Hands-On Workshop: S32K148 Using ENET

Jorge González

Automotive Applications Engineer

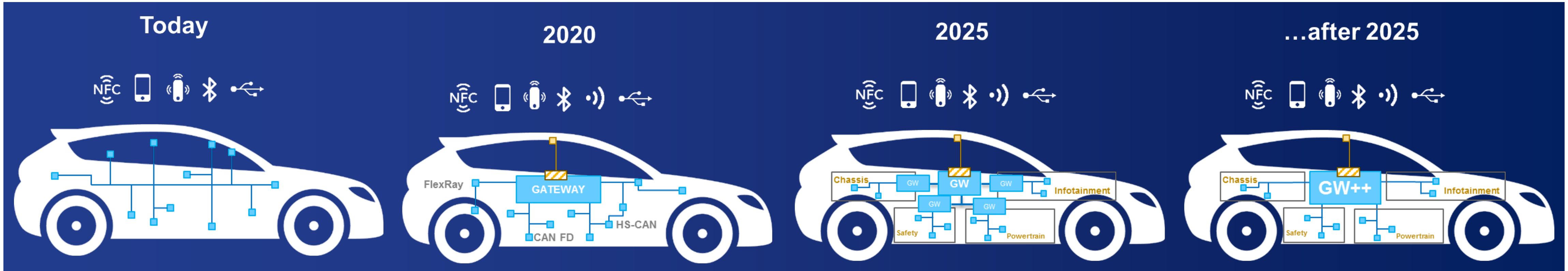
October 2019 | Session #AMF-AUT-T3821

SECURE CONNECTIONS FOR A SMARTER WORLD

Agenda

- Automotive Ethernet Overview
- S32K148 ENET Hardware and Software Enablement
- S32K148 LWIP TCP/IP + FreeRTOS
- LWIP TCP/IP Server-Client Hands-On

Ethernet Trend



Full network exposed to attacks
(LOW protection level)

Gateway limits impact
Unprotected sub-networks
(MEDIUM protection level)

Isolated systems
Domain GWs w/ controlled access
(HIGH protection level)

Multi-Domain Controller
Function clustering/virtualization
(full self-driving car)

Motivation for Ethernet

- The cable weight and cost reducing
- Synchronous and low latency - extending of AVB protocols (TSN)
- High network bandwidth – need for fast routing
- Mixed data traffic and security - distributed (Central and Domain) gateways

Infotainment

- AVB/Lip-synched Audio/Video playback
- Interface with Telematics
- High Bandwidth
- Medium QoS

Driver Assistance

- Distributed Synchronizat ion
- High Quality of Service
- Medium bandwidth
- Delivery Guarantees
- Latent Security

Gateway/Body

- Domain Partitioning
- Active Security
- Deep Packet Inspection
- Device Authentication

In-Vehicle Networking Trends and Challenges

IVN Today

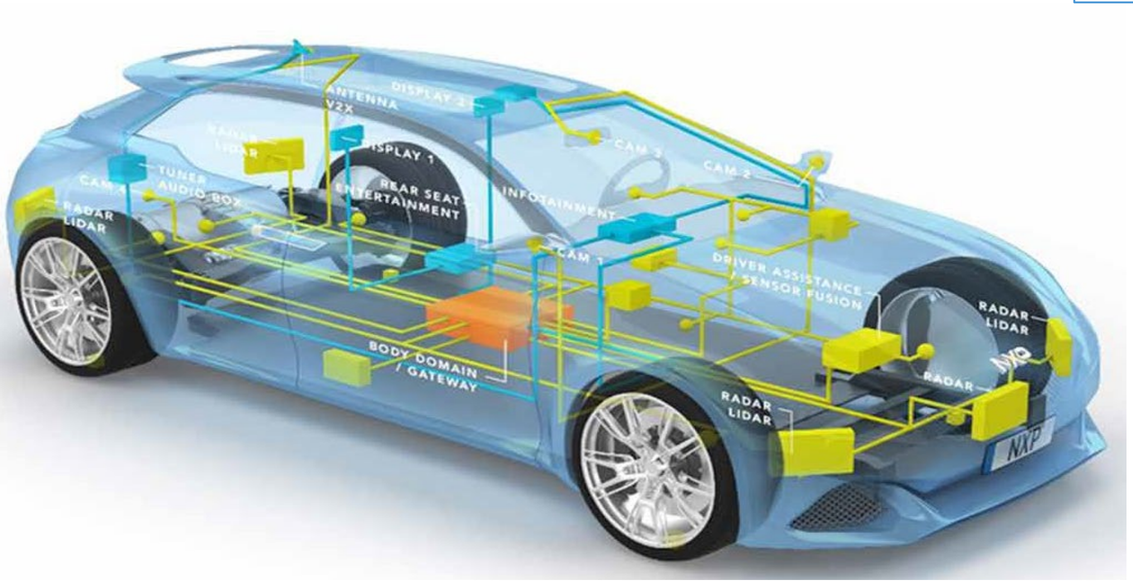
- Dominated by classic CAN
- No security
- Few gateways
- Squeezed systems (bandwidth, topology, CPU, EMC)

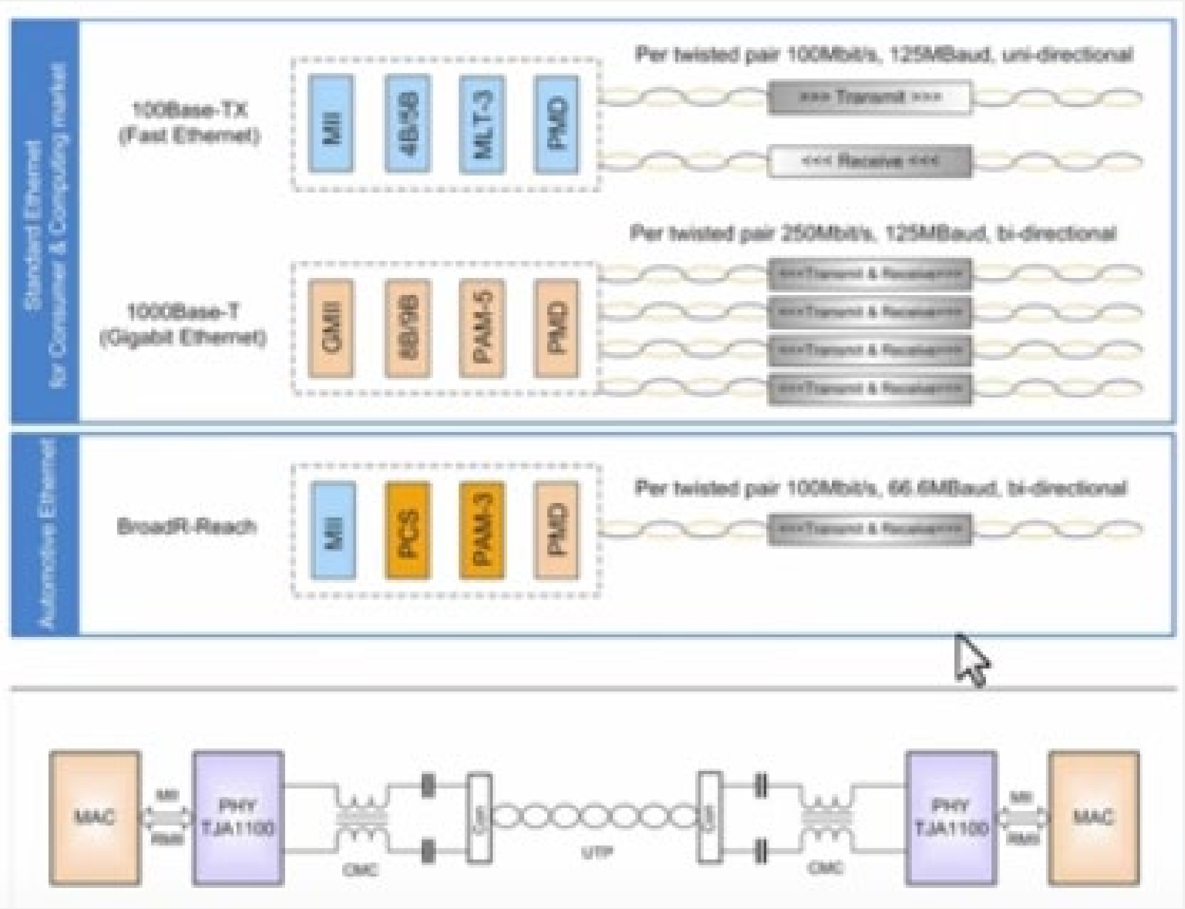
Challenges

- Major investments in network re-architecture
- Strong security not possible on CAN 2.0
- CAN FD hampered by ringing and EMC
- Lack of CAN FD and Secure MCUs
- Auto Ethernet eco-system still not mature
- Ensure the transition remains manageable

IVN Tomorrow

- CAN FD, Ethernet and more
- IDS and Crypto security
- Central and Domain gateways
- Tighter EMC specs
- Wider topology range



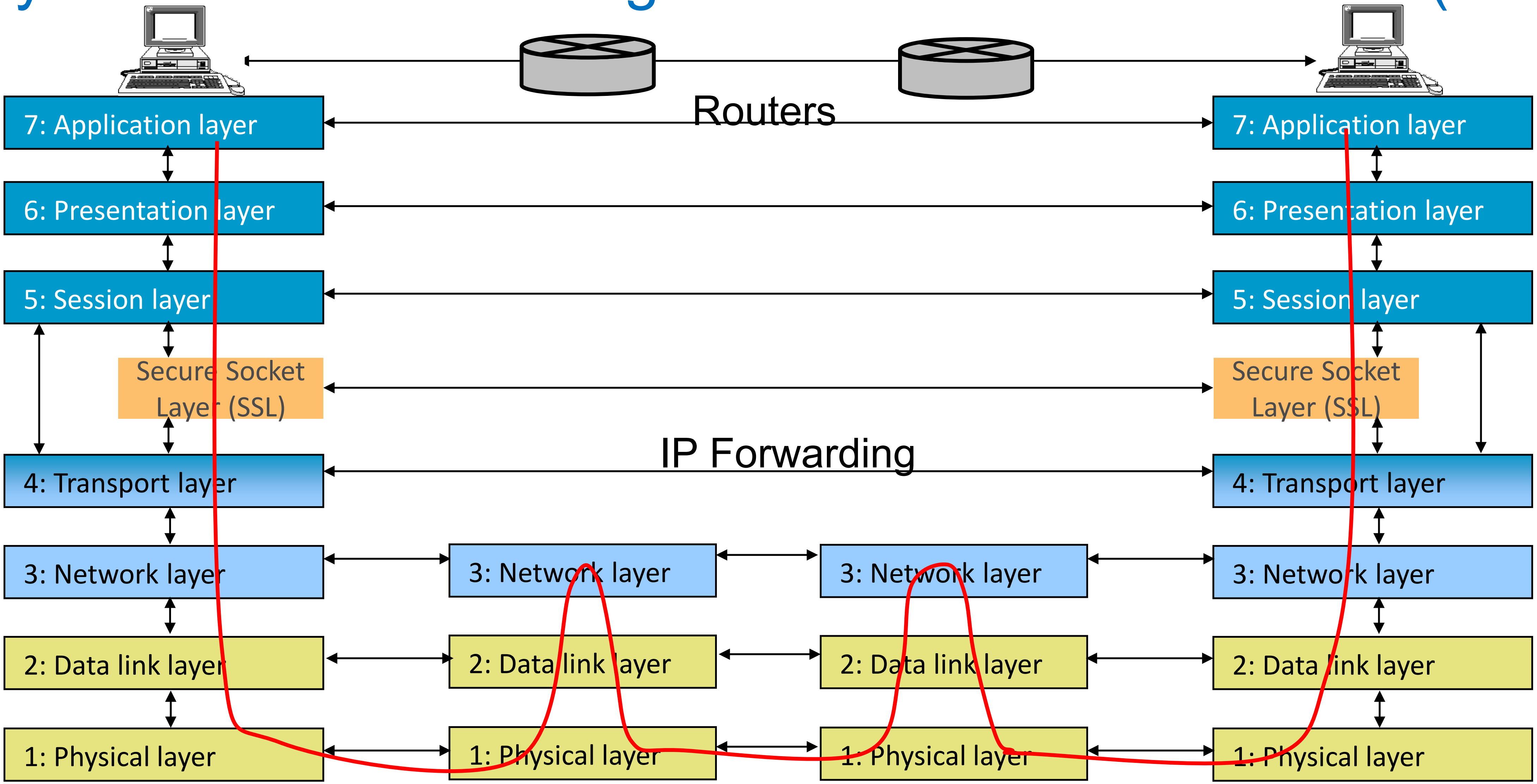


Evolution

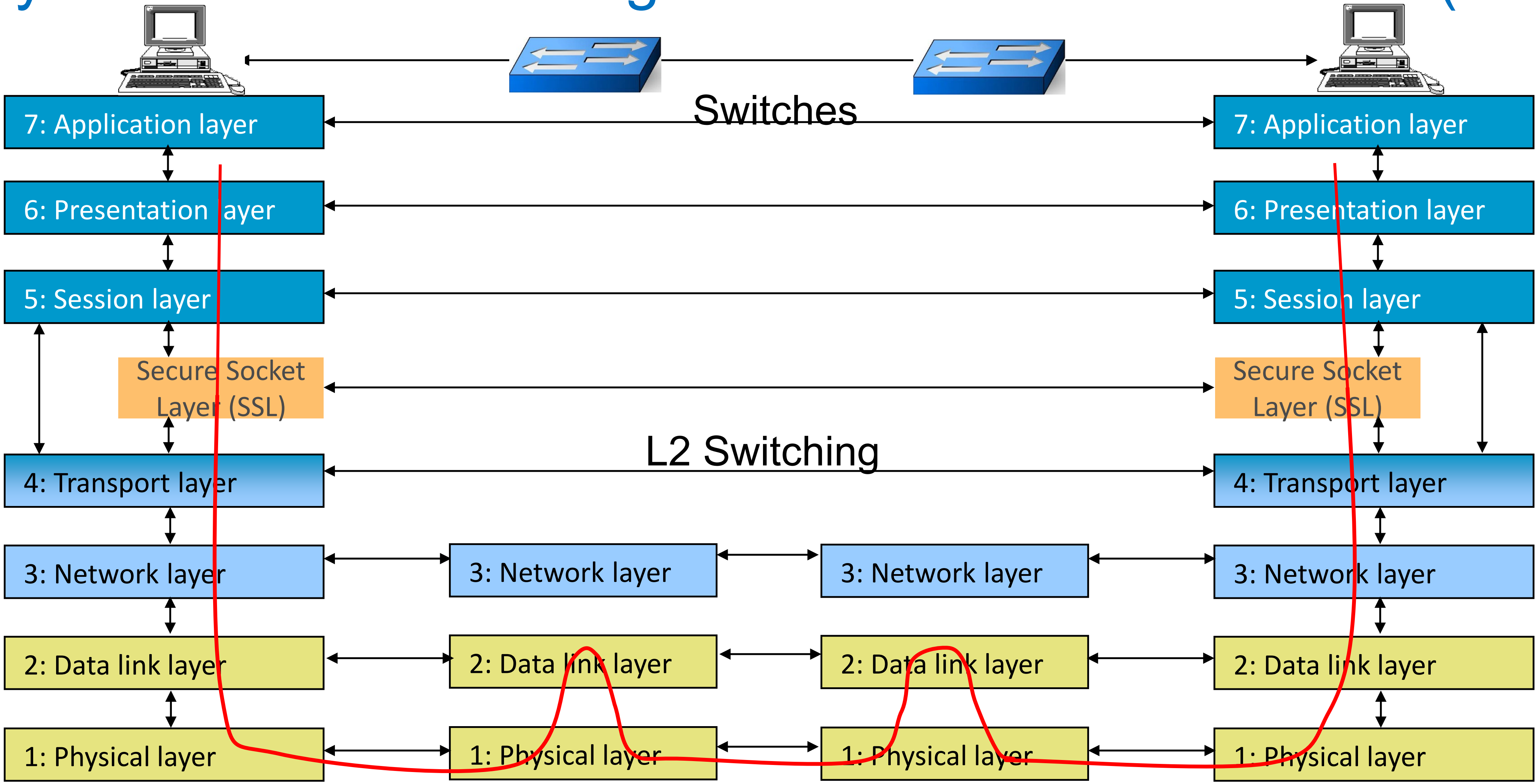
- 100Base-T1 is derived from 100BASE-TX and 1000BASE-T Ethernet
- Standarized by the OPEN alliance
- 3 Levels PAM3 Modulation
- Full-duplex communication
- Still looks like Ethernet for the Software

Enables the re-usage of all Ethernet-related Software stacks

Systems Communicating Over a Wide Area Network (WAN)



Systems Communicating Over a Local Area Network (LAN)



Automotive Ethernet / IEEE 100Base-T1 Ethernet

Standardized

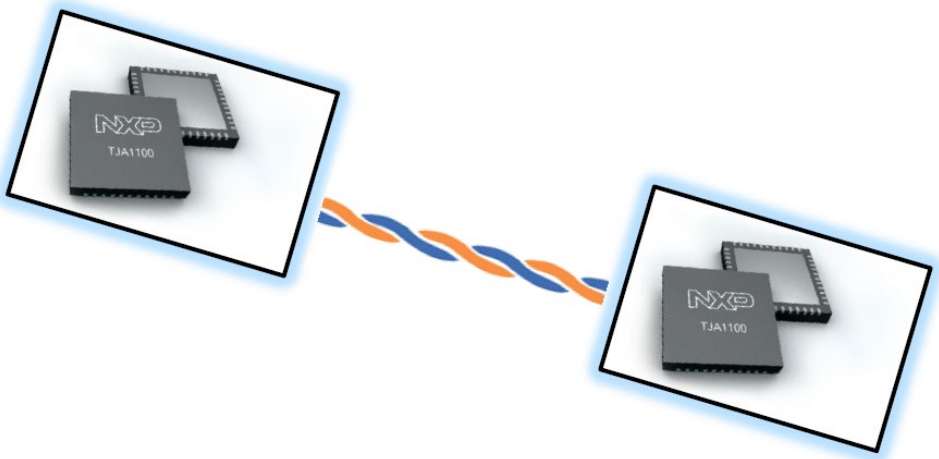
- IEEE Standard –100 BASE-T1
- Large number of suppliers and equipment
- Derived from the OPEN Alliance BroadR-Reach
- (OABR)

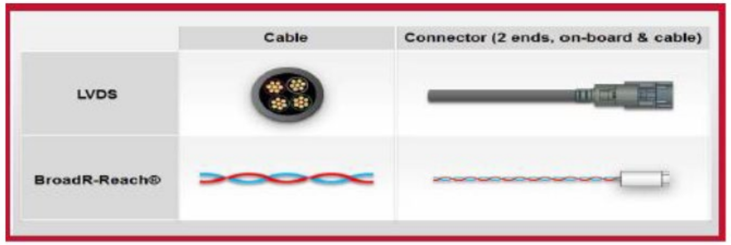
Cost Effective

- Supports Unshielded Twisted Pair up to 15m
- Similar cable as CAN and FlexRay
- Cheaper and easier than LVDS

Automotive

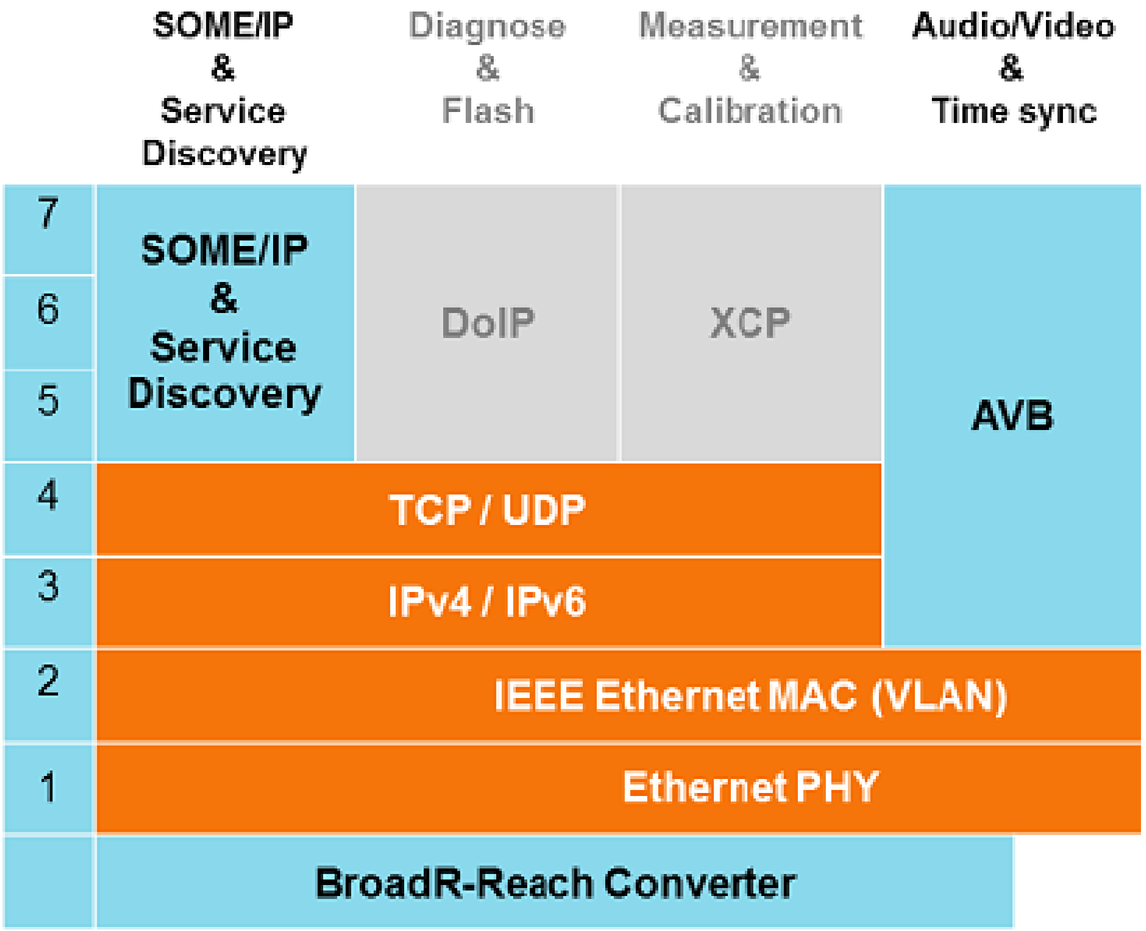
- Limited EMC emissions, within Automotive Specs
- Compatible with automotive cabling and connectors





	Cable	Connector (2 ends, on-board & cable)
LVDS		
BroadR-Reach®		

TCP/IP vs. AVB



SOME/IP & Service Discovery

Diagnose & Flash

Measurement & Calibration

Audio/Video & Time sync

7

6

5

SOME/IP & Service Discovery

DoIP

XCP

AVB

4

TCP / UDP

3

IPv4 / IPv6

2

IEEE Ethernet MAC (VLAN)

1

Ethernet PHY

BroadR-Reach Converter

Deterministic Latency

- Time critical communication

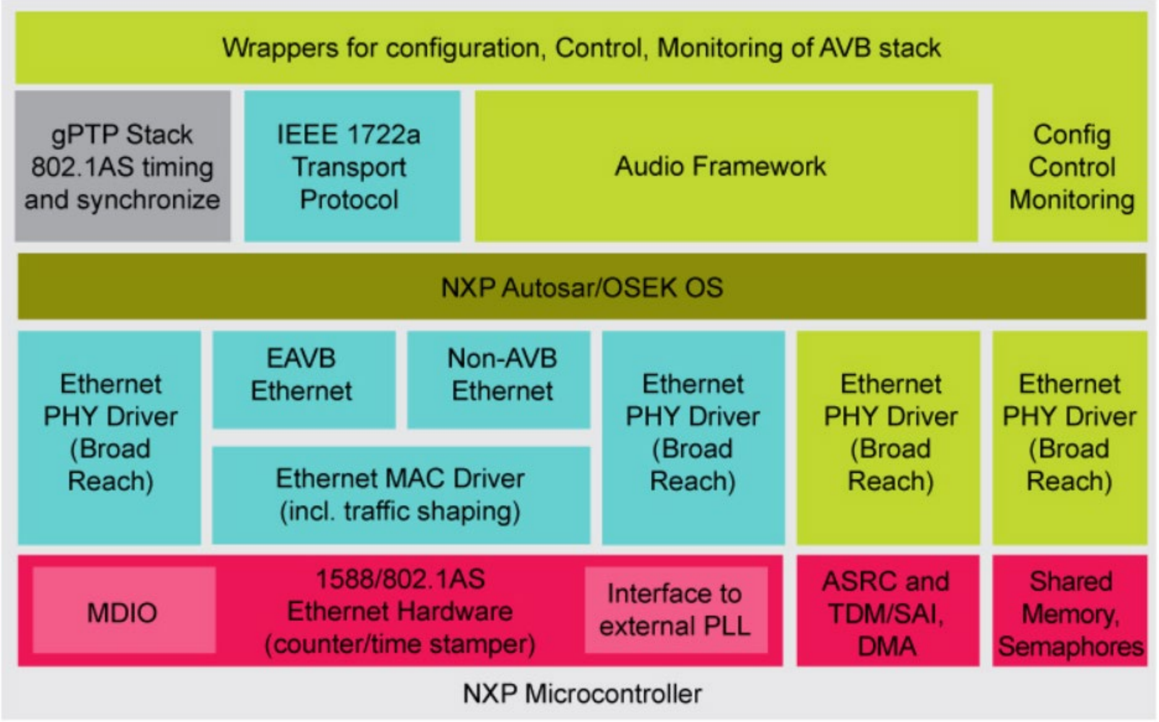
Audio Device

Microphone

Head Unit

Features IEEE 802.1 BA:

- Talker/ Listener arquitecture
- AVB Switches
- Stream Reservation
- Time Syncronization
- Traffic Reservation



Wrappers for configuration, Control, Monitoring of AVB stack

gPTP Stack 802.1AS timing and synchronize

IEEE 1722a Transport Protocol

Audio Framework

Config Control Monitoring

NXP Autosar/OSEK OS

Ethernet PHY Driver (Broad Reach)

EAVB Ethernet

Non-AVB Ethernet

Ethernet MAC Driver (incl. traffic shaping)

Ethernet PHY Driver (Broad Reach)

Ethernet PHY Driver (Broad Reach)

Ethernet PHY Driver (Broad Reach)

MDIO

1588/802.1AS Ethernet Hardware (counter/time stamper)

Interface to external PLL

ASRC and TDM/SAI, DMA

Shared Memory, Semaphores

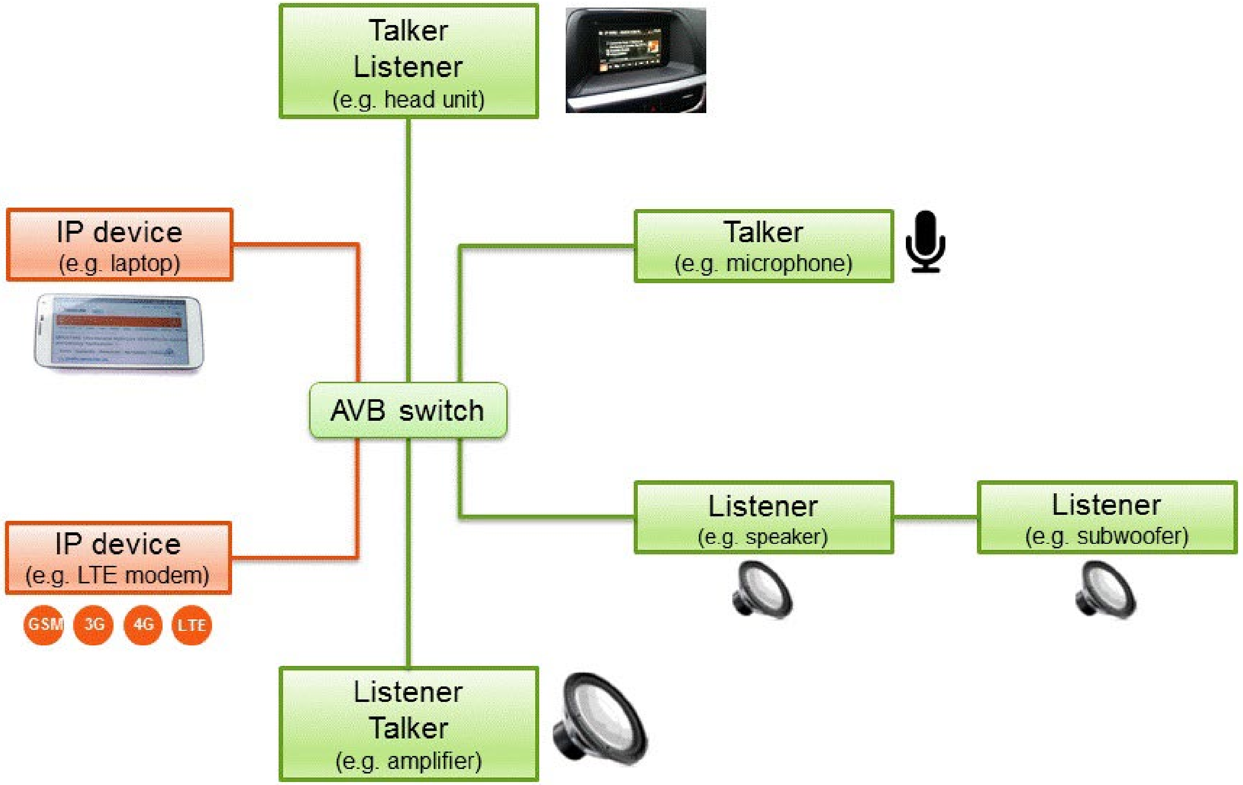
NXP Microcontroller

AVB – Automotive Ethernet Audio Video Bridging

- Syntonized audio playback over multiple audio end nodes
- Highly optimized for Resource Constrained Systems and embedded RTOS
- Audio samples processing without CPU (enhanced DMA algorithms)
- Streaming of audio data from external source or memory
- Audio data management and interconnection of stream data sources and sinks.
- Audio sample rate conversion (48kHz, 44.1kHz)
- Diagnostic data output (Ethernet/UART) and significant event callbacks
- Enhanced functions like audio output muting
- Configuration, initialization and handling of the SAI, DMA, Asynchronous Sample Rate Converter peripherals
- Media clock capture and recovery
- Abstraction through OS abstraction layer, currently available for Autosar/OSEK OS and FreeRTOS

AVB Automotive Ethernet Audio Video Bridging

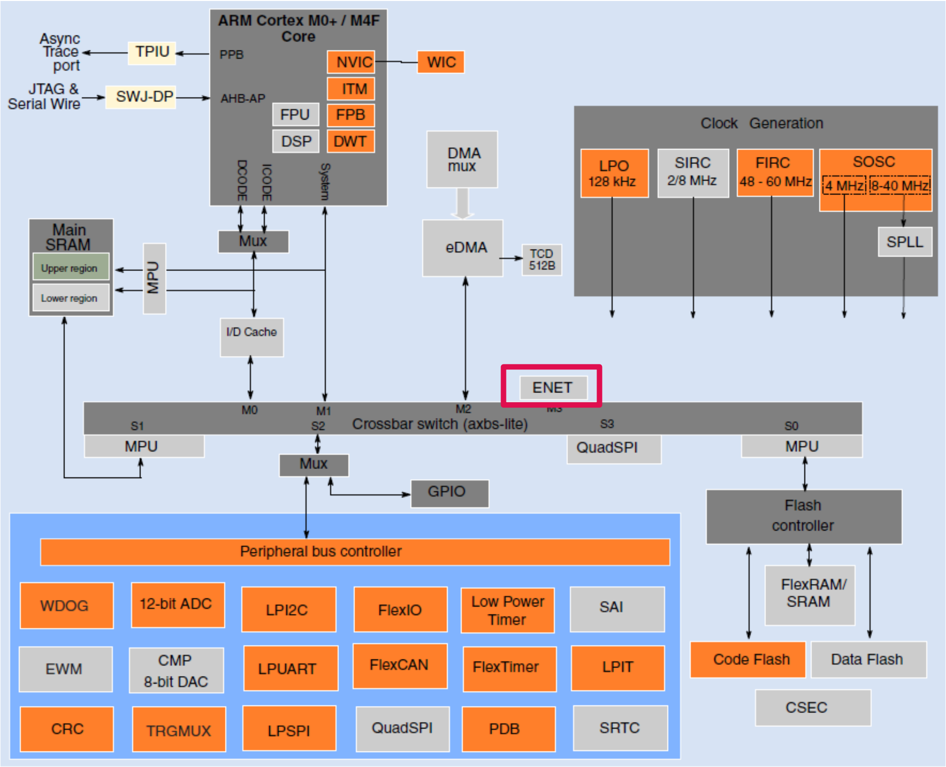
AVB Network Example

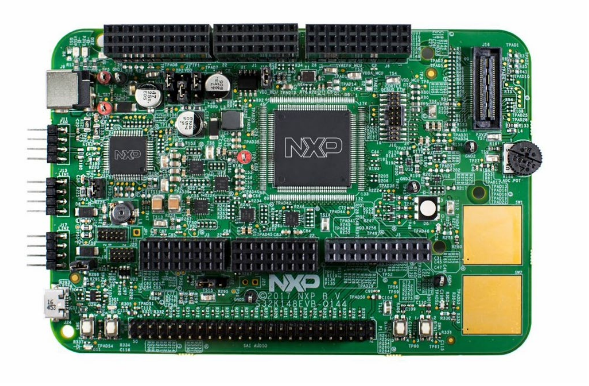


S32K148 ENET + IVN Ethernet Chipsets

S32K

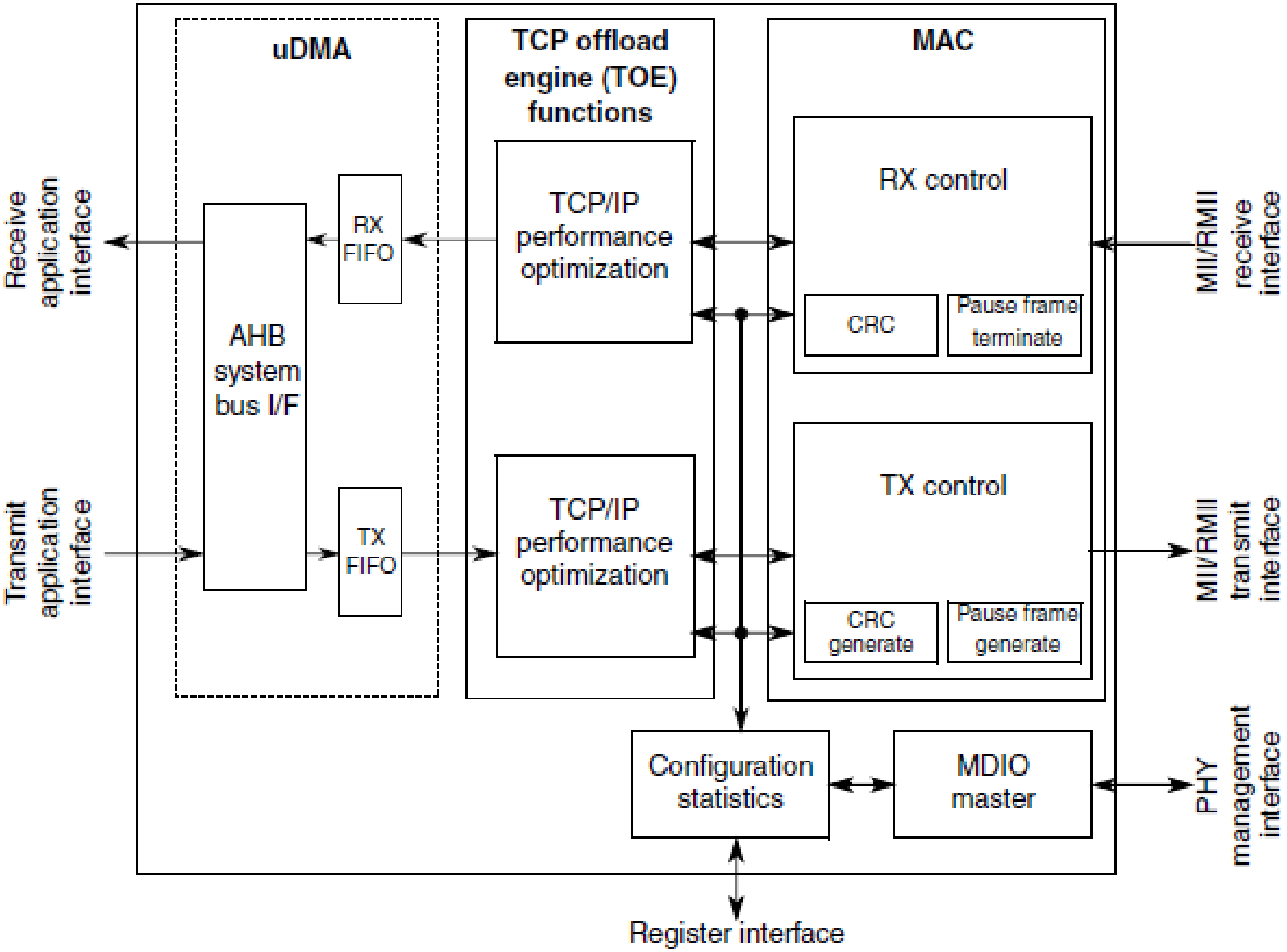
- Scalable family of AEC-Q100 qualified 32-bit Arm® Cortex®-M4F and Cortex-M0+ based MCUs targeted for general purpose automotive and high-reliability industrial applications.
- Scalability: hardware and software compatible families with multiple performance, memory and feature options.
- Integration – ISO CAN FD, CSEc hardware security, ASIL-B ISO26262 functional safety, ultra-low power performance
- Software Free production grade Software Development Kit (SDK) and S32 Design Studio IDE.
- AUTOSAR and MCAL Support, third-party ecosystem.





S32K148EVB

S32K148 ENET Module



uDMA
AHB system bus I/F
RX FIFO
TX FIFO
TCP offload engine (TOE) functions
TCP/IP performance optimization
TCP/IP performance optimization
MAC
RX control
CRC
Pause frame terminate
TX control
CRC generate
Pause frame generate
Configuration statistics
MDIO master
Receive application interface
Transmit application interface
MII/RMII receive interface
MII/RMII transmit interface
PHY management interface
Register interface

S32K148 ENET + IVN Ethernet Chipsets

TJA1101 Ethernet PHY Transceiver

- IEEE 802.3bw (100BASE-T1) compliant TJA1101 Ethernet PHY
- 100 Mbit/s transmit/receive capability over unshielded twisted pair (UTP)

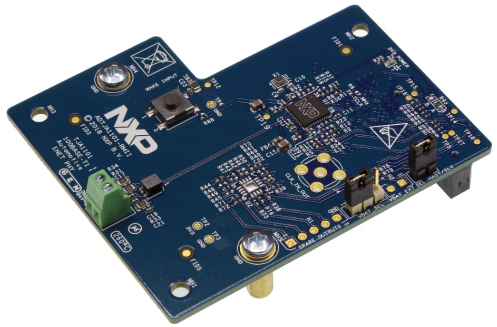
TJA1102 DUAL Ethernet PHY Transceiver

- 100BASE-T1 compliant dual-port Ethernet PHY
- 100 Mbit/s per port full duplex and supports lightweight UTP cables of up to 30 m in repeater mode

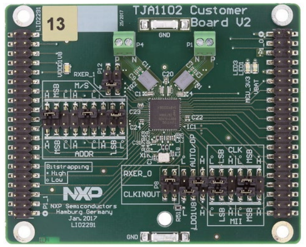
TJA1101 / TJA1102 Common Features

- AEC-Q100 Grade 1 qualified
- MII- and RMII-compliant interfaces
- Receive and forward CLK signal
- Diagnosis of cabling errors (shorts and opens)

OM14500-TJA1101
PHY Evaluation Board



ADTJA1101-RMII
Adapter Board



OM14500-TJA1102
PHY Evaluation Board

TJA110x Driver Pre-integration With S32K SDK

- Production grade driver support is pre-integrated with SDKs for NXP microcontrollers

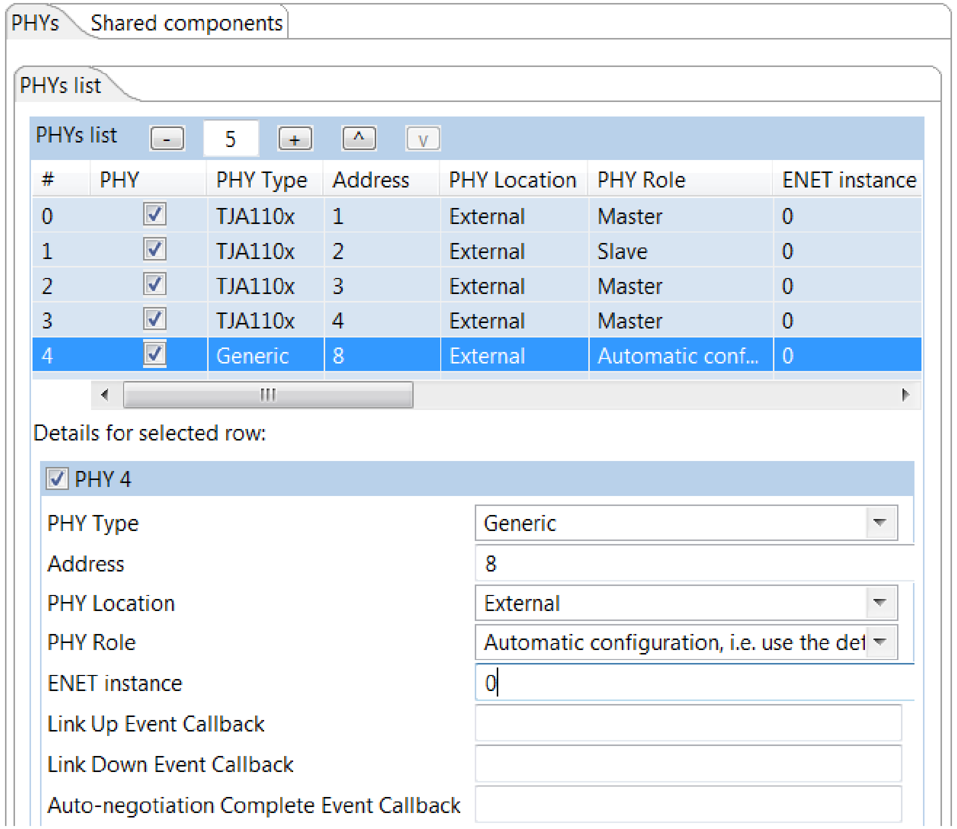
- GUI configuration support in S32 Design Studio IDE

  → see screenshot

- Supports TJA1100, TJA1101 and TJA1102(S)
  - Support for generic PHY devices using IEEE registers

- Download S32K SDK (integraded with S32 Design Studio)

GUI Driver Configuration



#	PHY	PHY Type	Address	PHY Location	PHY Role	ENET instance
0	☑	TJA110x	1	External	Master	0
1	☑	TJA110x	2	External	Slave	0
2	☑	TJA110x	3	External	Master	0
3	☑	TJA110x	4	External	Master	0
4	☑	Generic	8	External	Automatic conf...	0

Details for selected row:

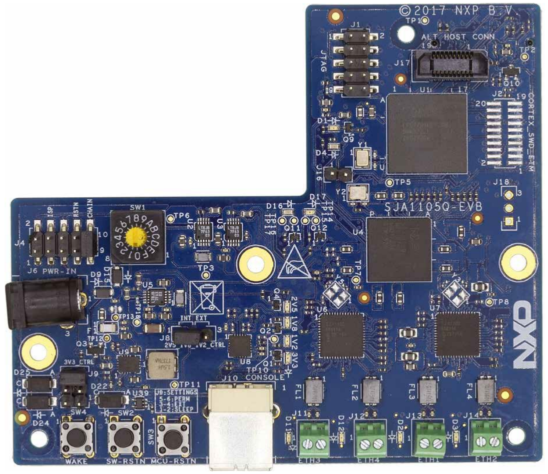
☑ PHY 4

PHY Type	Generic
Address	8
PHY Location	External
PHY Role	Automatic configuration, i.e. use the del
ENET instance	0
Link Up Event Callback	
Link Down Event Callback	
Auto-negotiation Complete Event Callback	

S32K148 ENET + IVN Ethernet Chipsets

SJA1105P / SJA1105Q / SJA1105R / SJA1105S Ethernet Switch

- Fully automotive AEC-Q100 qualified
- Automotive Grade 2 operation
- LFBGA-159 pin package (12 mm x 12 mm)
- Five ports capable of 10/100/1000 Mbit/s data rate including MII/RMII/RGMII and SGMII interfaces
- Non-blocking full gigabit switching capability
- Support for AVB and TSN/802.1Qbv scheduled traffic standard
- MAC address filtering and black/white listing



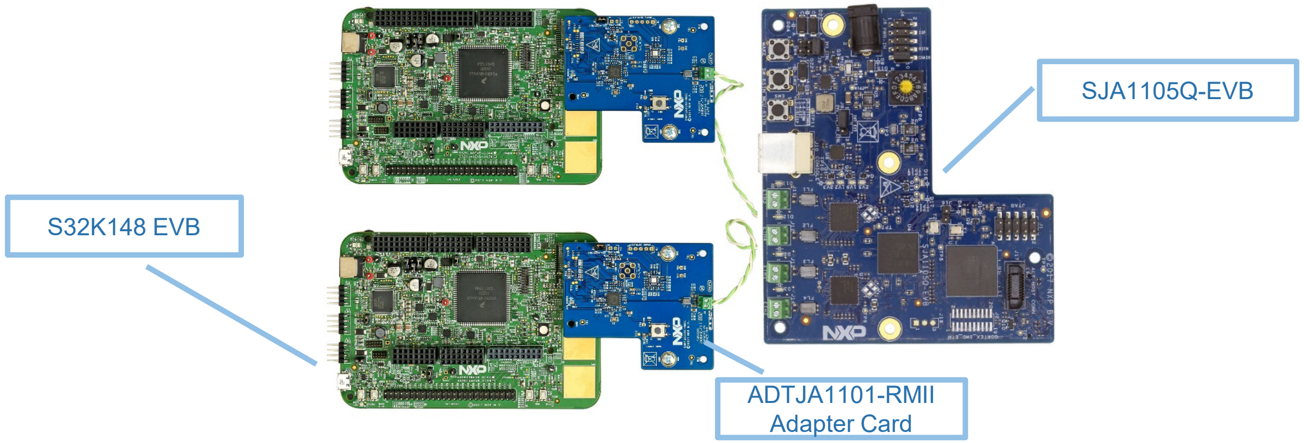
SJA1105Q-EVB
Ethernet Switch and PHY Evaluation Board

SJA1105SMBEVM
Gateway Prototyping Platform

S32K148 ENET + IVN Ethernet Chipsets

Connected System Examples

#1

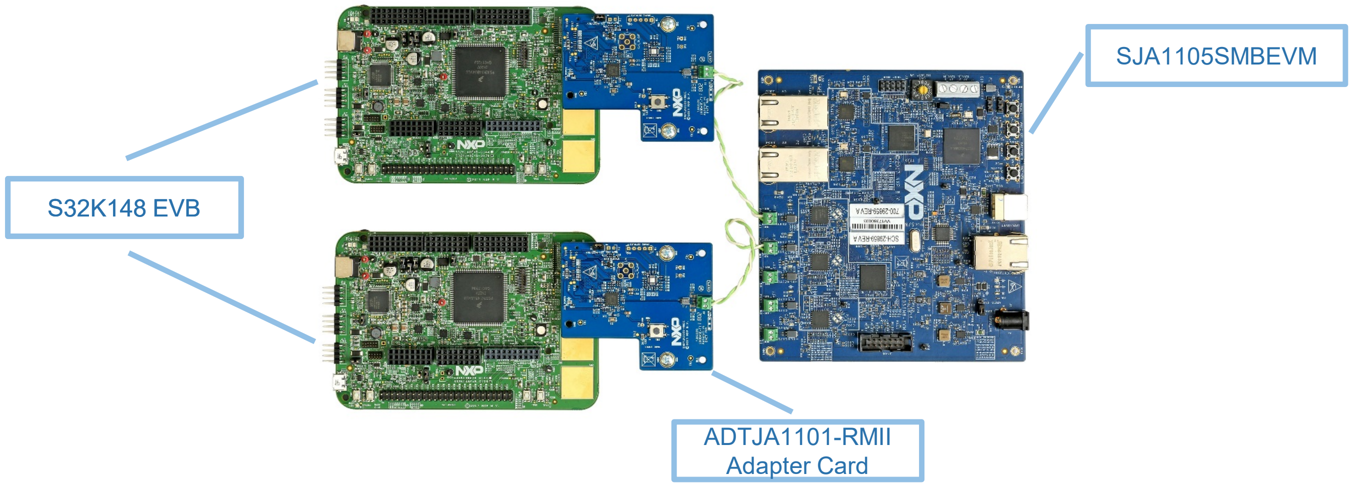


SJA1105Q-EVB

S32K148 EVB

ADTJA1101-RMII Adapter Card

#2



SJA1105SMBEVM

S32K148 EVB

ADTJA1101-RMII Adapter Card

S32K148 LWIP TCP/IP + FreeRTOS: FreeRTOS

- A real time operating system for MCUs
  - RTOS: Guarantees delivery within a defined time frame. Deterministic.

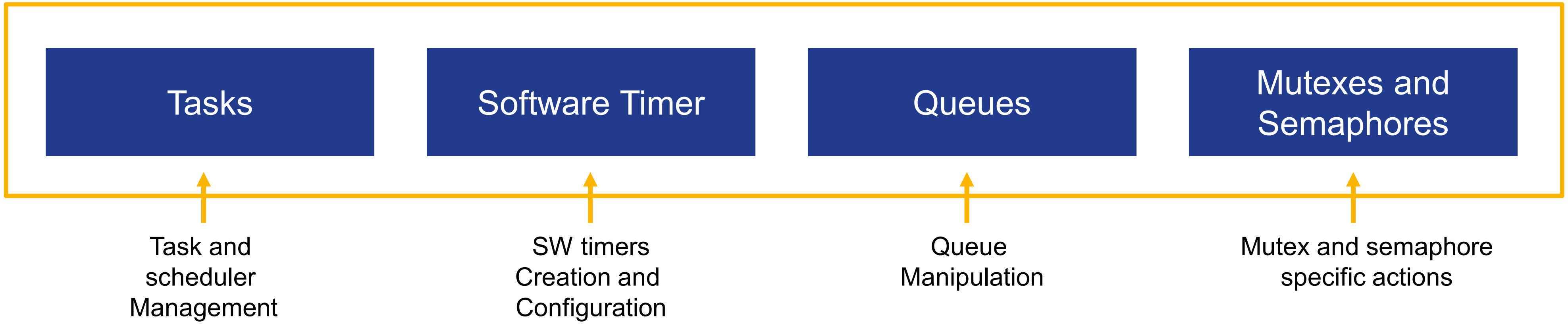
- FreeRTOS is designed to be small and simple

- FreeRTOS provides methods for:
  - Multiple threads (or ‘tasks’)
  - Scheduling
  - Semaphores
  - Mutexes
  - Software Timers
  - Tick-less mode for low power

S32K148 LWIP TCP/IP + FreeRTOS: FreeRTOS

- Very simple – core has 3 C files (`tasks.c`, `queue.c`, and `list.c`), suitable for small embedded devices
- Lightweight – 6-12K RAM
- Free of charge – no payments needed, not even for production
- Open source – lots of online support (tutorials, examples, demos)
- Ported on NXP S32K14x devices – delivered with the SDK

Go faster to market

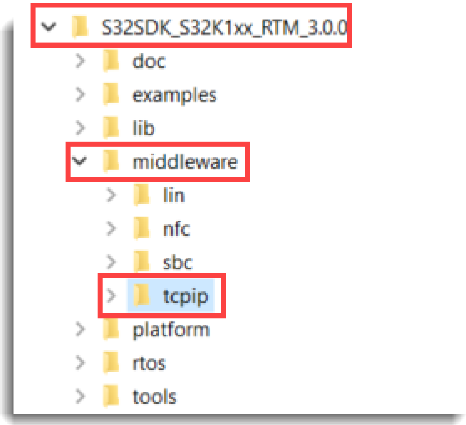

Tasks
Software Timer
Queues
Mutexes and Semaphores
Task and scheduler Management
SW timers Creation and Configuration
Queue Manipulation
Mutex and semaphore specific actions

S32K148 LWIP TCP/IP + FreeRTOS: LWIP

- Open source TCP/IP stack designed for embedded systems
- Features:

Internet layer	**Transport layer**	**Application layer**
IP: IPv4 & IPv6	UDP	DNS
ICMP	TCP	SNMP
IGMP	Raw sockets	DHCP client

- Ported on NXP ***S32K148*** (delivered with the SDK)



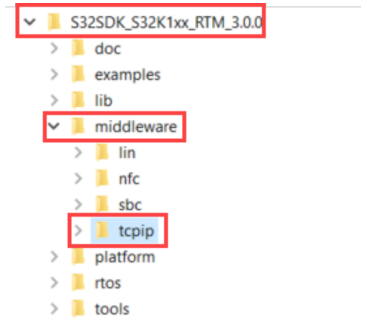
```
S32SDK_S32K1xx_RTM_3.0.0
  doc
  examples
  lib
  middleware
    lin
    nfc
    sbc
    tcpip
  platform
  rtos
  tools
```

S32K148 LWIP TCP/IP + FreeRTOS: LWIP

- Open source TCP/IP stack designed for embedded systems
- Features:

Internet layer	Transport layer	Application layer
IP: IPv4 & IPv6	UDP	DNS
ICMP	TCP	SNMP
IGMP	Raw sockets	DHCP client

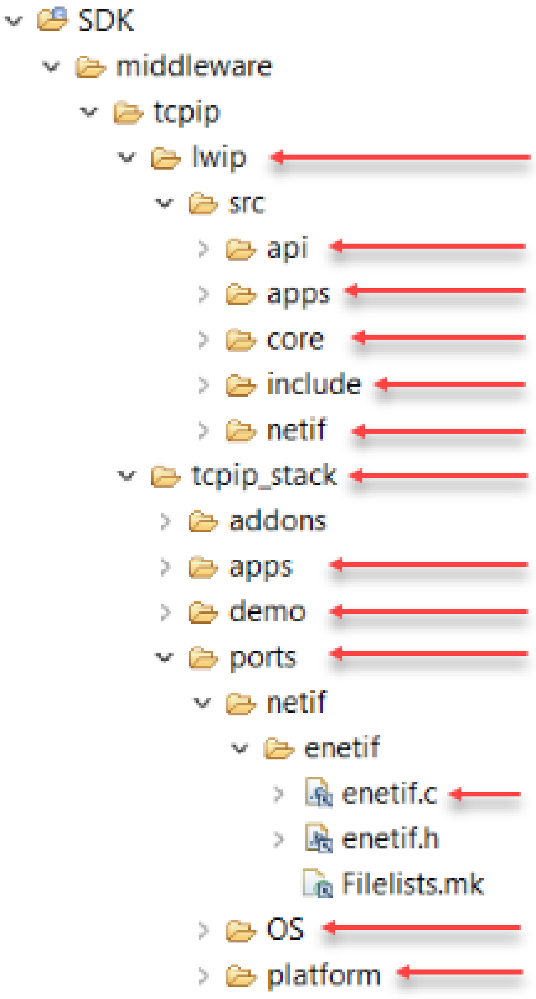
- Ported on NXP S32K148 (delivered with the SDK)



```
S32SDK_S32K1xx_RTM_3.0.0
  doc
  examples
  lib
  middleware
    lin
    nfc
    sbc
    tcpip
  platform
  rtos
  tools
```

S32K148 LWIP TCP/IP + FreeRTOS: LWIP

Stack source code structure

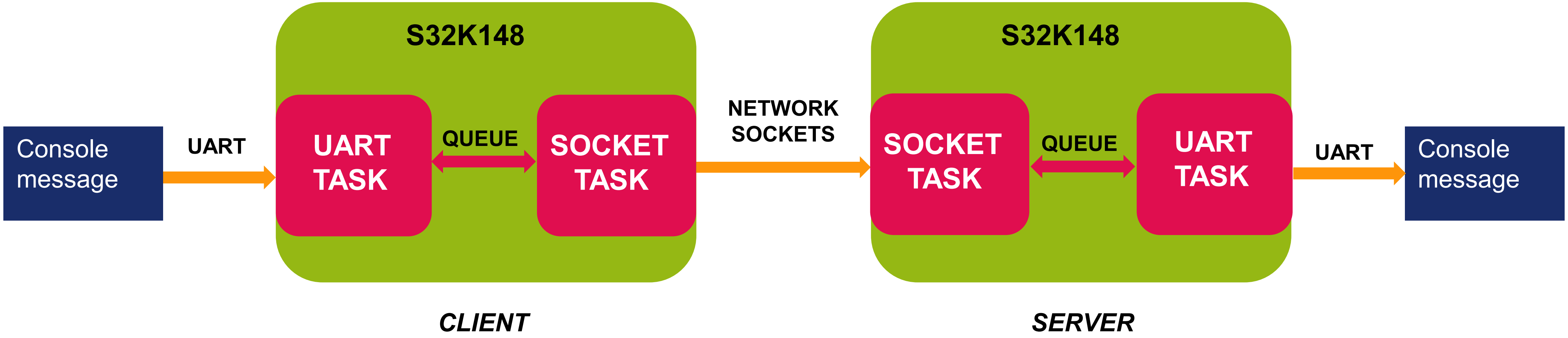


```
SDK
  middleware
    tcpip
      lwip                 ← LWIP stack source
        src
          api              ← Netconn API, Socket API and TCP/IP thread
          apps             ← LWIP stack application layer source files (DNS, MQTT, SNMP, TFTP, …)
          core             ← Core code: DHCP, TCP, UDP and support code (mem, netif, …)
          include          ← All headers and includes
          netif            ← PPP and Ethernet Network Interface low level driver skeletons (example driver files)
      tcpip_stack          ← Adaption layer to S32 SDK for S32K148 and FreeRTOS of LWIP stack
        addons
        apps               ← Example LWIP applications
        demo               ← Contains the "test.c" source file which initializes LWIP stack and the different applications
        ports              ← Adaption files for compiler, RTOS and interface code to low level driver (ENET driver for S32K148)
          netif
            enetif
              enetif.c     ← Implements the low level calls to the S32 SDK ENET driver for handling Tx/Rx of data
              enetif.h
              Filelists.mk
          OS               ← RTOS abstraction files to handle common services via FreeRTOS (thread creation, mutex, semaphore, …)
          platform         ← Compiler specific adaption code
```

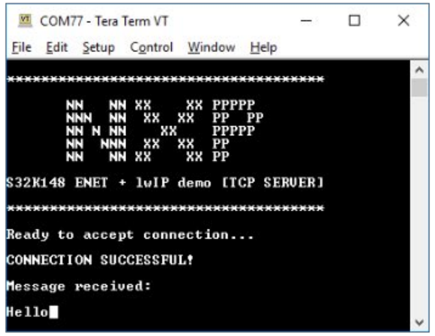
LWIP TCP/IP Server-Client Hands-On

LWIP TCP/IP Server-Client Hands-On

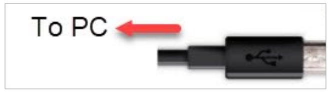
- The purpose of this lab is to establish a communication between two S32K148 boards using network sockets
- Messages are read from the console by the client, then sent to the server; upon receival, the server displays the message to the console
- UART and sockets are handled by different OS tasks, communicating using queues

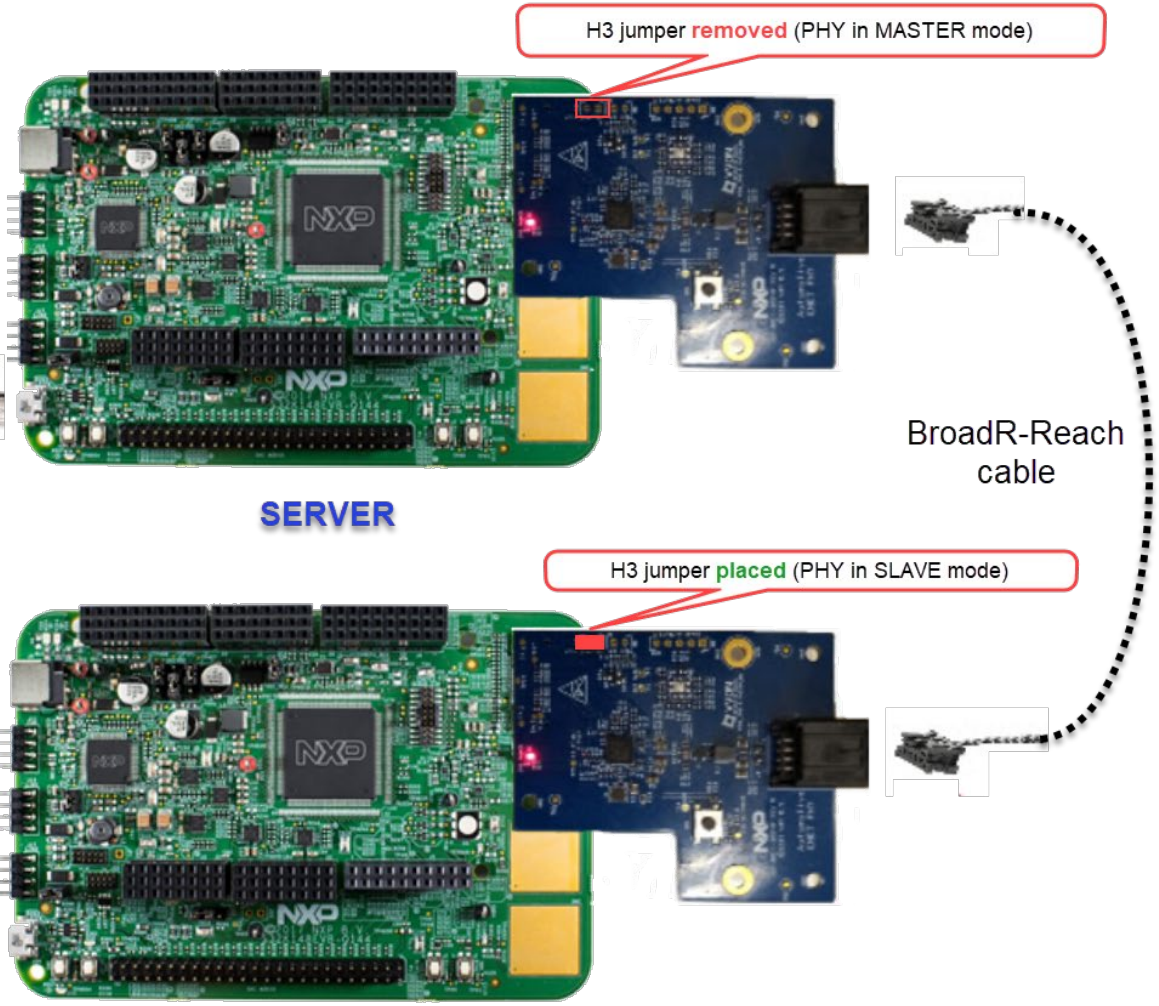


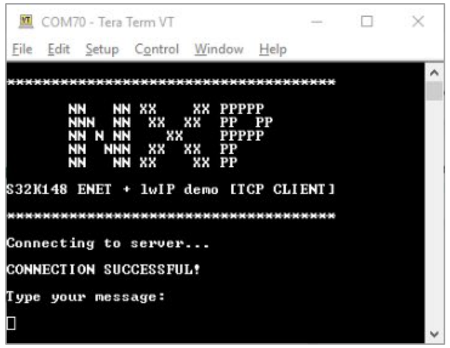
LWIP TCP/IP Server-Client Hands-On



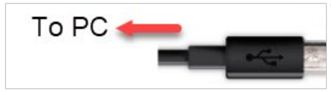
```
*************************************

       NN   NN XX    XX PPPPP
       NNN  NN  XX  XX  PP  PP
       NN N NN    XX    PPPPP
       NN  NNN  XX  XX  PP
       NN   NN XX    XX PP

S32K148 ENET + lwIP demo [TCP SERVER]

*************************************

Ready to accept connection...

CONNECTION SUCCESSFUL!

Message received:

Hello
```




H3 jumper removed (PHY in MASTER mode)
SERVER
BroadR-Reach cable
H3 jumper placed (PHY in SLAVE mode)
CLIENT



```
*************************************

       NN   NN XX    XX PPPPP
       NNN  NN  XX  XX  PP  PP
       NN N NN    XX    PPPPP
       NN  NNN  XX  XX  PP
       NN   NN XX    XX PP

S32K148 ENET + lwIP demo [TCP CLIENT]

*************************************

Connecting to server...

CONNECTION SUCCESSFUL!

Type your message:

```

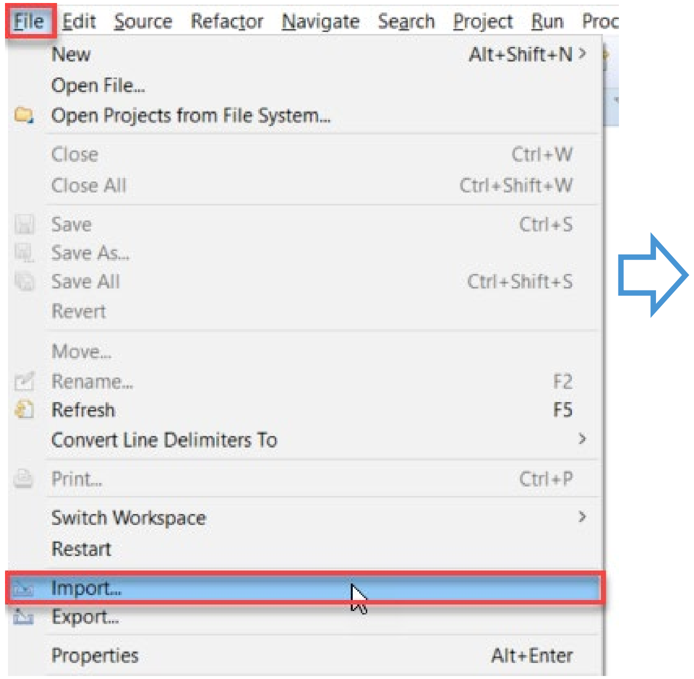


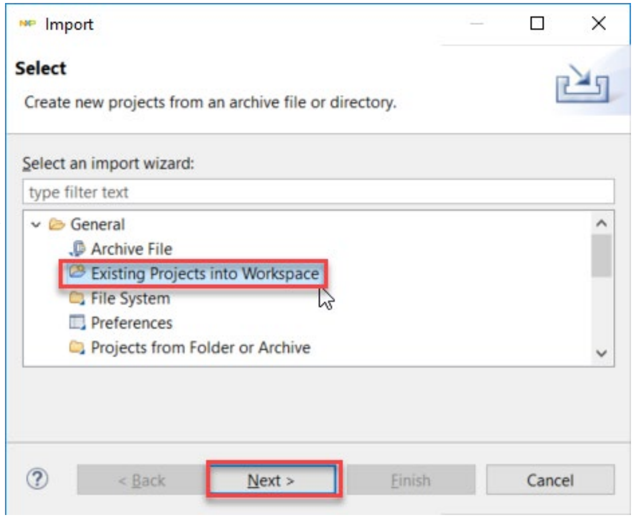
S32K148 TCP/IP + FreeRTOS: Import Project

- Go to *File -> Import -> General -> Existing Projects into Workspace.*
- Import project ***lwip_s32k148_HandsOn_Server***

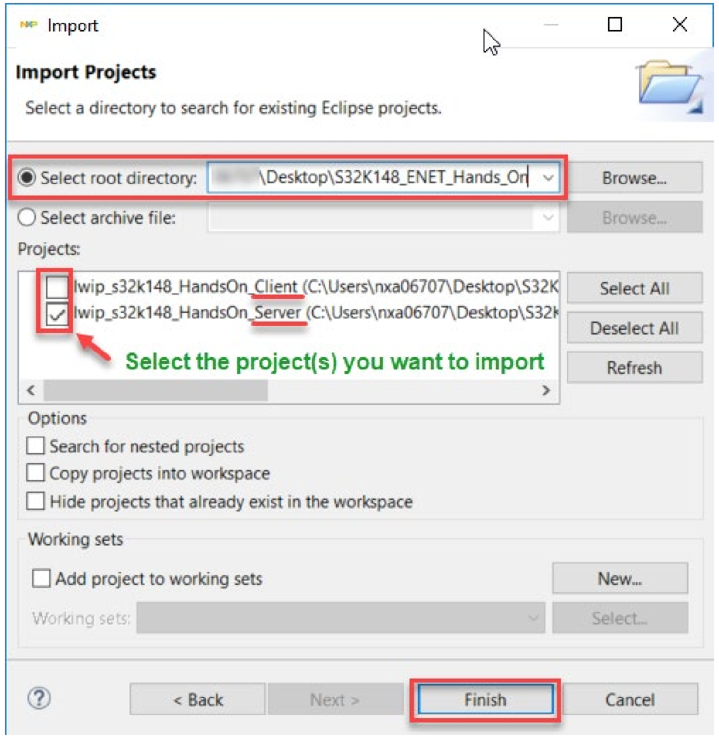
or

- Import project ***lwip_s32k148_HandsOn_Client***





Import
Select
Create new projects from an archive file or directory.
Select an import wizard:
type filter text
General
Archive File
Existing Projects into Workspace
File System
Preferences
Projects from Folder or Archive
< Back
Next >
Finish
Cancel


Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: [redacted]\Desktop\S32K148_ENET_Hands_On

○ Select archive file:

Projects:

☐ lwip_s32k148_HandsOn_Client (C:\Users\nxa06707\Desktop\S32K

☑ lwip_s32k148_HandsOn_Server (C:\Users\nxa06707\Desktop\S32K

Select the project(s) you want to import

Options

☐ Search for nested projects

☐ Copy projects into workspace

☐ Hide projects that already exist in the workspace

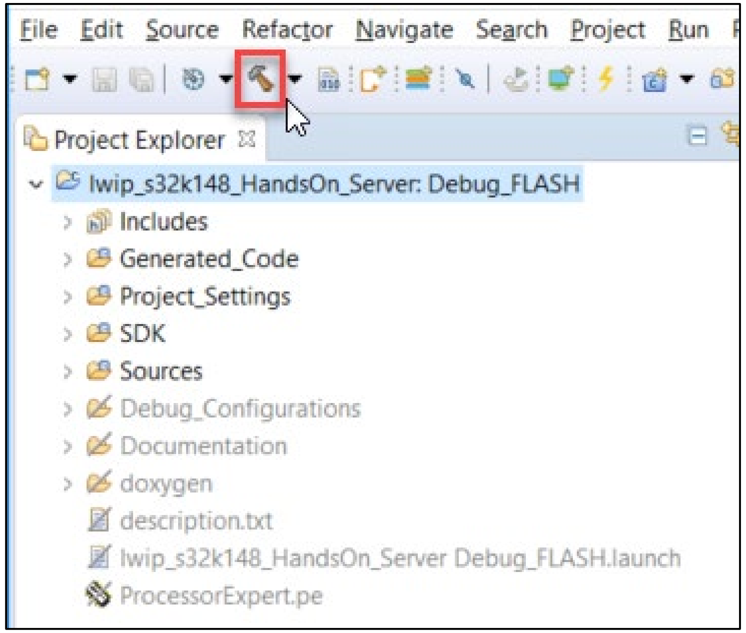
Working sets

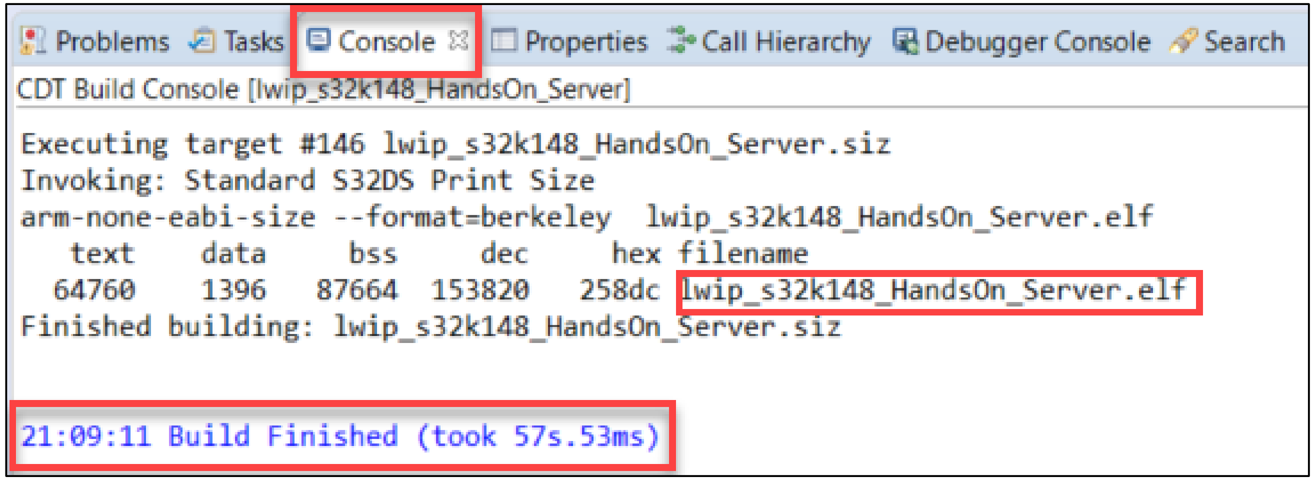
☐ Add project to working sets

Working sets:

S32K148 TCP/IP + FreeRTOS: Build

- Click the ***Build*** button (hammer icon) to build the project.
- Check the builder console for any errors.



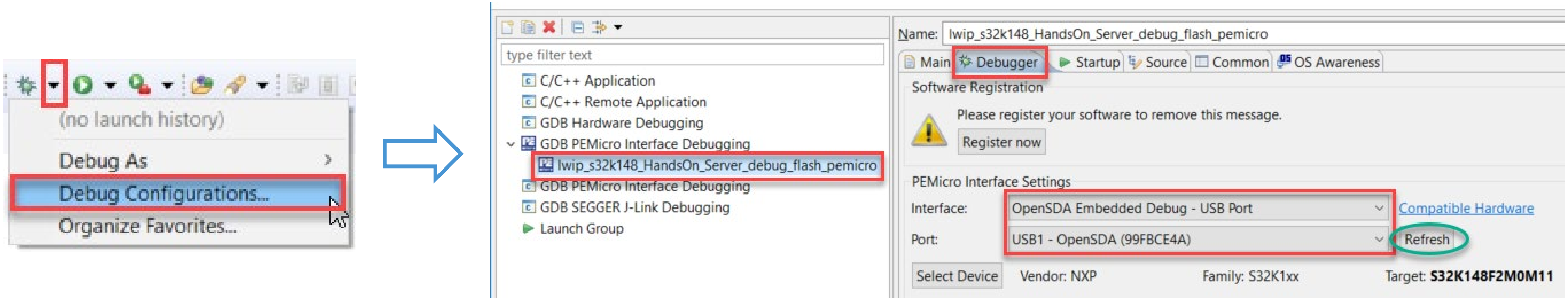


CDT Build Console [lwip_s32k148_HandsOn_Server]

```
Executing target #146 lwip_s32k148_HandsOn_Server.siz
Invoking: Standard S32DS Print Size
arm-none-eabi-size --format=berkeley  lwip_s32k148_HandsOn_Server.elf
   text    data     bss     dec     hex filename
  64760    1396   87664  153820   258dc lwip_s32k148_HandsOn_Server.elf
Finished building: lwip_s32k148_HandsOn_Server.siz


21:09:11 Build Finished (took 57s.53ms)
```

S32K148 TCP/IP + FreeRTOS: Debug

- Open the debug configurations window (click on the small arrow by the side of the bug icon or go to tab *Run -> Debug Configurations*).
- Select the P&E Micro debug configuration and go to Debugger tab.
- If the S32K148EVB is already connected through USB, the OpenSDA interface is detected. Otherwise connect the board and click on ***Refresh***.



Name: lwip_s32k148_HandsOn_Server_debug_flash_pemicro

Please register your software to remove this message.

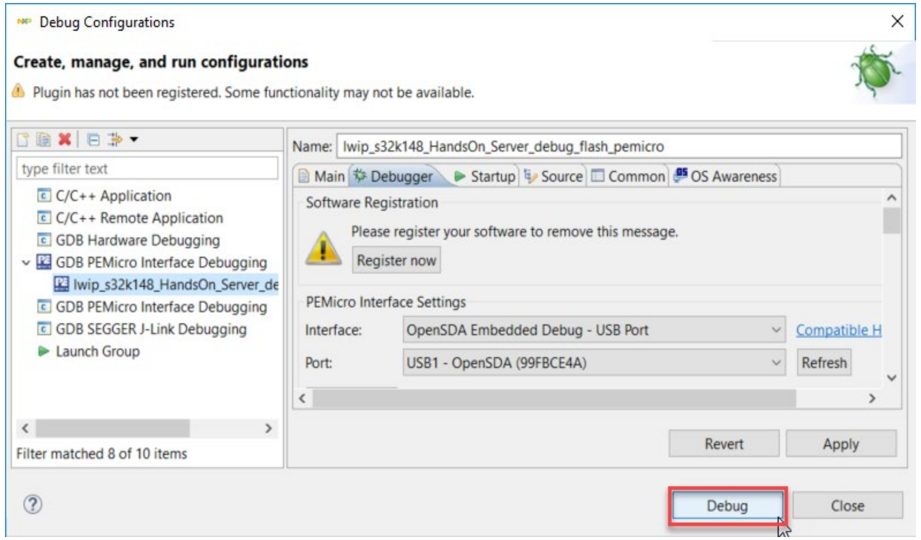
Interface: OpenSDA Embedded Debug - USB Port

Port: USB1 - OpenSDA (99FBCE4A)

Vendor: NXP Family: S32K1xx Target: **S32K148F2M0M11**

S32K148 TCP/IP + FreeRTOS: Debug

- Click on ***Debug*** to download the binary to the target.
  - If you see the window to confirm Perspective switch, just click on *Yes*.



Debug Configurations

Create, manage, and run configurations

Plugin has not been registered. Some functionality may not be available.

type filter text
C/C++ Application
C/C++ Remote Application
GDB Hardware Debugging
GDB PEMicro Interface Debugging
lwip_s32k148_HandsOn_Server_de
GDB PEMicro Interface Debugging
GDB SEGGER J-Link Debugging
Launch Group

Filter matched 8 of 10 items

Name: lwip_s32k148_HandsOn_Server_debug_flash_pemicro

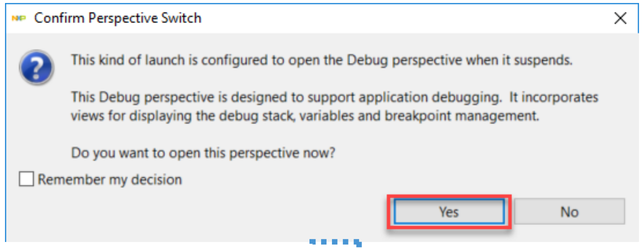
Software Registration

Please register your software to remove this message.

PEMicro Interface Settings

Interface: OpenSDA Embedded Debug - USB Port

Port: USB1 - OpenSDA (99FBCE4A)



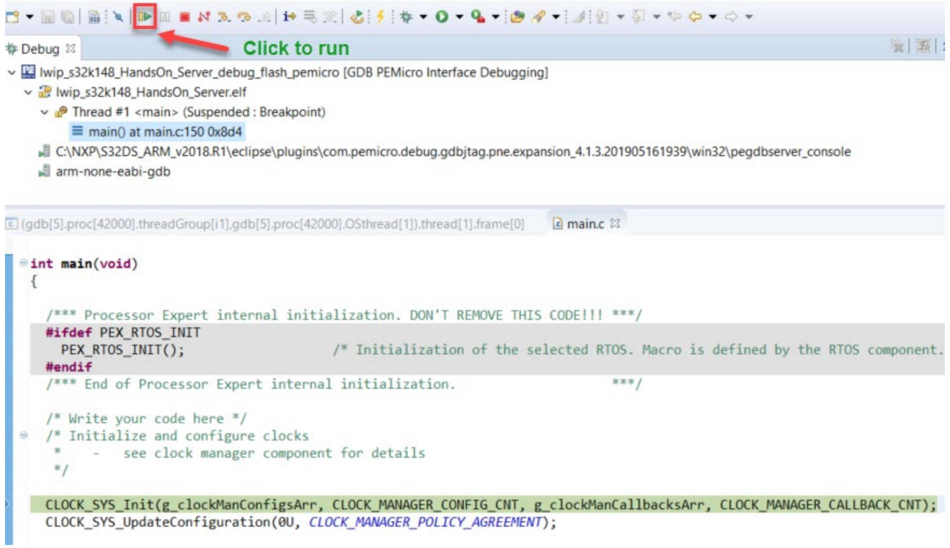
Confirm Perspective Switch

This kind of launch is configured to open the Debug perspective when it suspends.

This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.

Do you want to open this perspective now?

☐ Remember my decision



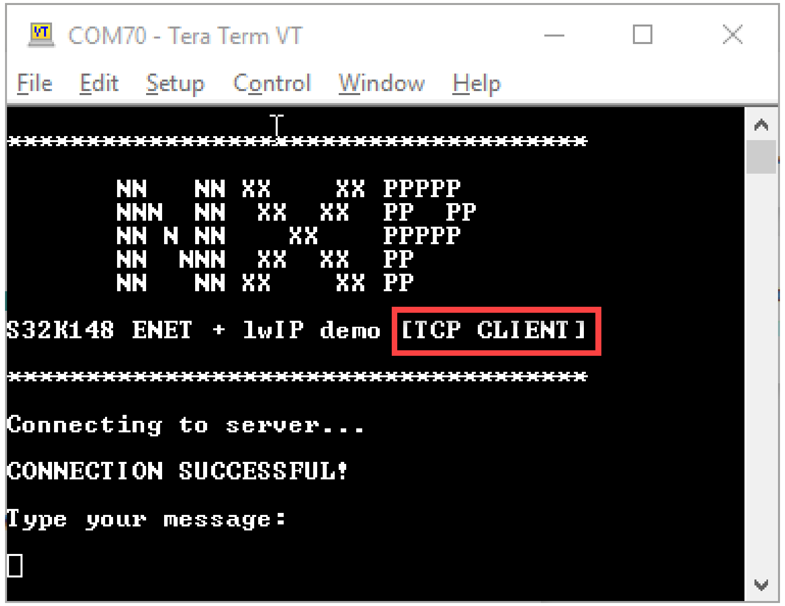
```
lwip_s32k148_HandsOn_Server_debug_flash_pemicro [GDB PEMicro Interface Debugging]
  lwip_s32k148_HandsOn_Server.elf
    Thread #1 <main> (Suspended : Breakpoint)
      main() at main.c:150 0x8d4
  C:\NXP\S32DS_ARM_v2018.R1\eclipse\plugins\com.pemicro.debug.gdbjtag.pne.expansion_4.1.3.201905161939\win32\pegdbserver_console
  arm-none-eabi-gdb
```

(gdb[5].proc[42000].threadGroup[i1],gdb[5].proc[42000].OSthread[1]).thread[1].frame[0]
main.c

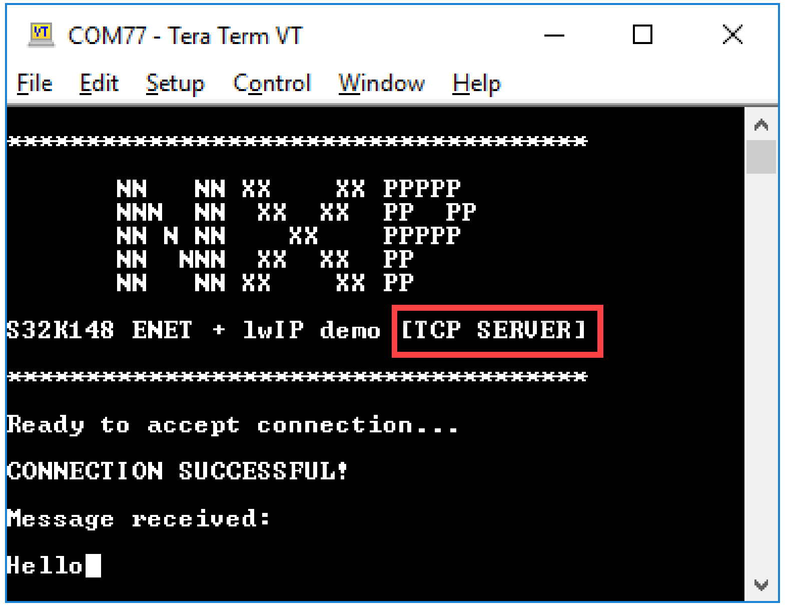
```
int main(void)
{

  /*** Processor Expert internal initialization. DON'T REMOVE THIS CODE!!! ***/
  #ifdef PEX_RTOS_INIT
    PEX_RTOS_INIT();                 /* Initialization of the selected RTOS. Macro is defined by the RTOS component.
  #endif
  /*** End of Processor Expert internal initialization.                    ***/

  /* Write your code here */
  /* Initialize and configure clocks
   *    -   see clock manager component for details
   */

  CLOCK_SYS_Init(g_clockManConfigsArr, CLOCK_MANAGER_CONFIG_CNT, g_clockManCallbacksArr, CLOCK_MANAGER_CALLBACK_CNT);
  CLOCK_SYS_UpdateConfiguration(0U, CLOCK_MANAGER_POLICY_AGREEMENT);
```

S32K148 TCP/IP + FreeRTOS: Test Example

- Open a terminal interface (e.g. Tera Term) and select the OpenSDA COM Port.
  Settings: Baudrate – 115200 / Bitcount – 8 / No parity / 1 stop bit.
- Make sure that ethernet daughter cards from server and client boards are connected.
- Run both projects from the debugger (or by pushing reset switch if boards already programmed).
- When connection is established, the message “CONNECTION SUCCESSFUL!” is displayed.
- Start typing from the computer on client side. Data should be printed on server’s side terminal.



```
*************************************

       NN   NN XX    XX PPPPP
       NNN  NN  XX  XX  PP  PP
       NN N NN   XX     PPPPP
       NN  NNN  XX  XX  PP
       NN   NN XX    XX PP

S32K148 ENET + lwIP demo [TCP CLIENT]

*************************************

Connecting to server...

CONNECTION SUCCESSFUL!

Type your message:

```



```
*************************************

       NN   NN XX    XX PPPPP
       NNN  NN  XX  XX  PP  PP
       NN N NN   XX     PPPPP
       NN  NNN  XX  XX  PP
       NN   NN XX    XX PP

S32K148 ENET + lwIP demo [TCP SERVER]

*************************************

Ready to accept connection...

CONNECTION SUCCESSFUL!

Message received:

Hello
```

SECURE CONNECTIONS
FOR A SMARTER WORLD