

Ingeniería de Sistemas Electrónicos

Protocolos de comunicaciones: Ethernet



Dpto. de Ingeniería Telemática y
Electrónica

Conocimientos

□ SBM.

- Buses
- CMSIS
- RTOS
- Placa de desarrollo
- Programación en C

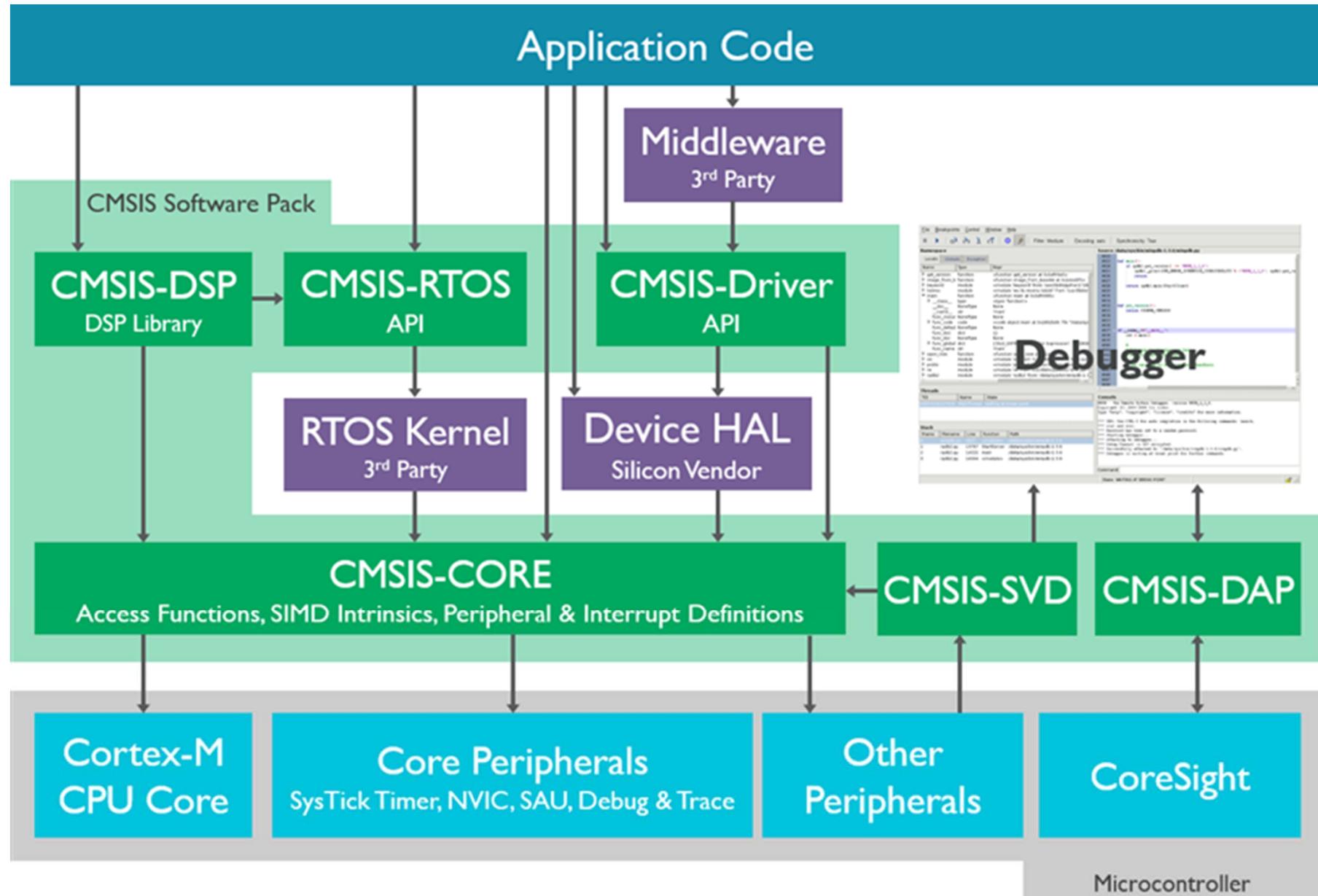
□ EA2 y IE

- Circuitos de medida
- Acondicionar señales

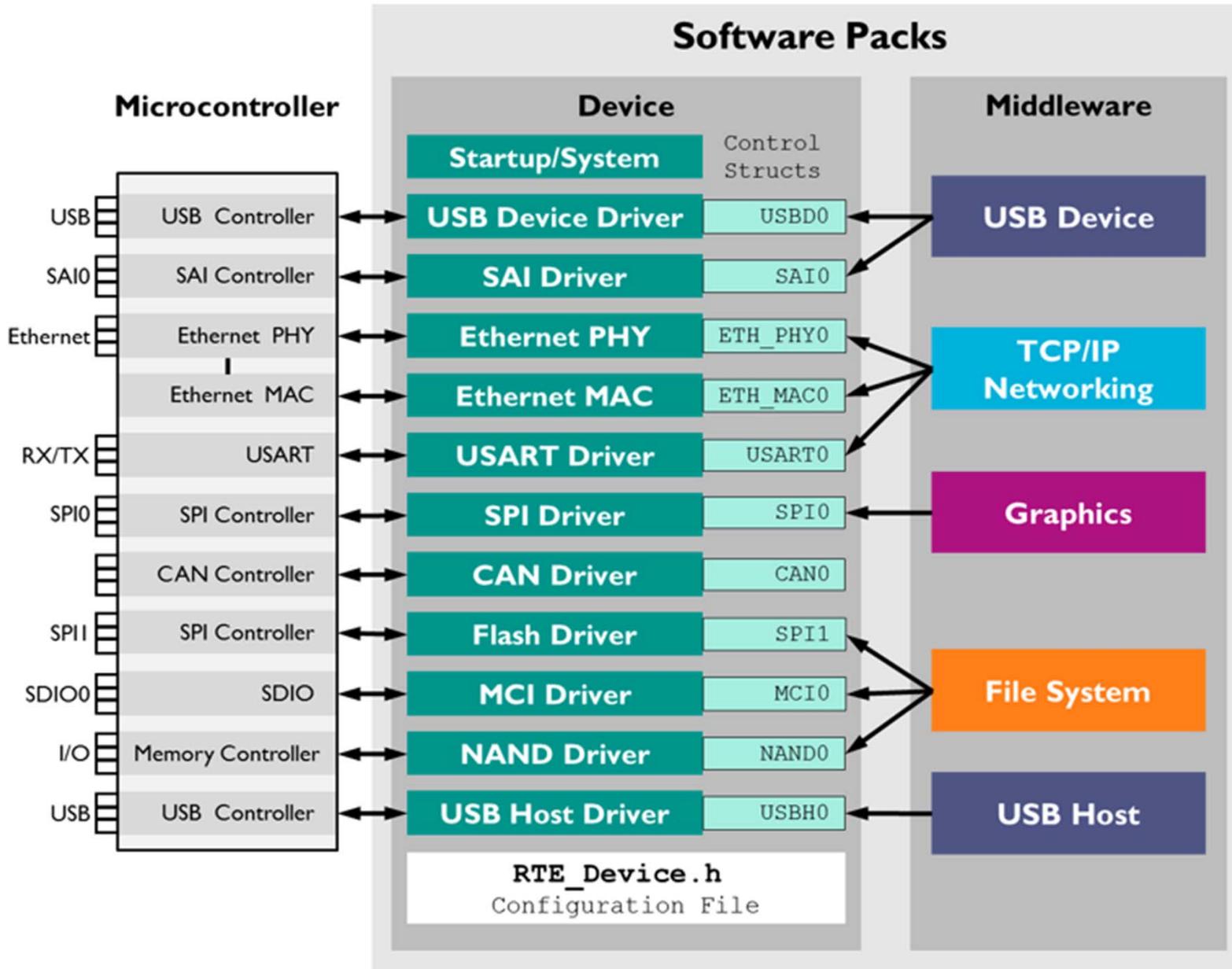
□ Redes de Ordenadores

- Protocolo TCP/IP
- Servicios

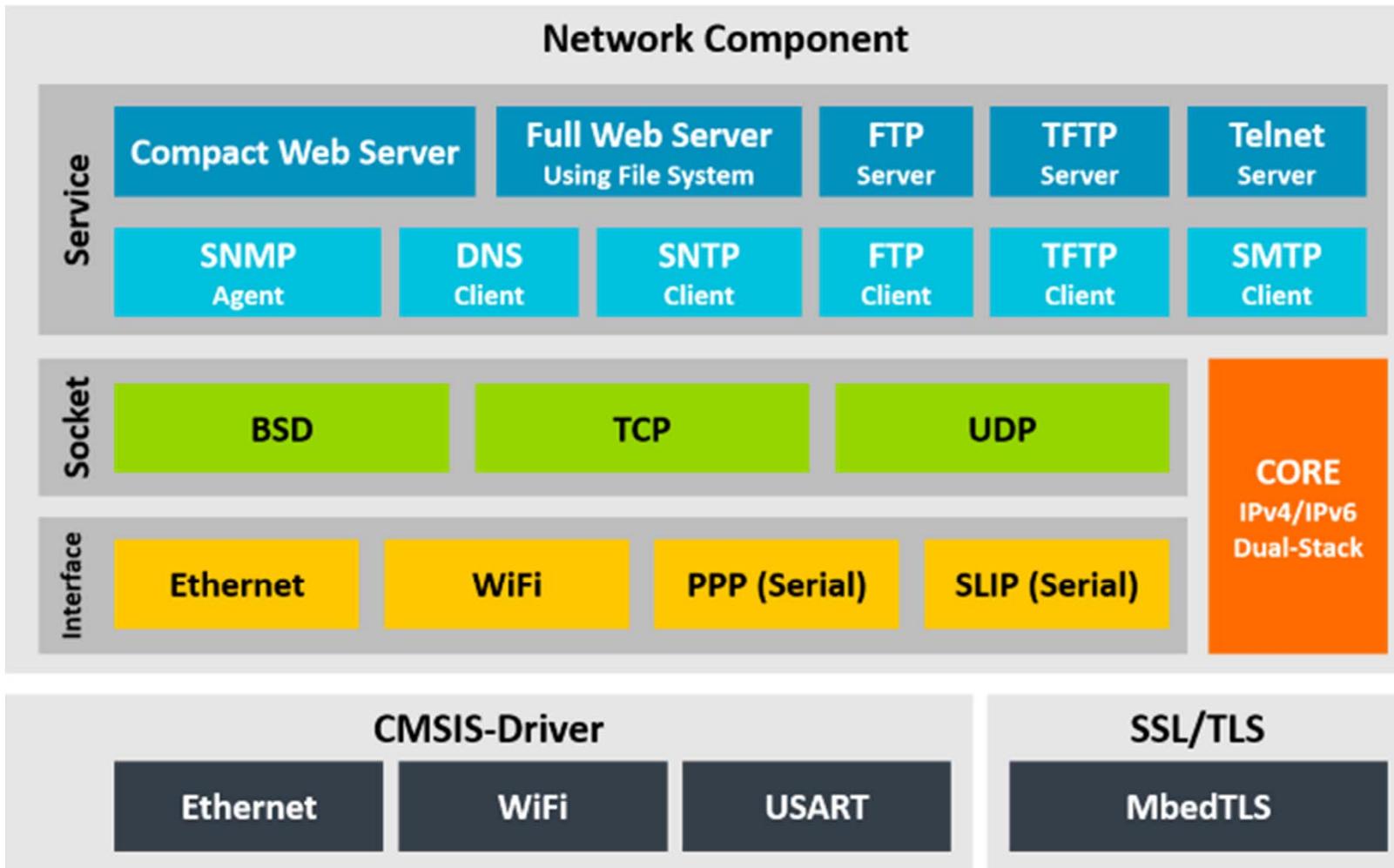
CMSIS. Recordatorio



Software Pack



Network Component



Network Component Version 7.15.0
MDK Middleware for IPv4 and IPv6 Networking

Network Component



Run-Time environment

Software Component	Sel.	Variant	Version	Description
Interface				Connection Mechanism
ETH	1		7.15.0	Network Ethernet Interface
PPP		Custom Modem	7.15.0	Network PPP over Serial Interface
SLIP		Custom Modem	7.15.0	Network SLIP Interface
WiFi	0		7.15.0	Network WiFi Interface
Service				Network Services
DNS Client			7.15.0	DNS Client
FTP Client			7.15.0	FTP Client
FTP Server			7.15.0	FTP Server
SMTP Client			7.15.0	Email Client (SMTP)
SNMP Agent			7.15.0	SNMP Agent
SNTP Client			7.15.0	SNTP Client
TFTP Client			7.15.0	TFTP Client
TFTP Server			7.15.0	TFTP Server
Telnet Server			7.15.0	Telnet Server
Web Server Compact		HTTP	7.15.0	Web Server (HTTP) with Read-only Web Resources (Compact)
Web Server		HTTP	7.15.0	Web Server (HTTP) with Web Resources on File System
Socket				Network Sockets
BSD			7.15.0	BSD Socket
TCP			7.15.0	TCP Socket
UDP			7.15.0	UDP Socket



Pack installer

+ Keil::LPCXpresso55S69...	Install+	NXP LPC55S69 Series LPCXpresso55S69 Board Support Pack
- Keil::MDK-Middleware	Update+	Middleware for Keil MDK-Professional and MDK-Plus
7.16.0 (2022-11-11)	Install+	Middleware for Keil MDK-Professional and MDK-Plus
7.13.0 (2021-05-25)	Remove	Middleware for Keil MDK-Professional and MDK-Plus
7.12.0 (2020-07-01)	Remove	Middleware for Keil MDK-Professional and MDK-Plus
7.10.0 (2019-11-08)	Remove	Middleware for Keil MDK-Professional and MDK-Plus
7.4.1 (2017-04-21)	Remove	Middleware for Keil MDK-Professional and MDK-Plus
+ Previous		Keil::MDK-Middleware - Previous Pack Versions

Network Component Version 7.15.0
MDK Middleware for IPv4 and IPv6 Networking

Network Component

C:/Keil_v5/ARM/PACK/Keil/MDK-Middleware/7.13.0/Doc/Network/html/index.html

www.keil.com/pack/doc/mw/Network/html/index.html (**online - 7.18!!!**)

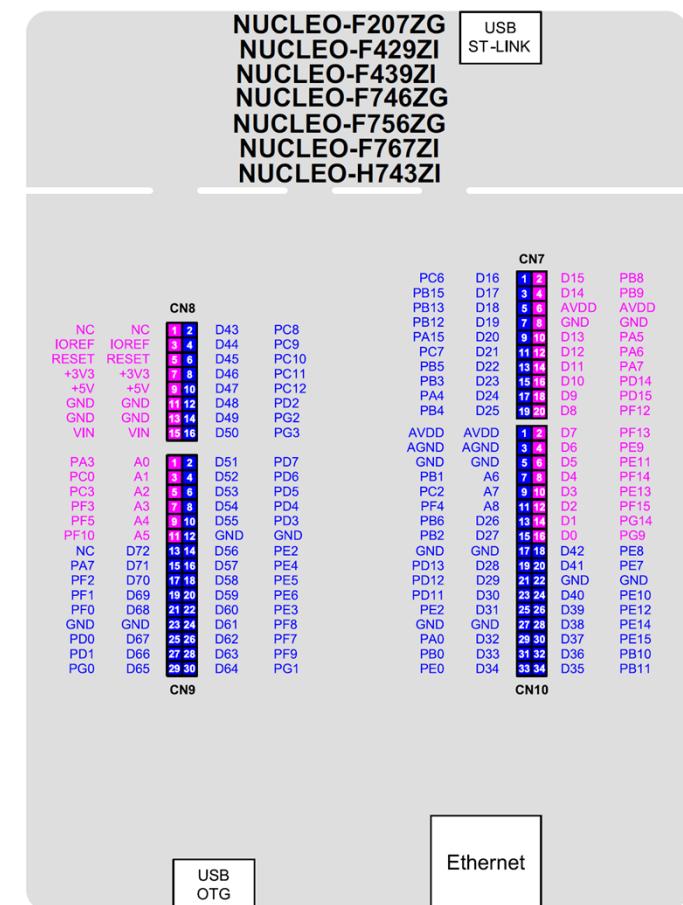
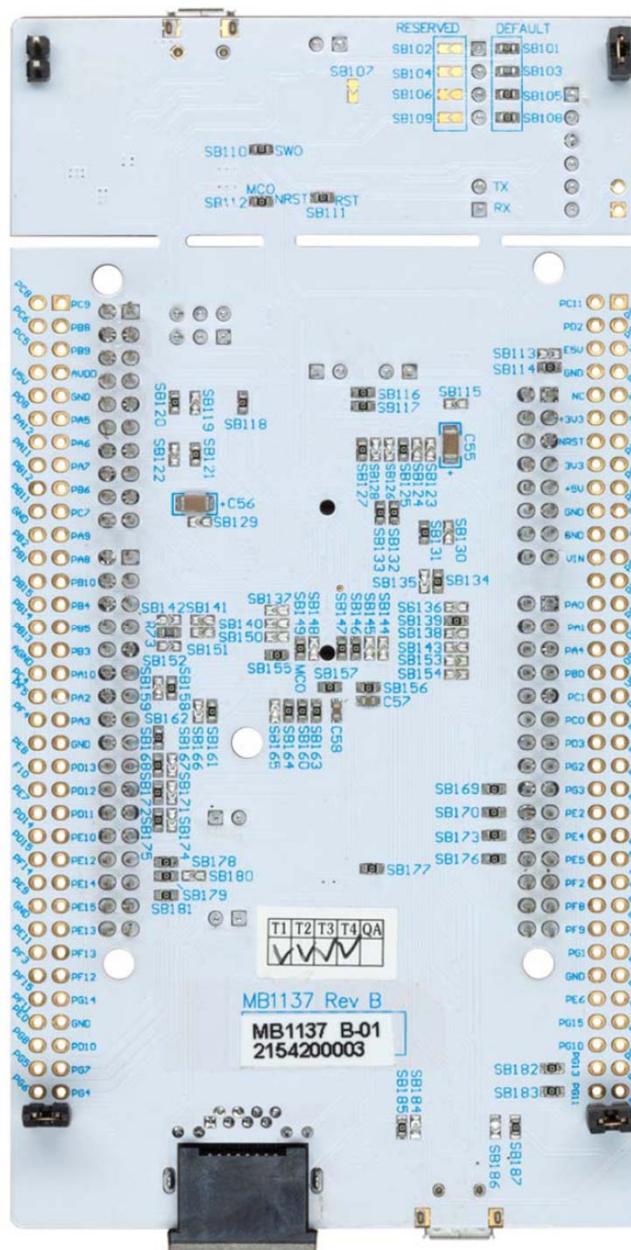
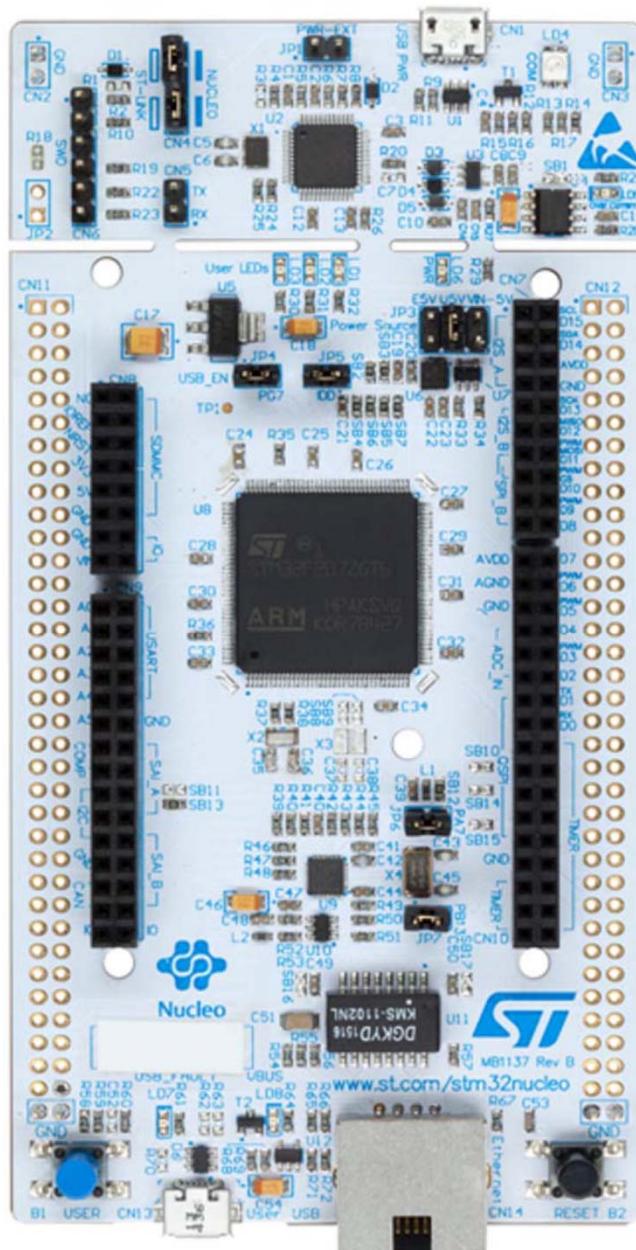
The screenshot shows the Network Component Documentation page. At the top, there's a navigation bar with tabs: General, File System, Graphic, Network (which is selected), USB, and Board Support. Below the tabs, there are sub-tabs: Main Page, Usage and Description, and Reference. A search bar is also present. The main content area has a title "Network Component Documentation". It includes a brief description: "The Network Component v7 contains services, protocol sockets, and physical communication interfaces for creating IPv4 and IPv6 networking applications." Below this is a diagram titled "Network Component" showing the architecture. The diagram is organized into three horizontal layers: "Service", "Socket", and "Interface". The "Service" layer contains boxes for Compact Web Server, Full Web Server Using File System, FTP Server, TFTP Server, Telnet Server, SNMP Agent, DNS Client, SNTP Client, FTP Client, TFTP Client, and SMTP Client. The "Socket" layer contains boxes for BSD, TCP, and UDP. The "Interface" layer contains boxes for Ethernet, WiFi, PPP (Serial), and SLIP (Serial). To the right of the interface layer is a vertical orange box labeled "CORE IPv4/IPv6 Dual-Stack". Below the diagram is a section titled "Network Overview" which lists CMSIS-Driver components (Ethernet, WiFi, USART) and SSL/TLS components (MbedTLS). At the bottom of the page, there's a note about services providing program templates for common networking tasks, followed by a bulleted list of features.

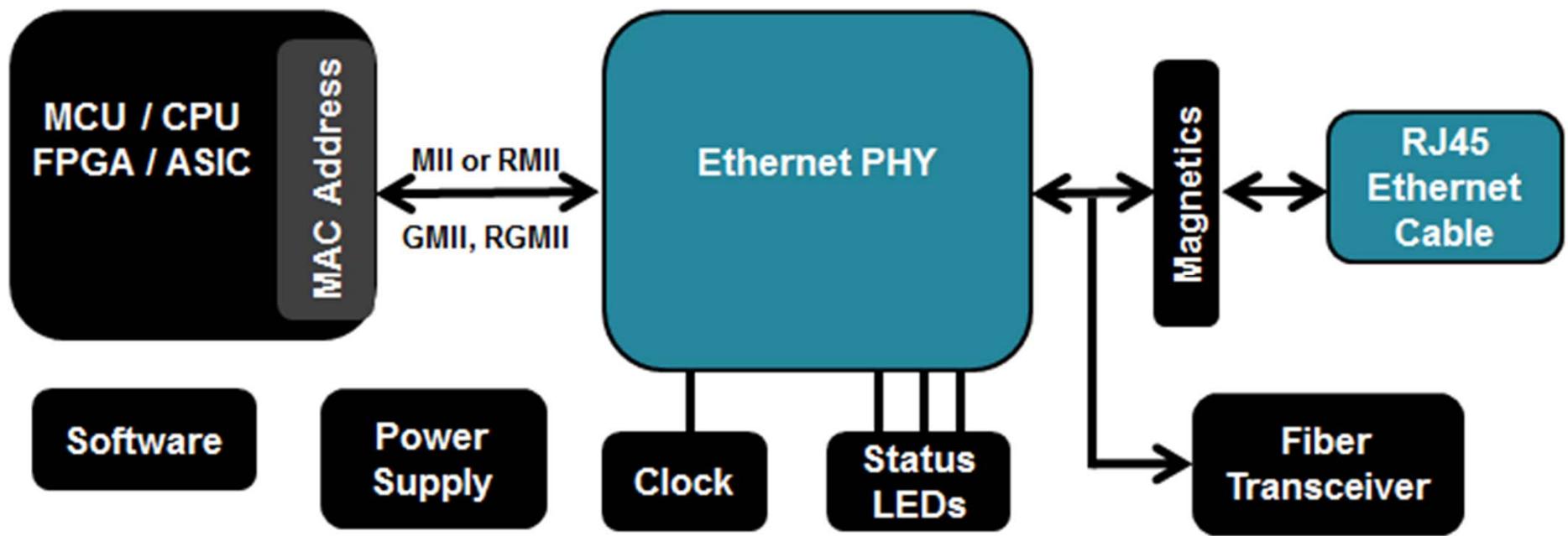
The Network Component Documentation page is displayed in a web browser window. The URL is https://www.keil.com/pack/doc/mw/Network/html/index.html. The page title is "Network Component Version 7.15.0". The main content area is titled "Network Component Documentation". It includes a brief description of the Network Component v7 and a detailed diagram of its architecture. The diagram shows the Network Component as a stack of services, sockets, and interfaces. The services include Compact Web Server, Full Web Server Using File System, FTP Server, TFTP Server, Telnet Server, SNMP Agent, DNS Client, SNTP Client, FTP Client, TFTP Client, and SMTP Client. The sockets include BSD, TCP, and UDP. The interfaces include Ethernet, WiFi, PPP (Serial), and SLIP (Serial). A vertical orange box on the right is labeled "CORE IPv4/IPv6 Dual-Stack". Below the diagram is a section titled "Network Overview" which lists CMSIS-Driver components (Ethernet, WiFi, USART) and SSL/TLS components (MbedTLS). At the bottom of the page, there's a note about services providing program templates for common networking tasks, followed by a bulleted list of features.

The various **services** provide program templates for common networking tasks:

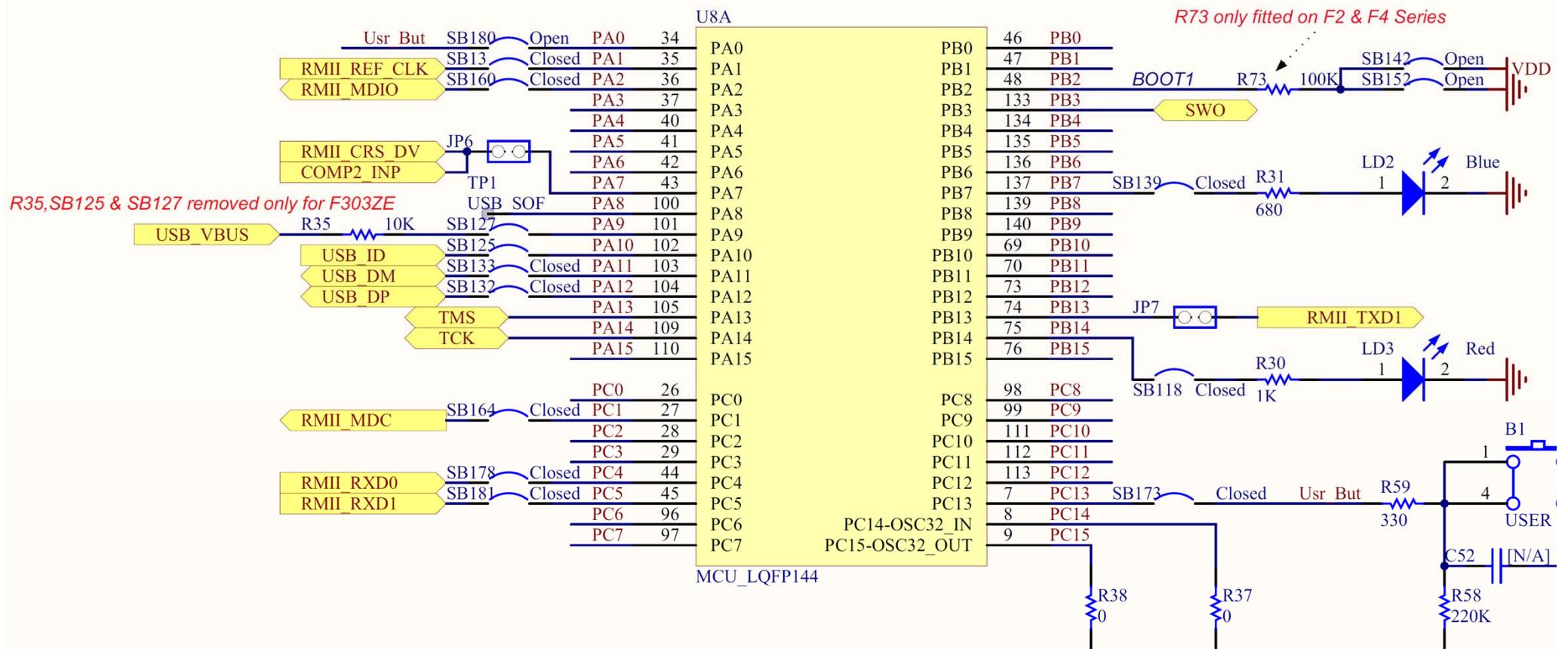
- The **HTTP Server** supports static and dynamic page content using **scripting**, **AJAX** and **SOAP** technologies. Web resource files are accessed either via the File System (for component **Web Server**) or are directly stored in ROM (for component **Web Server Compact**). Using the **ARM mbed TLS** component, you can enable secure communication for both web servers using HTTPS.
- **FTP server/client** and **TFTP server/client** are used for file transfer. TFTP supports only file upload and download and is typically used for boot-loading remote devices in a closed LAN. It has very low memory requirements. FTP provides full file access including sub-directories, log-in and a set of manipulation commands.

Generated on Tue May 25 2021 13:57:41 for Network Component by ARM Ltd. All rights reserved.

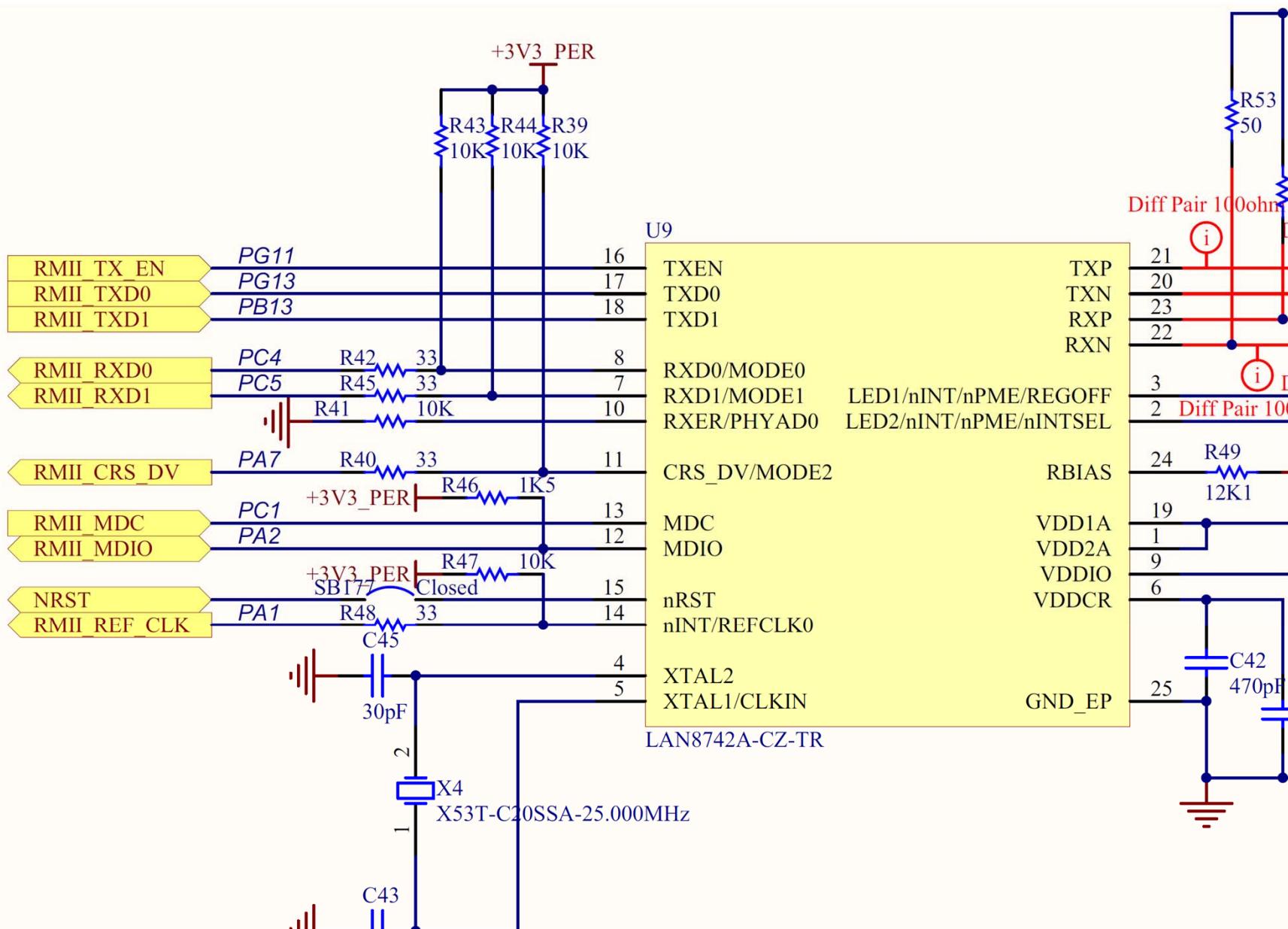




STM32F429Zi PinOut



PHY - PinOut



C:\Keil_v5\Ejemplos\Boards\Keil\MCB1700\Middleware\Network\HTTP_Server\HTTP_Server.uvprojx - μVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

LPC1768 Flash ARM_Driver_ETH_PHY_

Project

Project: HTTP_Server

LPC1768 Flash

Source

- HTTP_Server.c
- HTTP_Server_CGI.c
- Web.c
- SNMP_Agent_MIB.c
- Telnet_Server_UIF.c

Web files

Documentation

- Abstract.txt

Board Support

CMSIS

CMSIS Driver

Compiler

Device

File System

Network

- Net_Config.c (CORE)
- Net_Config_ETH_0.h (Interface:ETH)
- Net_Config_HTTP_Server.h (Service:Web Server Co)
- Net_Config_TCP.h (Socket:TCP)
- Net_Config_UDP.h (Socket:UDP)
- Net_CM3_L.lib (CORE)
- FTP_Client_FS.c (Service:FTP Client)
- FTP_Server_FS.c (Service:FTP Server)
- TFTP_Client_FS.c (Service:TFTP Client)
- TFTP_Server_FS.c (Service:TFTP Server)
- Net_Config_DNS_Client.h (Service:DNS Client)
- Net_Config_FTP_Client.h (Service:FTP Client)
- Net_Config_FTP_Server.h (Service:FTP Server)
- Net_Config_SMTP_Client.h (Service:SMTP Client)
- Net_Config_SNMP_Agent.h (Service:SNMP Agent)
- Net_Config_SNTP_Client.h (Service:SNTP Client)
- Net_Config_TFTP_Client.h (Service:TFTP Client)
- Net_Config_TFTP_Server.h (Service:TFTP Server)
- Net_Config_Telnet_Server.h (Service:Telnet Server)

Telnet_Server_UIF.c

```
1 /*-----  
2 * MDK Middleware - Component ::Network::Service  
3 * Copyright (c) 2004-2019 Arm Limited (or its affiliates). All rights reserved.  
4 *-----  
5 * Name: Telnet_Server_UIF.c  
6 * Purpose: Telnet Server User Interface  
7 * Rev.: V6.00  
8 */  
9 //! [code_Telnet_Server_UIF]  
10 #include <stdio.h>  
11 #include "rl_net.h"  
12  
13 // Request message for Telnet server session.  
14 uint32_t telnet_server_message (telnetServerMessage msg, char *buf, uint32_t len) {  
15     uint32_t rlen = 0;  
16  
17     switch (msg) {  
18         case telnetServerWelcome:  
19             // Initial welcome message  
20             /* Example  
21                 rlen = sprintf (buf, "\r\n"  
22                             "\r\n"  
23                             "Telnet Server ready\r\n");  
24             */  
25             break;  
26         case telnetServerPrompt:  
27             // Prompt message  
28             /* Example  
29                 rlen = sprintf (buf, "\r\n"  
30                             "Cmd> ");  
31             */  
32             break;  
33         case telnetServerLogin:  
34             // Login welcome message, if authentication is enabled  
35             /* Example  
36                 rlen = sprintf (buf, "\r\n"  
37                             "Embedded Telnet Server\r\n"  
38                             "\r\n"  
39                             "Please login...");  
40             */  
41             break;  
42         case telnetServerUsername:  
43             // Username request login message  
44             /* Example  
45                 rlen = sprintf (buf, "\r\n"  
46                             "Username: ");  
47             */  
48             break;  
49         case telnetServerPassword:  
50             // Password request login message  
51             /* Example  
52                 rlen = sprintf (buf, "\r\n"  
53                             "Password: ");  
54             */
```

CMSIS DRIVER

Manage Run-Time Environment				
Software Component	Sel.	Variant	Version	Description
Board Support		MCB1700	1.0.0	Keil Development Board MCB1700
CMSIS				Cortex Microcontroller Software Interface Components
CMSIS Driver				Unified Device Drivers compliant to CMSIS-Driver Specifications
CAN (API)			1.2.0	CAN Driver API for Cortex-M
Ethernet (API)			2.1.0	Ethernet MAC and PHY Driver API for Cortex-M
Custom			2.1.0	Access to #include Driver_ETH_PHY/MAC.h files and code templates for custom implement...
KSZ8851SNL			6.6.0	Ethernet MAC + PHY KSZ8851SNL/SNLI Driver
LAN9220			1.0.0	Ethernet MAC + PHY LAN9220 Driver
RNDIS			1.1.0	Ethernet MAC + PHY USB Device RNDIS Driver
Ethernet MAC (API)			2.1.0	Ethernet MAC Driver API for Cortex-M
Custom			2.1.0	Access to #include Driver_ETH_MAC.h file and code template for custom implementation
Ethernet MAC	<input checked="" type="checkbox"/>		2.9	Ethernet MAC Driver for LPC1700 Series
Ethernet PHY (API)			2.1.0	Ethernet PHY Driver API for Cortex-M
Custom			2.1.0	Access to #include Driver_ETH_PHY.h file and code template for custom implementation
DP83848C	<input checked="" type="checkbox"/>		6.2.0	Ethernet PHY DP83848C Driver
KSZ8061RNB			1.3.0	Ethernet PHY KSZ8061RNB Driver
KSZ8081RNA			6.3.0	Ethernet PHY KSZ8081RNA Driver
LAN8710A			1.0.0	Ethernet PHY LAN8710A Driver
LAN8720			6.2.0	Ethernet PHY LAN8720 Driver
LAN8742A			1.2.0	Ethernet PHY LAN8742A Driver
ST802RT1			6.2.0	Ethernet PHY ST802RT1 Driver
Flash (API)			2.2.0	Flash Driver API for Cortex-M
I2C (API)			2.3.0	I2C Driver API for Cortex-M
MCI (API)			2.3.0	MCI Driver API for Cortex-M
NAND (API)			2.3.0	NAND Flash Driver API for Cortex-M
SAI (API)			1.1.0	SAI Driver API for Cortex-M
SPI (API)			2.2.0	SPI Driver API for Cortex-M
USART (API)			2.3.0	USART Driver API for Cortex-M
USB Device (API)			2.2.0	USB Device Driver API for Cortex-M
USB Host (API)			2.2.0	USB Host Driver API for Cortex-M

Ethernet Driver

```
ARM_DRIVER_ETH_PHY  
ARM_Driver_ETH_PHY_(ETH_PHY_NUM) = {  
    GetVersion,  
    Initialize,  
    Uninitialize,  
    PowerControl,  
    SetInterface,  
    SetMode,  
    GetLinkState,  
    GetLinkInfo  
};
```

MAC Driver

```
/* MAC Driver Control Block */
ARM_DRIVER_ETH_MAC Driver_ETH_MAC0 = {
    GetVersion,
    GetCapabilities,
    Initialize,
    Uninitialize,
    PowerControl,
    GetMacAddress,
    SetMacAddress,
    SetAddressFilter,
    SendFrame,
    ReadFrame,
    GetRxFrameSize,
    GetRxFrameTime,
    GetTxFrameTime,
    ControlTimer,
    Control,
    PHY_Read,
    PHY_Write
};
```

RTE_Device

Option	Value
+---SPI1 (Serial Peripheral Interface 1) [Driver_SPI1]	<input type="checkbox"/>
+---SPI2 (Serial Peripheral Interface 2) [Driver_SPI2]	<input type="checkbox"/>
+---SPI3 (Serial Peripheral Interface 3) [Driver_SPI3]	<input type="checkbox"/>
+---SPI4 (Serial Peripheral Interface 4) [Driver_SPI4]	<input type="checkbox"/>
+---SPI5 (Serial Peripheral Interface 5) [Driver_SPI5]	<input type="checkbox"/>
+---SPI6 (Serial Peripheral Interface 6) [Driver_SPI6]	<input type="checkbox"/>
+---SDIO (Secure Digital Input/Output) [Driver_MCI0]	<input type="checkbox"/>
+---CAN1 (Controller Area Network 1) [Driver_CAN1]	<input type="checkbox"/>
+---CAN2 (Controller Area Network 2) [Driver_CAN2]	<input type="checkbox"/>
+---CAN3 (Controller Area Network 3) [Driver_CAN3]	<input type="checkbox"/>
-ETH (Ethernet Interface) [Driver_ETH_MAC0]	<input checked="" type="checkbox"/>
+---MII (Media Independent Interface)	<input type="checkbox"/>
+---RMII (Reduced Media Independent Interface)	<input type="checkbox"/>
+---Management Data Interface	
+---USB OTG Full-speed	<input type="checkbox"/>
+---USB OTG High-speed	<input type="checkbox"/>

Network Component

Software Component	Sel.	Variant	Version	Description
Interface				Connection Mechanism
ETH	1		7.15.0	Network Ethernet Interface
PPP	<input type="checkbox"/>	Custom Modem	7.15.0	Network PPP over Serial Interface
SLIP	<input type="checkbox"/>	Custom Modem	7.15.0	Network SLIP Interface
WiFi	0		7.15.0	Network WiFi Interface
Service				Network Services
DNS Client	<input type="checkbox"/>		7.15.0	DNS Client
FTP Client	<input type="checkbox"/>		7.15.0	FTP Client
FTP Server	<input type="checkbox"/>		7.15.0	FTP Server
SMTP Client	<input type="checkbox"/>	SMTP	7.15.0	Email Client (SMTP)
SNMP Agent	<input type="checkbox"/>		7.15.0	SNMP Agent
SNTP Client	<input type="checkbox"/>		7.15.0	SNTP Client
TFTP Client	<input type="checkbox"/>		7.15.0	TFTP Client
TFTP Server	<input type="checkbox"/>		7.15.0	TFTP Server
Telnet Server	<input type="checkbox"/>		7.15.0	Telnet Server
Web Server Compact	<input checked="" type="checkbox"/>	HTTP	7.15.0	Web Server (HTTP) with Read-only Web Resources (Compact)
Web Server	<input type="checkbox"/>	HTTP	7.15.0	Web Server (HTTP) with Web Resources on File System
Socket				Network Sockets
BSD	<input type="checkbox"/>		7.15.0	BSD Socket
TCP	<input checked="" type="checkbox"/>		7.15.0	TCP Socket
UDP	<input checked="" type="checkbox"/>		7.15.0	UDP Socket

Net_Config_ETH_0

Option	Value
Ethernet Network Interface 0	
Connect to hardware via Driver_ETH#	0
MAC Address	1E-30-6C-A2-45-5E
VLAN	<input type="checkbox"/>
VLAN Identifier	1
IPv4	<input checked="" type="checkbox"/>
IP Address	192.168.10.249
Subnet mask	255.255.255.0
Default Gateway	192.168.10.1
Primary DNS Server	8.8.8.8
Secondary DNS Server	8.8.4.4
IP Fragmentation	<input checked="" type="checkbox"/>
MTU size	1500
ARP Address Resolution	
Cache Table size	10
Cache Timeout in seconds	150
Number of Retries	4
Resend Timeout in seconds	2
Send Notification on Address changes	<input type="checkbox"/>
IGMP Group Management	<input checked="" type="checkbox"/>
Membership Table size	5
NetBIOS Name Service	<input checked="" type="checkbox"/>
Dynamic Host Configuration	<input type="checkbox"/>
Vendor Class Identifier	
Bootfile Name	<input type="checkbox"/>
Ethernet Network Interface 0	

NetConfig

Option	Value
Network System Settings	
Local Host Name	mcbstm32f400
Memory Pool Size	12000
Start System Services	<input checked="" type="checkbox"/>
OS Resource Settings	
Core Thread Stack Size	1024

Net_Config_HTTP

Option	Value
Network System Settings	
Local Host Name	mcbstm32f400
Memory Pool Size	12000
Start System Services	<input checked="" type="checkbox"/>
OS Resource Settings	
Core Thread Stack Size	1024

Net_Config_TCP / Net_Config_UDP

Option	Value
└ TCP Sockets	
Number of TCP Sockets	6
Number of Retries	5
Retry Timeout in seconds	4
Default Connect Timeout in seconds	120
Maximum Segment Size	1440
Receive Window Size	4320

Option	Value
└ UDP Sockets	
Number of UDP Sockets	5

Inicialización de la red

The screenshot shows a web browser window displaying the **arm KEIL Network Component** documentation. The title bar includes the URL https://www.keil.com/pack/doc/mw/Network/html/group_net_sys_func.html#ga0c68e74df79a37750413f72518f63f73. The main content area is titled "Network Component Version 7.15.0 MDK Middleware for IPv4 and IPv6 Networking". The left sidebar contains a navigation tree with sections like General, File System, Graphic, Network (which is selected), USB, Board Support, Main Page, Usage and Description, and Reference. The right panel details the **netStatus netInitialize (void)** function. It describes the function as initializing the Network Component and interfaces, noting it is **[not_thread-safe]**. The **Returns** section specifies a status code indicating execution status. A note states that the function must be called at system start-up to initialize the networking environment. Possible return values are **netOK** (success) and **netError** (failure). A code example shows how to call the function in a C main loop. Below this, the **const char * netSYS_GetHostName (void)** function is described, which retrieves the localhost name. A code example shows a printf statement using the function.

System Functions (User)

Network Component

MDK Middleware for IPv4 and IPv6 Networking

General **File System** **Graphic** **Network** **USB** **Board Support**

Main Page Usage and Description Reference

Search

netStatus netInitialize (void)

Initialize Network Component and interfaces. **[not_thread-safe]**.

Returns
status code that indicates the execution status of the function.

The function **netInitialize** initializes the Network Core's system resources, protocols, threads and applications. Since the function also creates RTOS objects, the RTOS must be previously initialized.

Note
You must call the function at system start-up to properly initialize the networking environment.

Possible **netStatus** return values:

- **netOK**: Network Core initialized successfully.
- **netError**: Failed to initialize the Network Core.

Code Example

```
int main (void) {  
    init_hal ();  
    // Initialize the Network Core  
    netInitialize ();  
    ...  
}
```

const char * netSYS_GetHostName (void)

Retrieve localhost name. **[thread-safe]**.

Returns
pointer to localhost name, a null-terminated string.

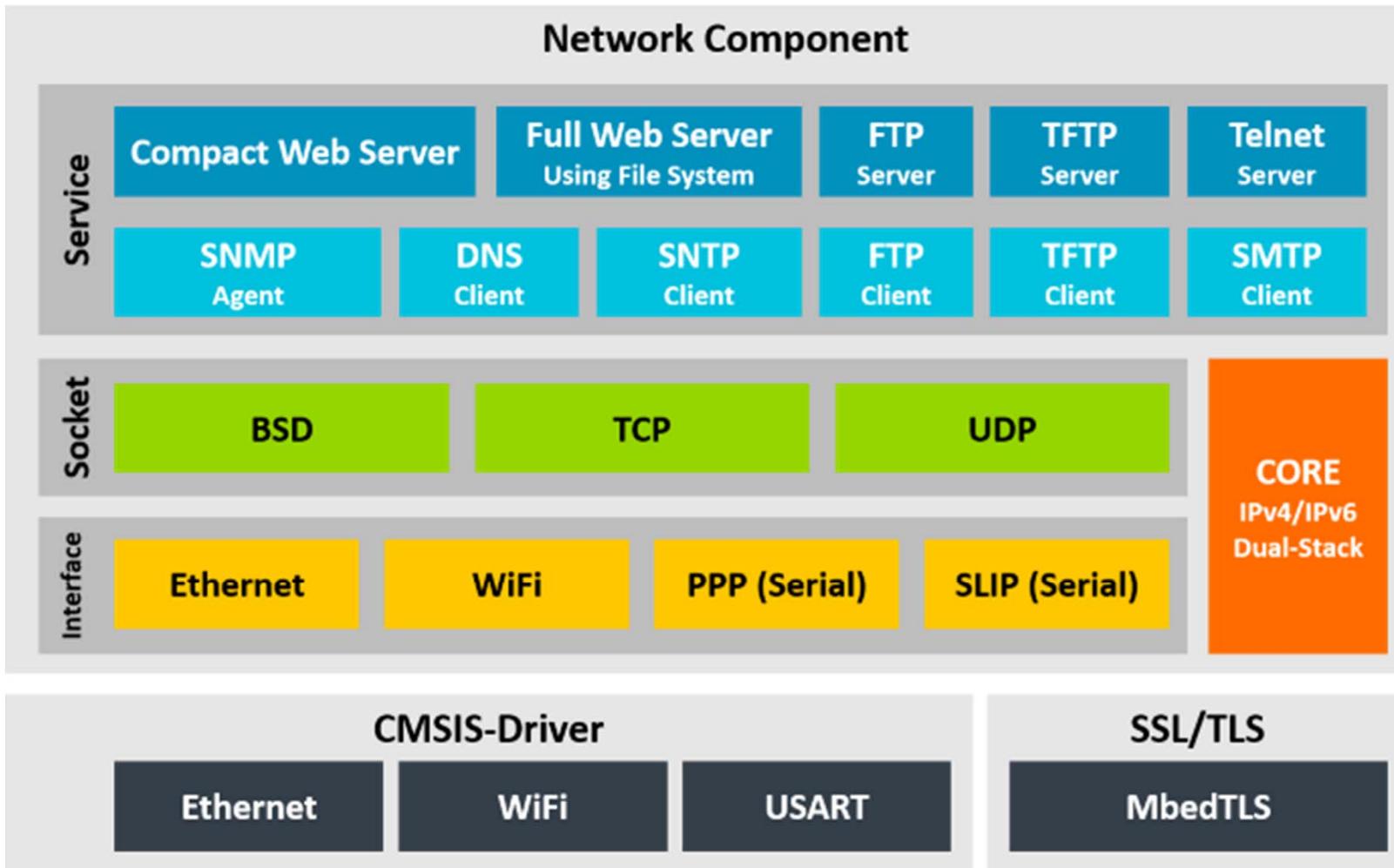
The function **netSYS_GetHostName** returns the name of the local host. The name is set with **NET_HOST_NAME** in the **Net_Config.c**, and can be changed at **runtime**.

Code Example

```
printf ("Localhost name is %s\n", netSYS_GetHostName ());
```

Generated on Tue May 25 2021 13:57:39 for Network Component by ARM Ltd. All rights reserved.

Network Component

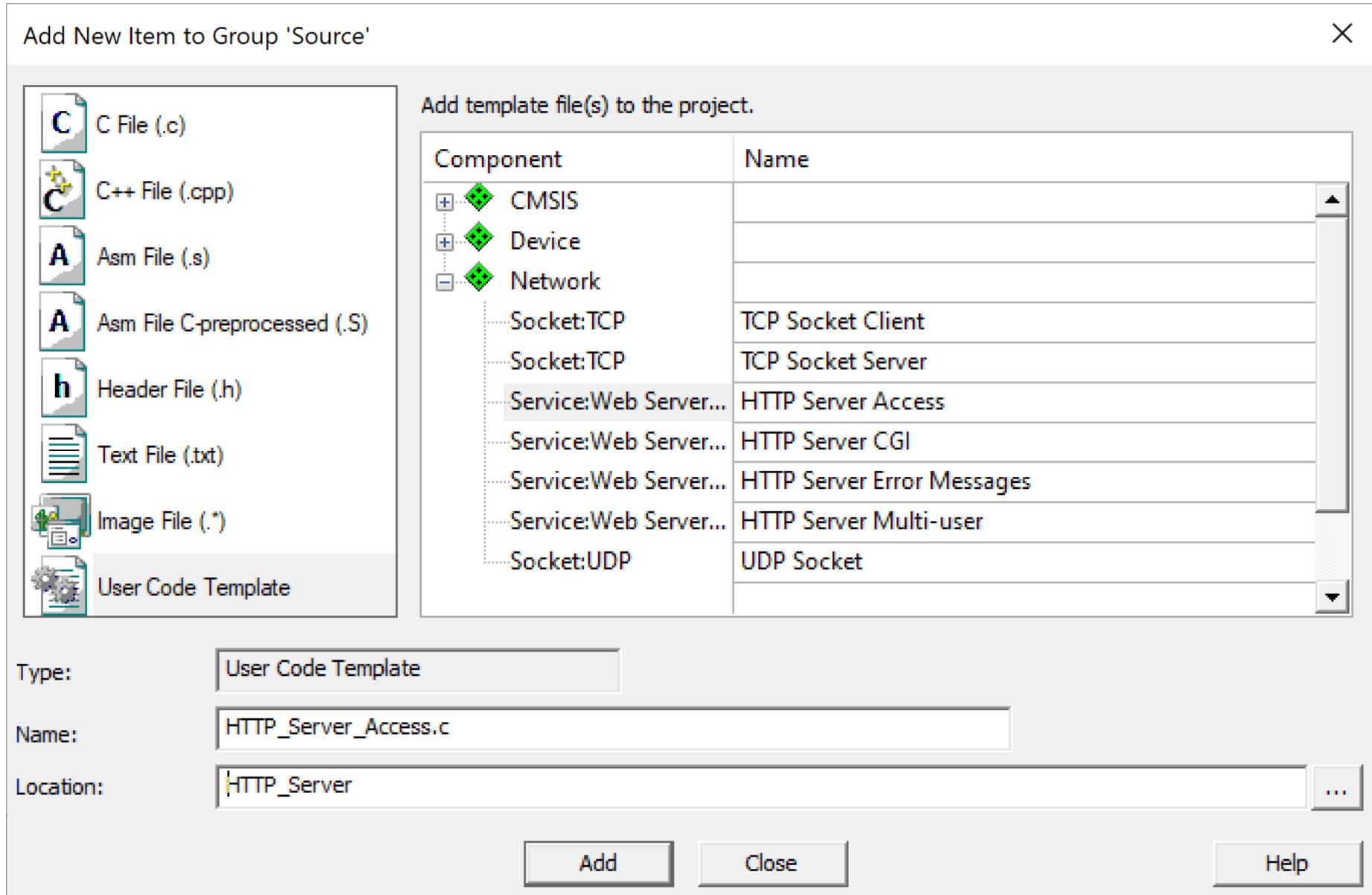


Network Component Version 7.15.0
MDK Middleware for IPv4 and IPv6 Networking

Entorno de desarrollo

- Keil 5.34.0.0
- STM32F4XX_DFP 2.15.0(2020-09-28)
- CMSIS 5.8.0
- CMSIS Driver 2.6.1
- ARM_Compiler 1.6.3
- MDK_Middleware 7.13.0
- RTOS 2 (RTX5)

Templates



Templates

The screenshot shows a web browser window displaying the **Network Component** documentation for MDK Middleware for IPv4 and IPv6 Networking, version 7.15.0. The URL is https://www.keil.com/pack/doc/mw/Network/html/network_create_app.html. The browser interface includes a menu bar with Archivo, Editar, Ver, Historial, Marcadores, Herramientas, Ayuda, and a toolbar with various icons.

The main content area features a navigation bar with tabs: General, File System, Graphic, Network (selected), USB, and Board Support. Below the tabs are sub-sections: Main Page, Usage and Description (selected), and Reference. A search bar is located at the top right.

The left sidebar contains a tree view of documentation categories: Network Component (selected), Revision History, Creating a Network Application (selected), Troubleshooting a Network Application, Secure Communication, Cyber Security, Network Examples, Migration, Resource Requirements, Function Overview, Reference, Data Structures, Data Structure Index, and Data Fields.

The right panel displays code snippets and documentation for User Code Templates. One snippet shows C code for PPP connection:

```
netPPP_Connect ("04213372", "Guest", "test")
osThreadTerminate(osThreadGetId());
for(;;);
```

User Code Templates

User code template files provide access to all functions that are required to communicate over the Network. The available functions are explained in the **Reference** section of the Network Component. These routines can be adapted to the needs of the microcontroller application, in case more functionality is needed.

The following templates are available for the Network component:

Template Name	User Functions
DNS_Client.c	dns_cbfunc (Callback function for notification about DNS client events), resolve_host (DNS resolving process)
FTP_Client_UIF.c	netFTPC_Process (Request parameters for FTP client session), netFTPC_Notify (Notify the user application when FTP client operation ends)
FTP_Server_Access.c	netFTPS_AcceptClient (Accept or deny connection from remote FTP client)
FTP_Server_Event.c	netFTPS_Notify (Notify the user application about events in FTP server service)
FTP_Server_Multiuser.c	netFTPS_CheckUsername (Check if an user account exists), netFTPS_CheckPassword (Check user account password), netFTPS_FileAccess (Check if remote user is allowed to access a file)
HTTP_Server_Access.c	netHTTPs_AcceptClient (Accept or deny connection from remote HTTP client)
HTTP_Server_CGI.c	netCGI_ProcessQuery (Process query string received by GET request), netCGI_ProcessData (Process data received by POST request), netCGI_Script (Generate dynamic web data from a script line)
HTTP_Server_Error.c	net_http_error (Define user-friendly HTTP error messages)
HTTP_Server_Multiuser.c	netHTTPs_CheckAccount (Check if an user account exists), netHTTPs_FileAccess (Check if remote user is allowed to access a file)
SMTP_Client_UIF.c	netSMTPC_Process (Request parameters for SMTP client session), netSMTPC_Notify (Notify the user application when SMTP client operation ends), netSMTPC_AcceptAuthentication (Accept or deny authentication requested by SMTP server)
SNMP_Agent_MIB.c	mib_table (Defines MIB information data table), register_mib_table (Registers a MIB table in SNMP agent)
TCP_Socket_Client.c	tcp_cb_client (Notify the user application about TCP socket events), send_data (Connect to TCP server and send data)
TCP_Socket_Server.c	tcp_cb_server (Notify the user application about TCP socket events)
Telnet_Server_Access.c	netTELNETs_AcceptClient (Accept or deny connection from remote Telnet client)
Telnet_Server_Multiuser.c	netTELNETs_CheckUsername (Check if an user account exists), netTELNETs_CheckPassword (Check user account password)
Telnet_Server_UIF.c	netTELNETs_ProcessMessage (Request message for Telnet server session), netTELNETs_ProcessCommand (Process a command and generate response)

Generated on Tue May 25 2021 13:57:37 for Network Component by ARM Ltd. All rights reserved.

Resources Requirements

- http://www.keil.com/pack/doc/mw/Network/html/nw_resource_requirements.html
 - Stack
 - Heap
 - CMSIS-RTOS v2
 - RTX V5
 - Memory Requirements

Services API

The screenshot shows a web browser window with the following details:

- Title Bar:** Archivo Editar Ver Historial Marcadores Herramientas Ayuda
- Tab:** Function Overview
- Address Bar:** https://www.keil.com/pack/doc/mw/Network/html/net_func_overview.html
- Toolbar:** Back, Forward, Stop, Refresh, Search, etc.

The main content area displays the following information:

arm KEIL Network Component Version 7.15.0

MDK Middleware for IPv4 and IPv6 Networking

Navigation Tabs: General, File System, Graphic, Network (selected), USB, Board Support

Sub-navigation: Main Page, Usage and Description, Reference (selected)

Function Overview: The following list provides a brief overview of all Network Component functions.

HTTP Server

Control Interface

- `netHTTPs_GetClient` — Get IP address and port number of a connected remote HTTP client. [thread-safe].
- `netHTTPs_GetContentType` — Get Content-Type HTML header, received in XML post request. [thread-safe].
- `netHTTPs_GetLanguage` — Retrieve the preferred language setting from the browser. [thread-safe].
- `netHTTPs_GetPassword` — Retrieve password of the built-in user account. [thread-safe].
- `netHTTPs_GetPort` — Get port number of the HTTP server. [thread-safe].
- `netHTTPs_GetRootPath` — Retrieve path to the root directory of HTTP server. [thread-safe].
- `netHTTPs_GetSession` — Get current session number of the HTTP server. [thread-safe].
- `netHTTPs_GetUsername` — Retrieve username of the built-in user account. [thread-safe].
- `netHTTPs_LoginActive` — Determine if the HTTP server authentication is enabled. [thread-safe].
- `netHTTPs_LoginOnOff` — Enable or disable HTTP server authentication. [thread-safe].
- `netHTTPs_Running` — Check if the HTTP server is running. [thread-safe].
- `netHTTPs_SetPassword` — Reset password of the built-in user account. [thread-safe].
- `netHTTPs_SetPort` — Set port number of the HTTP server. [thread-safe].
- `netHTTPs_SetRootPath` — Set path to the root directory of HTTP server. [thread-safe].
- `netHTTPs_SetUsername` — Set username of the built-in user account. [thread-safe].
- `netHTTPs_Start` — Start the HTTP server. [thread-safe].
- `netHTTPs_Stop` — Stop the HTTP server. [thread-safe].

Access and Multi-User Interface

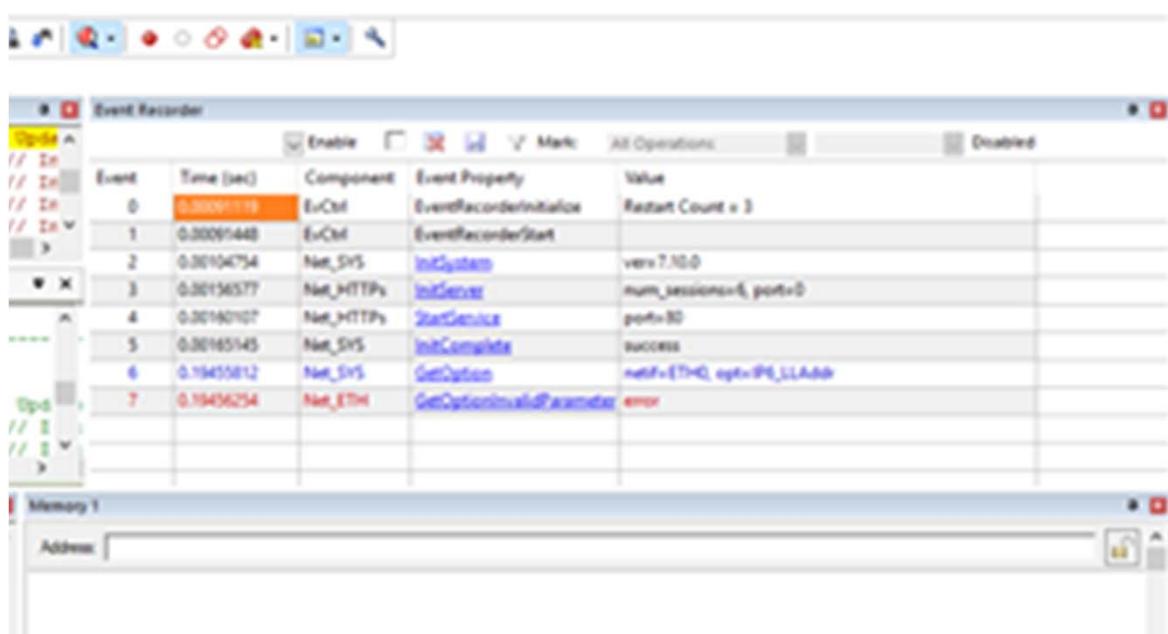
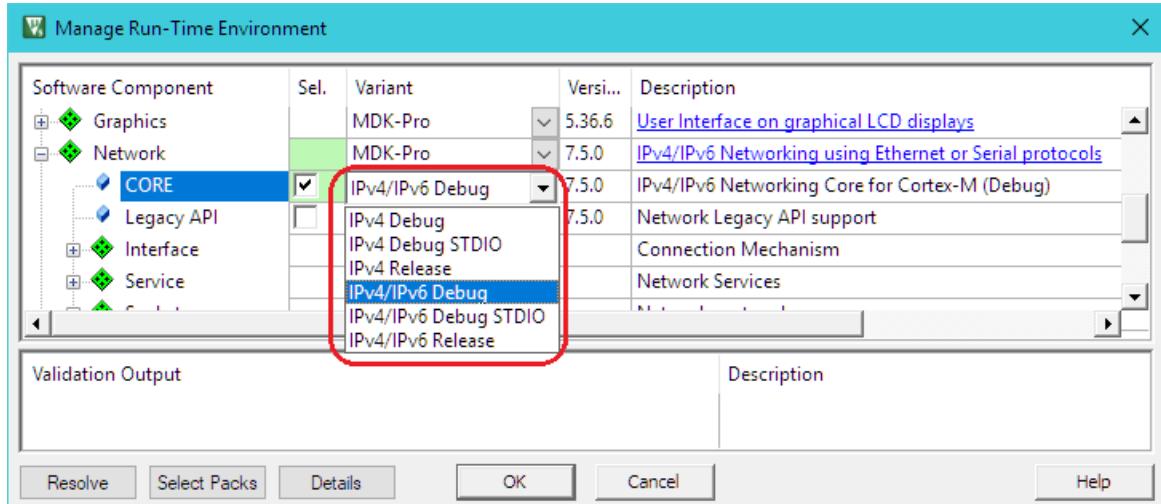
- `netHTTPs_AcceptClient` — Accept or deny a connection from a remote HTTP client. [user-provided].
- `netHTTPs_CheckAccount` — Check if an user account exist in the user database. [user-provided].
- `netHTTPs_GetUserSecret` — Retrieve the secret word for the selected user. [user-provided].
- `netHTTPs_FileAccess` — Check if remote user is allowed to access a file on HTTP server. [user-provided].
- `netHTTPs_GetUserId` — Retrieve the user identification. [thread-safe].
- `netHTTPs_CalcHashHA1` — Calculate HA1 hash value for the given credentials. [thread-safe].

File System Interface

- `netHTTPs_fclose` — Close a file previously open in HTTP server. [interface].

Generated on Tue May 25 2021 13:57:37 for Network Component by ARM Ltd. All rights reserved.

Trace. Event Recorder Window

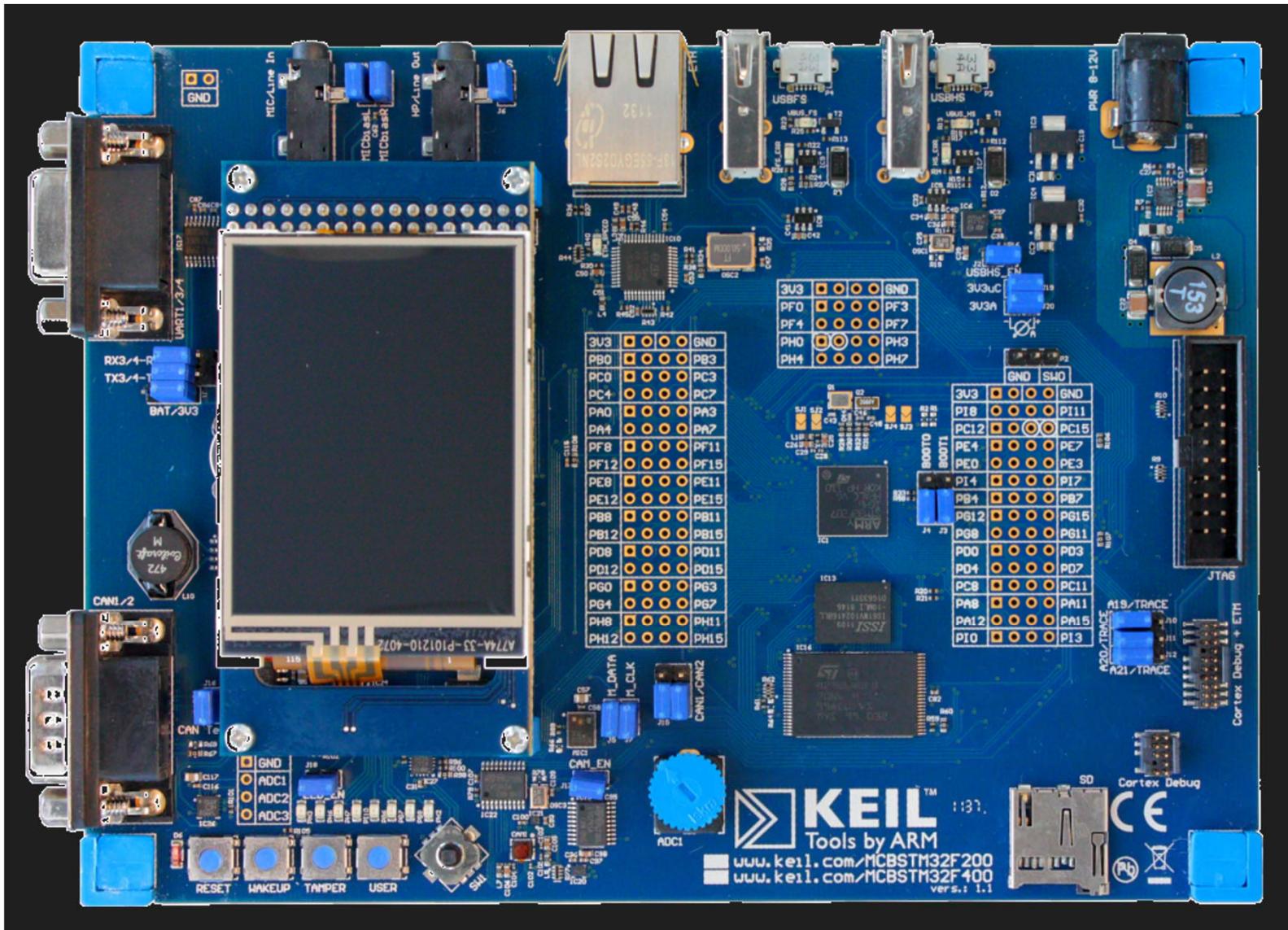


http://www.keil.com/pack/doc/compiler/EventRecorder/html/er_overview.html#AboutEventRecorder

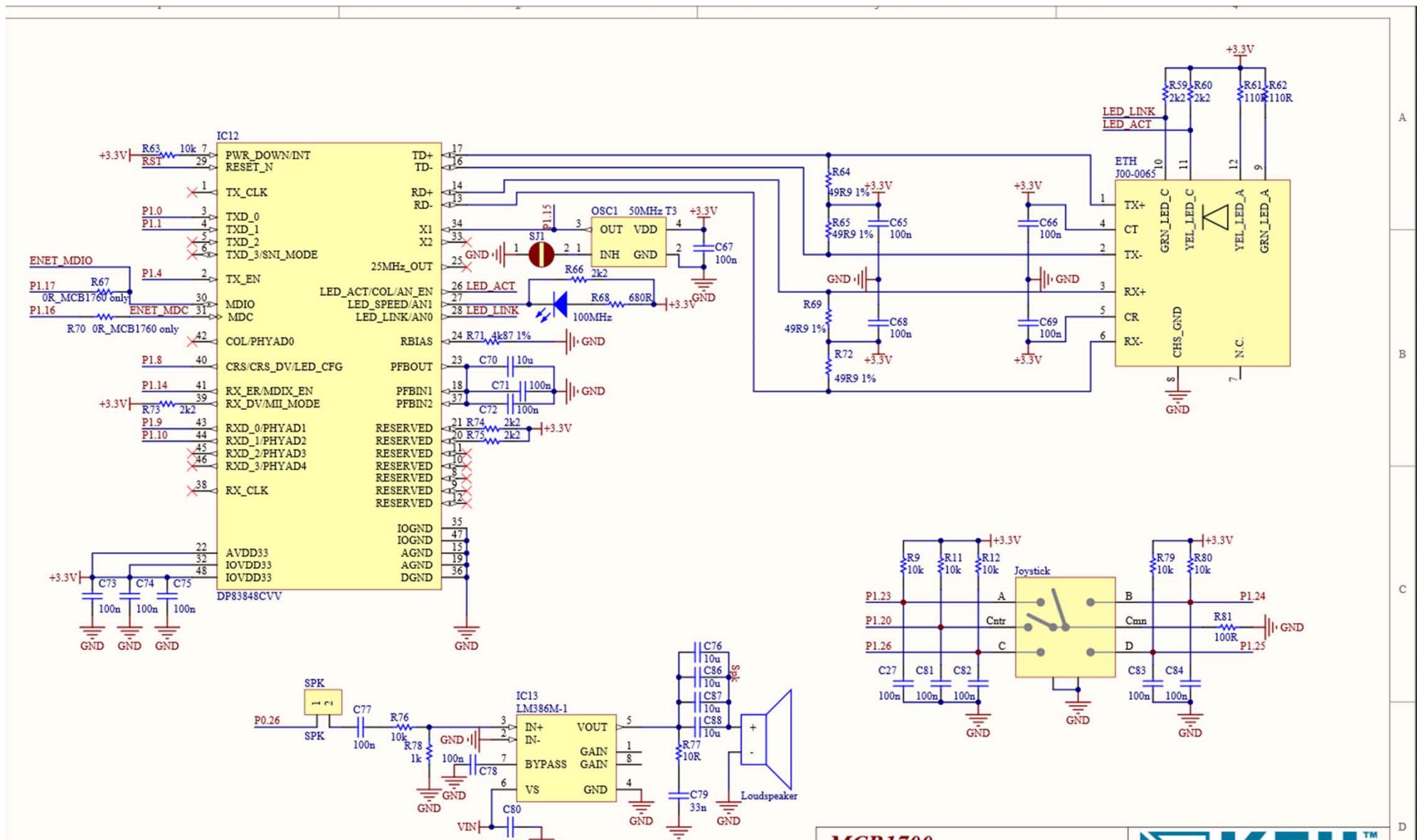
MCBSTM32F200_F400

<https://developer.arm.com/documentation/101679/latest>

<https://documentation-service.arm.com/static/62c33b12b334256d9ea8dfef?token=>



MCBSTM32F200_F400



Pack Installer

The screenshot shows the Pack Installer interface with two main windows:

- Devices Window:** Shows a tree view of STM32F series devices. The root node is "STM32F2 Series" with 46 devices. Other nodes include "STM32F3 Series" (90 devices), "STM32F4 Series" (205 devices), "STM32F401" (20 devices), "STM32F405" (5 devices), "STM32F407" (8 devices), "STM32F4071E" (2 devices), "STM32F4071G" (2 devices), "STM32F407VE" (1 device), "STM32F407VG" (1 device), and "STM32F407VGTx" (ARM Cortex-M4, 168 MHz, 192 kB RAM, 1 MB ROM).
- Examples Window:** Shows a list of examples categorized under "Packs". The list includes various CMSIS-RTOS, CycloneTCP, and CycloneSSL examples. A red box highlights the "HTTP Server IPv4/IPv6 (MCBSTM32F400)" example.

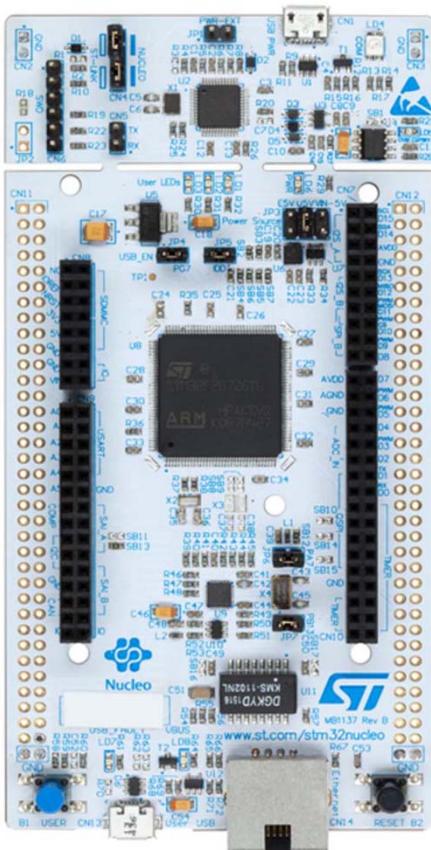
Output Window:

```

Output
Update available for Keil::ARM_Compiler (installed: 1.6.3, available: 1.7.2)
Update available for Keil::LPC1700_DFP (installed: 2.5.0, available: 2.7.1)
Update available for Keil::MDK-Middleware (installed: 7.13.0, available: 7.16.0)
Update available for Keil::STM32F4xx_DFP (installed: 2.15.0, available: 2.17.0)
Update available for Keil::STM32L4xx_DFP (installed: 2.5.0, available: 2.6.1)
Ready
  
```

WEB SERVER

file:///C:/Keil_v5/ARM/PACK/Keil/MDK-Middleware/7.13.0/Doc/Network/html/_compact_web_server_example.html
http://www.keil.com/pack/doc/mw/Network/html/_compact_web_server_example.html



Embedded Development Tools

KEIL™
An ARM® Company

Keil Embedded WEB Server Example for

ST

[[Network](#) | [System](#) | [LED](#) | [LCD](#) | [AD](#) | [Button](#) | [Language](#) | [Statistics](#)]

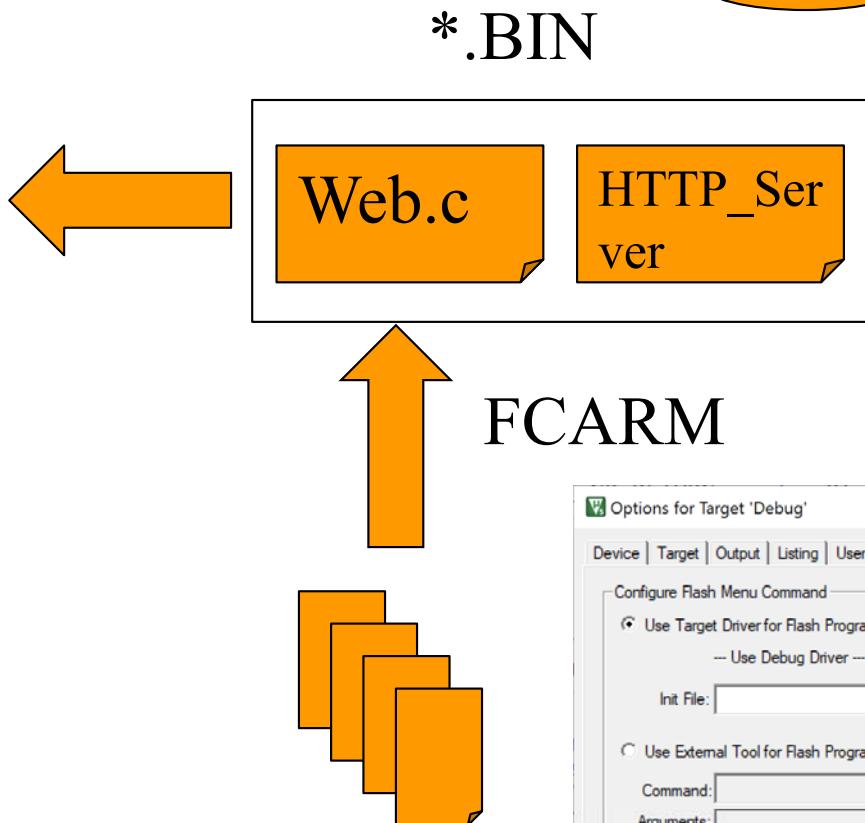
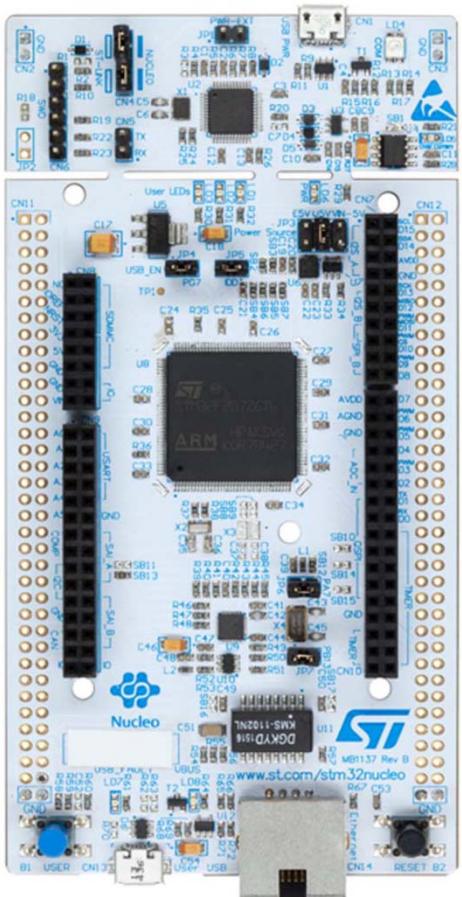
These web pages are served from a Web server,
which is part of a [Network Component](#) in [MDK-Professional](#).

Copyright © 2004-2014 KEIL - An ARM Company All rights reserved.

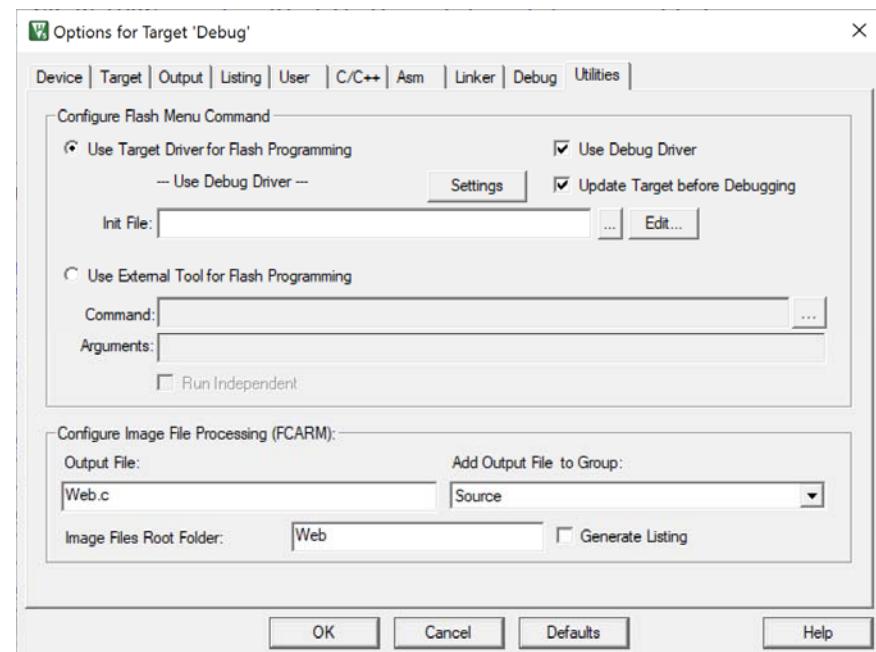
LABs DTE:
Máscara: 255.255.0.0
Gateway: 10.48.0.1
D8211 10.48.11.x
D8217 10.48.17.x

Conf. Direcciones IP privadas:
10.0.0.0 – 10.255.255.255
172.16.0.0 – 172.31.255.255
192.168.0.0 – 192.168.255.255
169.254.0.0 – 169.254.255.255

Web Server Compact



¿Qué pasa si no hay
Sistema de Ficheros?



Servidor HTTP Compacto

