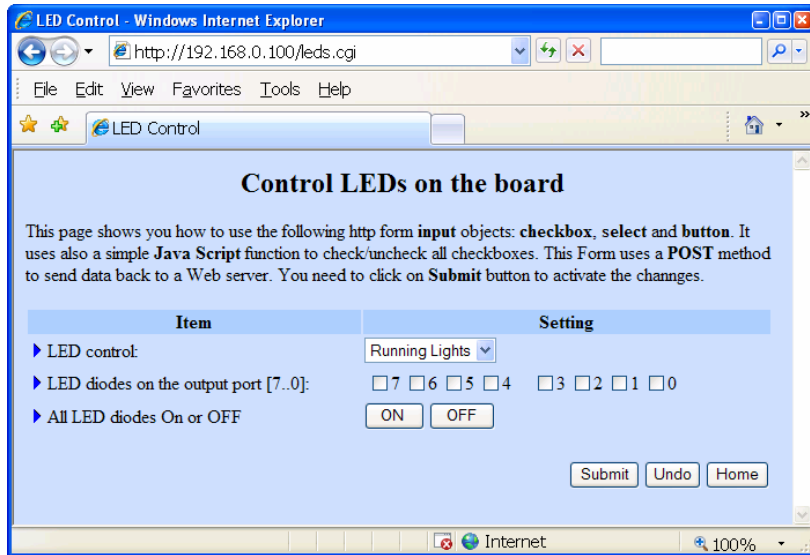
We will demonstrate control of the board functions via TCP/IP for:

- Control of LED's
- Control of LCD Characters
- Password Setting/Reset.

## LED Control

This window controls the LED's on the board.

**!** Ensure LED Control **Browser** is selected to enable control from browser window.

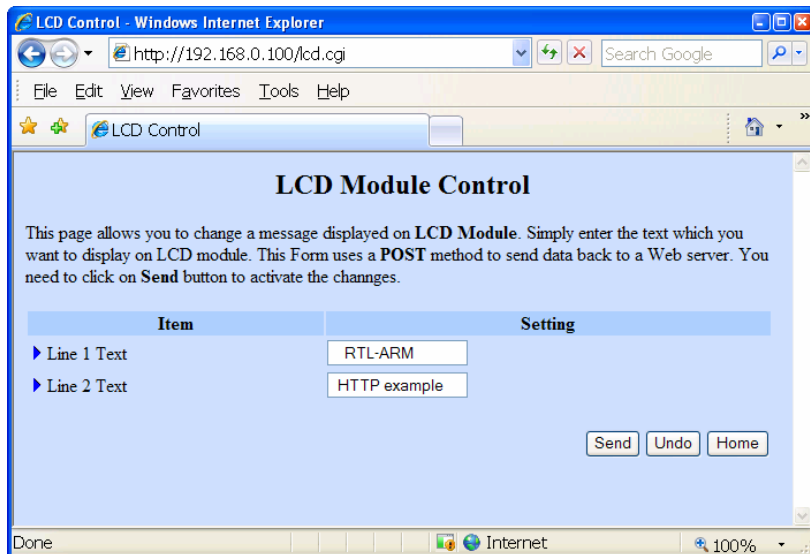


**Home** takes you back to the HTTP Server home page.

## LCD Control

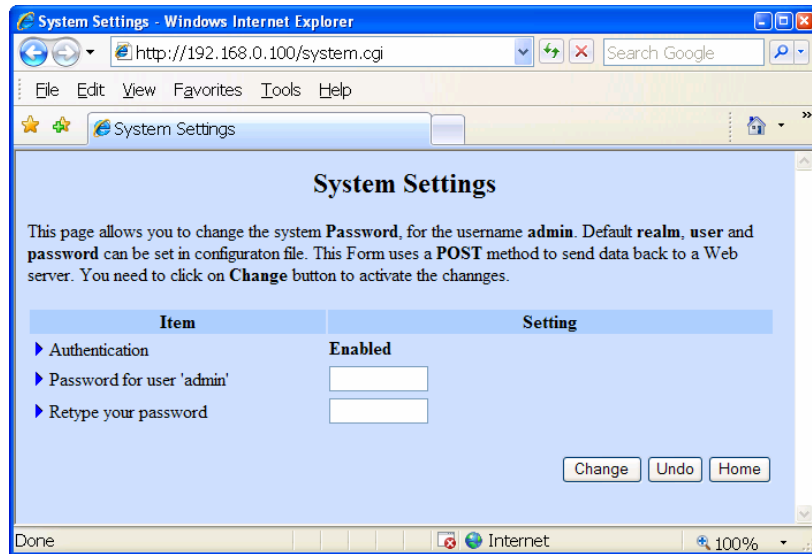
Allows you to enter text to display on the board LCD.

Go to HTTP Server Home page and select **LCD**  
Enter new words in Text fields - **Send**



## Set Password

System page allows user to set password.



**Demonstration End**

## Appendix - IP Address Setup

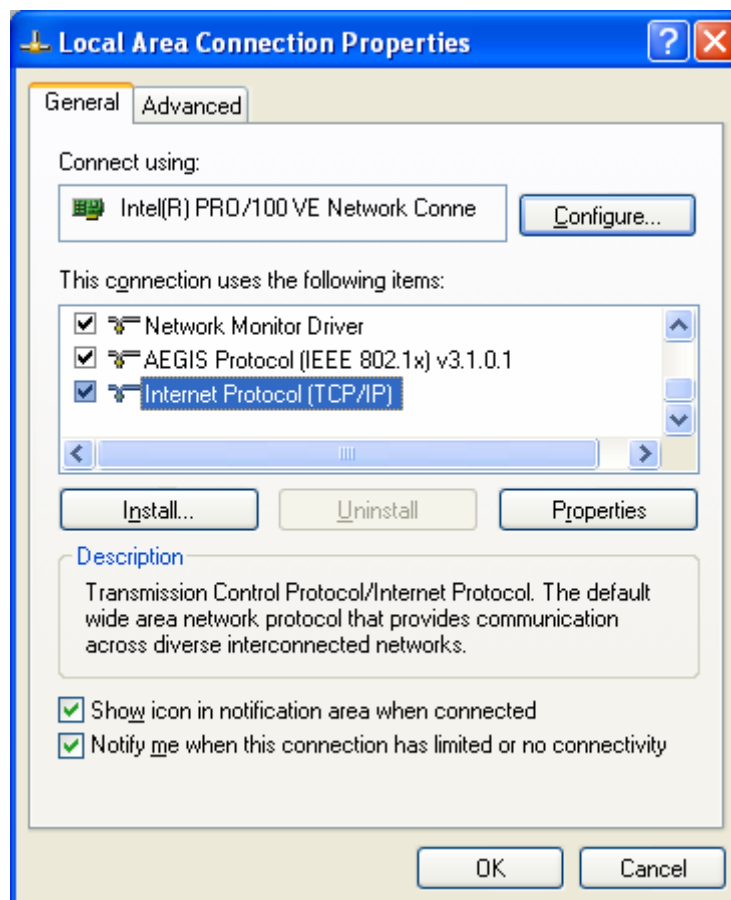
The setup of a TCP/IP connection requires several IP addresses:

- **IP Address** specifies a local static four byte IP address for the PC. This address must be unique for each device that connects to a LAN.
- **Subnet Mask** specifies the Net Mask. Typical (for most LANs) the class C mask **255.255.255.0** is used. The subnet mask is used to determine if an IP request is within the LAN or must be sent to the Default Gateway.
- **Default Gateway** specifies the IP address of the default gateway that is used when the external WAN (Internet) is accessed. If your TcpNet application will be used only in the LAN, the Default Gateway address is not required.

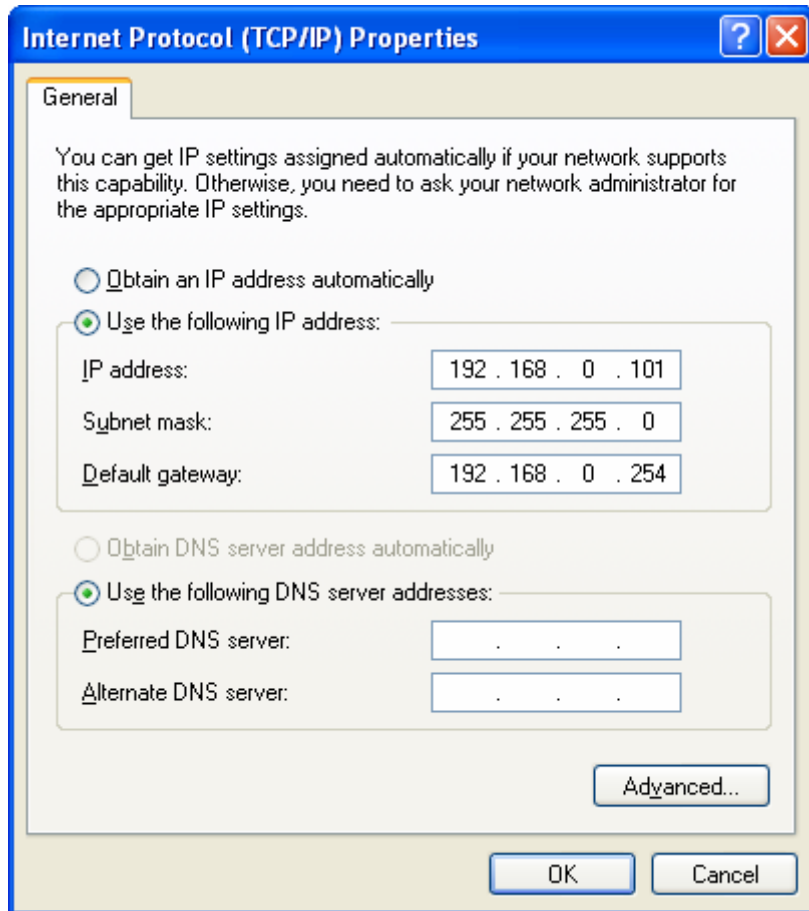
Typically the network administrator provides fixed IP addresses for each device on the LAN. A certain address range of the LAN may be assigned by a DHCP (**D**ynamic **H**ost **C**onfiguration **P**rotocol) server. Using DHCP you may obtain IP addresses automatically from the DHCP server.

### IP Address Setup on Windows 2000/XP.

The network parameters of a PC are configured under **Control Panel - Network Connections - Local Area Connection**.



To configure the TCP/IP parameters select the item **Internet Protocol (TCP/IP)** and click on **Properties**. In case that this item is not listed click on **Install** to add this protocol.



The dialog allows you to configure your computer for a fixed IP address.

- **IP Address** is the IP address for the PC and must be unique for each device in the LAN. For running the TCP/IP application on a simple test LAN you may use **192.168.0.1** for your PC.
- **Subnet Mask** specifies the Net Mask. Typical (for a small LAN) the class C mask **255.255.255.0** is used.
- **Default Gateway** specifies the IP address of the default gateway that is used when the external WAN (Internet) is accessed. If your TcpNet application will be used on local LAN only, you do not have to specify the Default Gateway.

The settings **Obtain an IP address automatically** and **Obtain DNS server address automatically** obtains the IP addresses from a DHCP server.