AH1805

TC10 Starter Kit Application Hint

Rev.1.0 — 8. Nov 2018   **User Manual**

Revision history

Rev	Date	Description
1.0	08.11.2018	Initial release

Contact information

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: mailto:salesaddresses@nxp.com

AH1805

User Manual **Rev.1.0 — 8. Nov 2018** **2 of 26**

1. Introduction

This user manual describes the TC10 Starter kit hardware and the software components to allow the user to quickly bring up working wake-up and sleep scenarios that allow for further adaptations towards a customized system.

After the HW description the manual walks the user through the process to create and download the TC10 kit firmware images.

2. Overview

2.1 TC10 standard

TC10 is the abbreviation for the "10th technical committee of the Open Alliance Special Interest Group". The Open-Alliance focuses on One-Pair Ethernet and is an open industry alliance of mainly automotive industry and technology providers collaborating to encourage wide scale adoption of Ethernet based networks as the standard in automotive networking applications.

TC10 defines methods for sleep mode and a wake-up mechanism tailored for automotive use cases[1].

2.2 Objectives and features of the kit

The TC10 starter kit demonstrates the wakeup and sleep capabilities of the 100BASE-T1 PHY products TJA1101 and TJA1102. Furthermore, it allows to study how to exploit these capabilities in own implementations and provides the HW test bed for prototypical implementation.

2.3 Hardware and Software components

The starter kit consists of the following components (Fig 1):

---

[1] www.opensig.org/download/document/220/TC10+Wake-up+and+Sleep+Specification+for+Automotive+Ethernet_11-2017.pdf

Fig 1.   TC10 Starter Kit Components
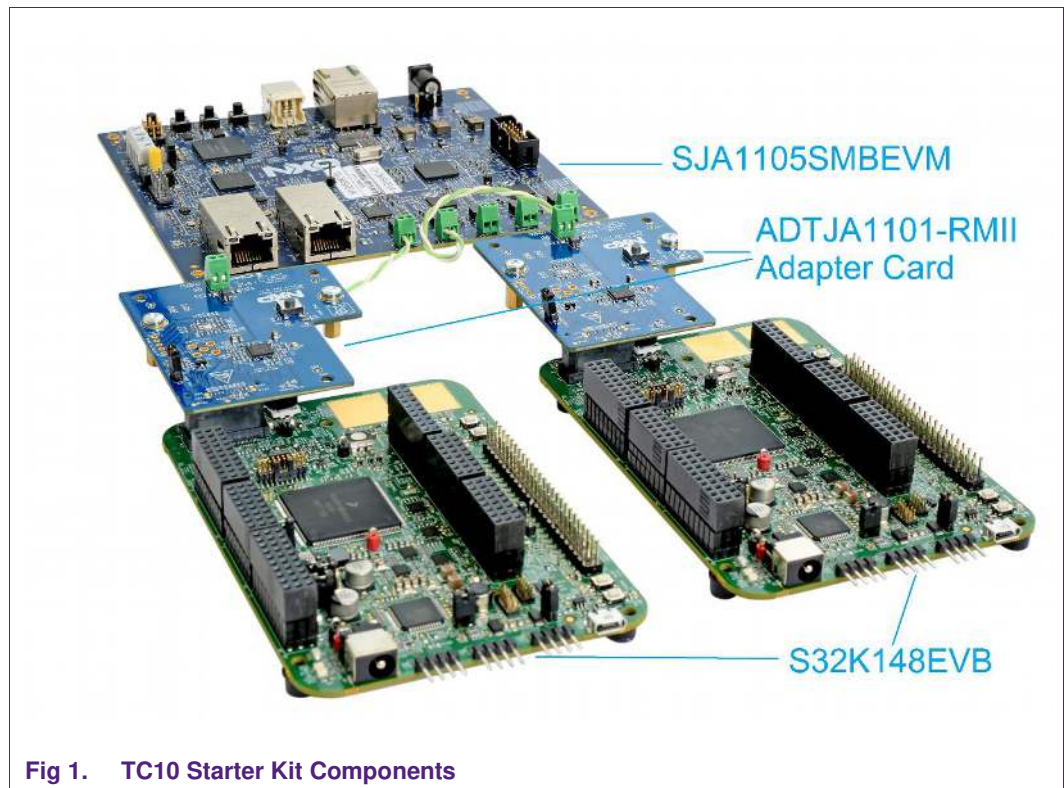
Gateway (central) node:

SJA1105SMBEVM

Satellite (end) node:

S32K148EVB with TJA1101 Adapter Card (ADTJA1101-RMII)

Please note that Fig 1 only shows the minimum setup, but further extension with more end nodes / switches is possible.

3.   Getting started

3.1   Getting components and documents

The required components can be ordered or downloaded via the links below:

- S32K148EVB

  https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/s32-automotive-platform/s32k148-evaluation-board:S32K148EVB

- ADTJA1101-RMII

  http://www.nxp.com/products/analog/interfaces/in-vehicle-network/ethernet/automotive-ethernet-phy-transceivers/adtja1101-rmii-automotive-ethernet-adapter-for-s32k148evb:ADTJA1101-RMII

- SJA1105SMBEVB

AH1805

User Manual **Rev.1.0 — 8. Nov 2018** **4 of 26**

https://www.nxp.com/support/developer-resources/evaluation-and-development-boards/analog-toolbox/gateway-prototyping-platform:SJA1105SMBEVM

- S32K SW project

  https://www.nxp.com/search?keyword=s32k%20144%20projects

- SJA1105SMBEVM SW project

  https://www.nxp.com/support/developer-resources/evaluation-and-development-boards/analog-toolbox/gateway-prototyping-platform:SJA1105SMBEVM?tab=Design_Tools_Tab

Additional materials such as SJA1105S/TJA1101/TJA1102 datasheet, application hints, ADTJA1101-RMII schematics, SJA1105SMBEVM schematics, user manual, firmware can be downloaded from NXP's document repository www.docstore.nxp.com.

- Register (NDA required) / Login

- Products -> In-Vehicle Networking -> Automotive Ethernet -> …(corresponding folders)

Other documents such as S32K148 datasheet, errata, S32K148EVB user manual can be downloaded from https://www.nxp.com by searching keyword "S32K148" or below link:

- https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/s32-automotive-platform/s32k148-evaluation-board:S32K148EVB?fpsp=1&tab=Documentation_Tab

The firmware of the S32K148EVB is included in the newest S32DS for ARM. Please refer to chapter 3.4.2 for more details.

3.2 Additional prerequisites

Besides the components and documents which are described in chapter 3.1, you may need the following support materials: power supply, USB cable, debugger and Ethernet cable.

3.2.1 Power supply

The SJA1105SMBEVM is powered by a single 12V supply input. It's recommended to use an adapter that can deliver at least 1A of current.

The S32K148EVB can be powered by either a single 12V supply input or by USB power. By default, USB power supply is enabled by J8 jumper. Please refer to the user manual of the board for more details of jumper settings.

The ADTJA1101-RMII is powered from the S32K148EVB via the SABRE connector.

3.2.2 Micro USB Cable

A micro USB cable is necessary for the S32K148EVB, which can be used for SW debug and download. Additionally it also can be used to supply the S32K148EVB in case no external 12V power supply is available.

3.2.3 Debugger

If you want to download and run SW on the SJA1105SMBEVM you also need a JTAG-Debug adapter. The easiest solution is the PEMicro "USB Multilink Universal" USB-to-JTAG debug probe (http://www.pemicro.com/products/product_viewDetails.cfm?product_id=15320168) as this is fully supported by S32 Design Studio IDE.

No special debugger for the S32K148EVB is needed, since it contains a Single-Wire-Debug probe based on OpenSDA. Only a Micro-USB cable is required to connect the board via J7 to the PC.

3.2.4 Ethernet cable

A standard IEEE100BASE-T1 cable is provided with the ADTJA1101-RMII board. The customer can check with the cable suppliers to get the standard 100BASE-T1 cable in case more connections are needed between 100BASE-T1 ports.

3.3 Installation and Connection

The ADTJA1101-RMII board is connected to the S32K148EVB board via the SABRE (Smart Application Blueprint for Rapid Engineering) connector, which is shown in Fig 2. ADTJA1101-RMII board and S32K148EVB board connection is shown in Fig 3.
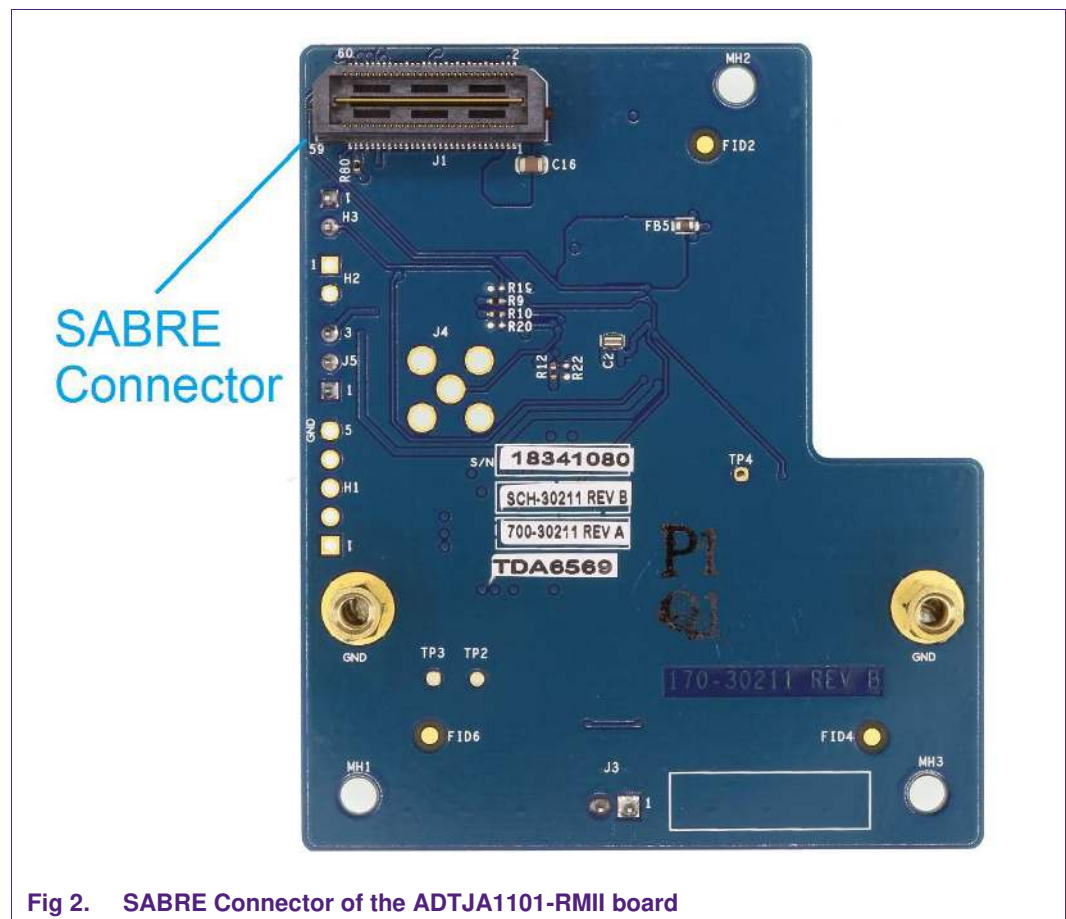

Fig 2.   SABRE Connector of the ADTJA1101-RMII board

AH1805

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2018. All rights reserved.

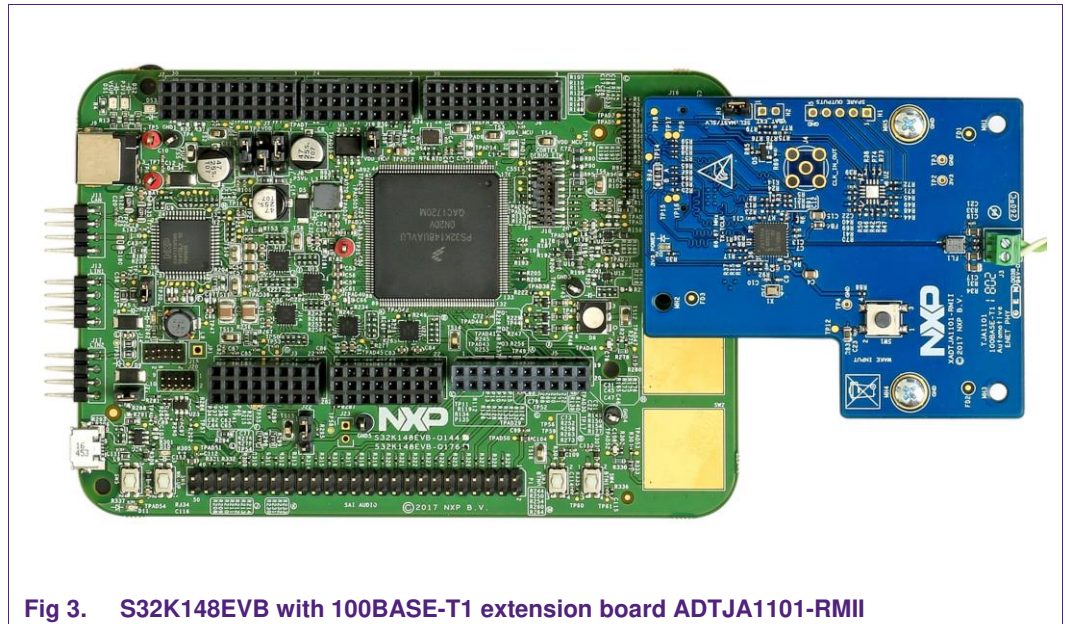**User Manual** **Rev.1.0 — 8. Nov 2018** **6 of 26**

Fig 3.   S32K148EVB with 100BASE-T1 extension board ADTJA1101-RMII

3.4 Firmware Installation on S32K148EVB

3.4.1 S32DS Installation

Installation of the SW development environment S32DS for ARM 2018 is necessary for bringing the S32K148EVB board to operation.

The S32 Design Studio for ARM 2018 home page is at:
https://www.nxp.com/support/developer-resources/run-time-software/s32-design-studio-ide/s32-design-studio-ide-for-arm-based-mcus:S32DS-ARM

The following packages are needed:

- the basic packet "S32 Design Studio for ARM 2018.R1 - Windows/Linux" and

- the cumulative update "S32 Design Studio for ARM 2018.R1 Update 7 with S32K1xx SDK BETA 2.9.0"

Please refer to the screenshots in Fig 4 for downloading the development environment and the SDK update.

The S32DS has its own documentation on installation and usage of the software, please refer to the documents for details of installation and usage.
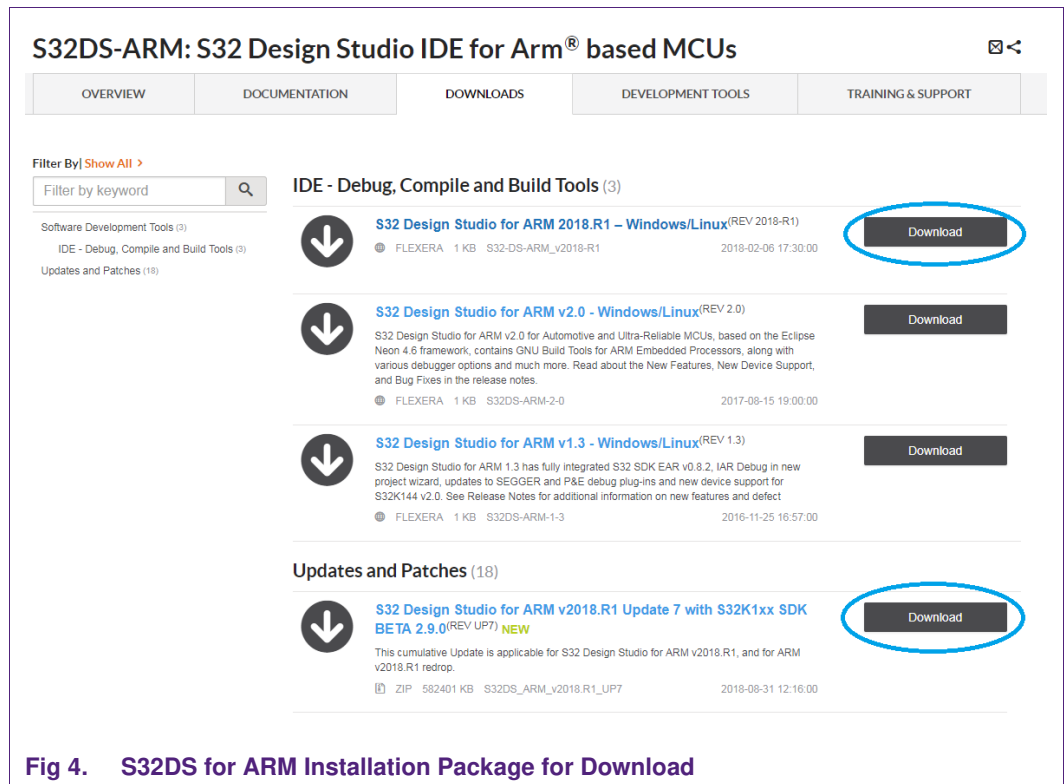
Fig 4. S32DS for ARM Installation Package for Download

Installing the base package is straight forward: just execute the downloaded program and answer the questions. At one point enter the license number you have received via email.

SDK 2.9.0 is not part of the base package, but it must be installed in a separate step.

This can be done either

- Online: S32DS/Eclipse downloads the package itself and installs it then. This requires that Eclipse can access the internet, which is often not possible due to corporate firewall rules and proxy architectures. Web browsers are usually capable to deal with these topics, but for other SW this is cumbersome.

- Offline: the update package is downloaded with the web browser and temporarily stored in the filesystem, just like the base package. For installing the update, the user must point Eclipse to the downloaded package.

The offline method is shown in the following steps:
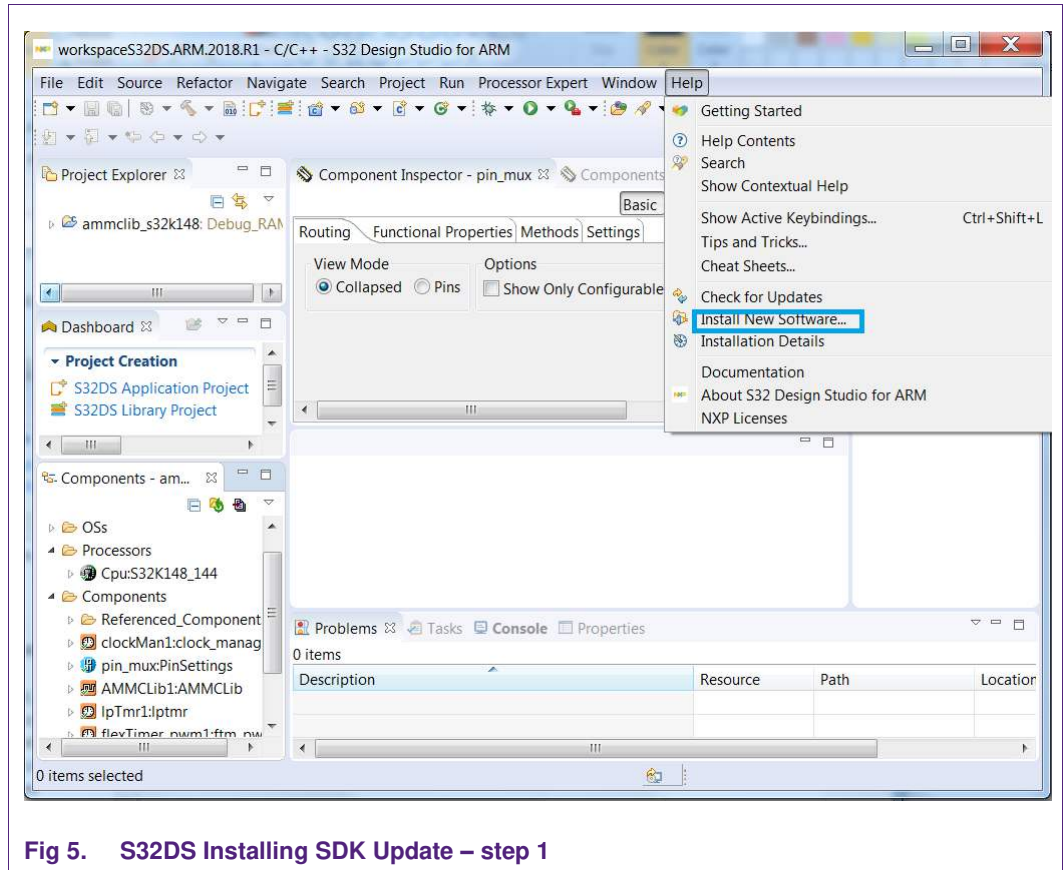
- In S32DS IDE select "Help -> Install New Software"

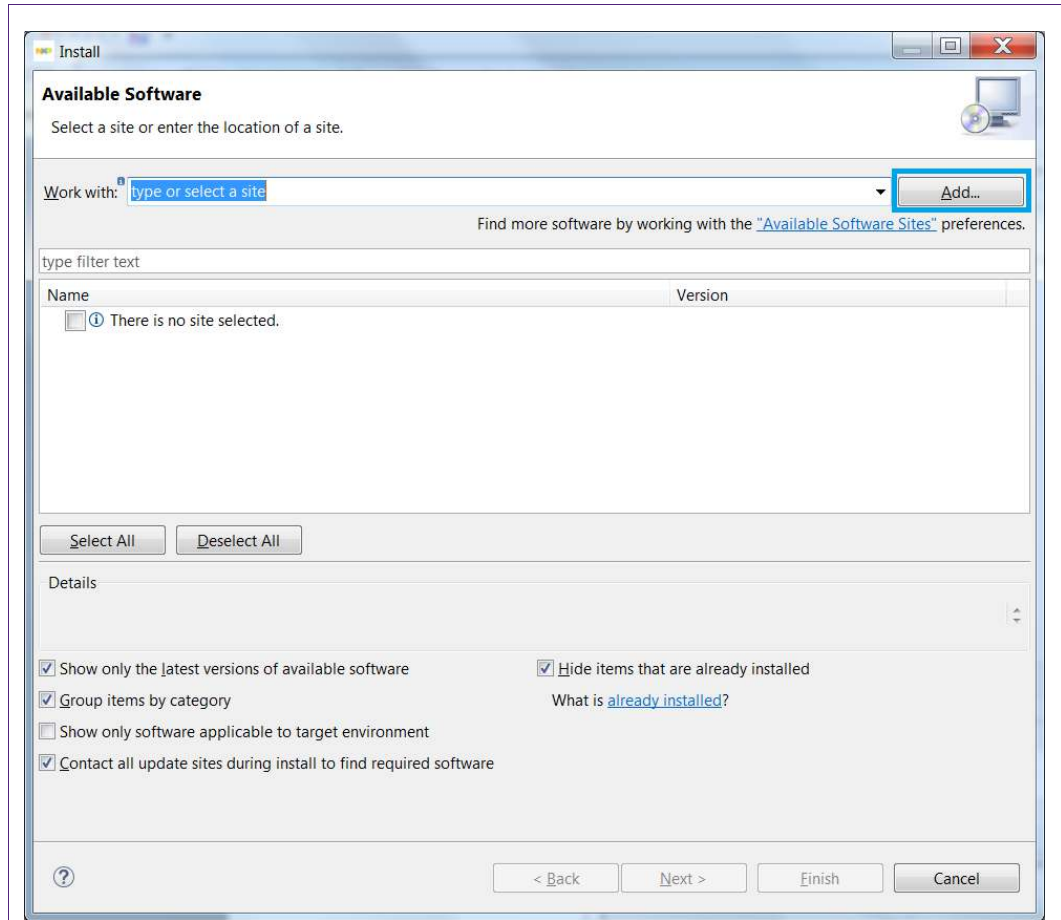**Fig 5.   S32DS Installing SDK Update – step 1**

- Click "Add…"

Fig 6.   S32DS Installing SDK Update – step 2

- Click "Archive…" and select the .zip file with the update/patch downloaded earlier and stored in the filesystem. Click "OK"
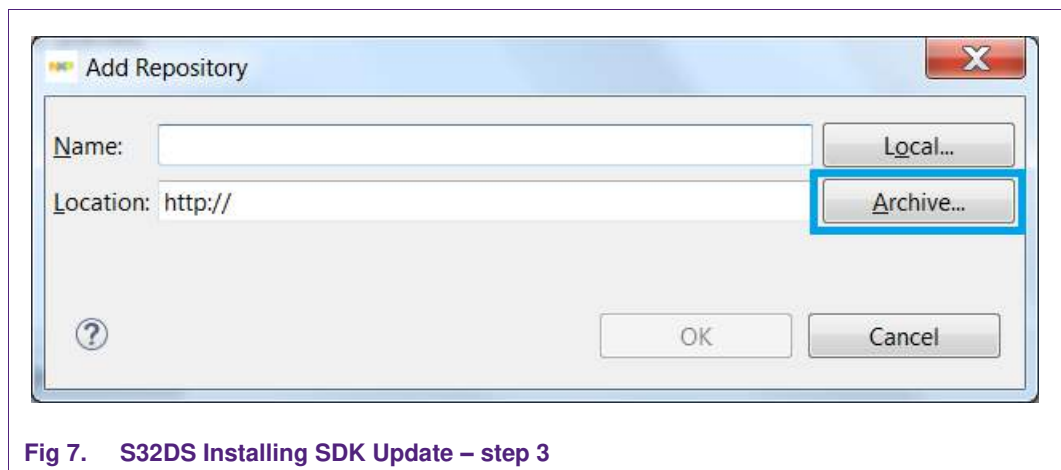

Fig 7.   S32DS Installing SDK Update – step 3

- It will start installing the patch to the S32DS ARM installation tree. Upon completion, you should see the Beta_2.9.0 folder.
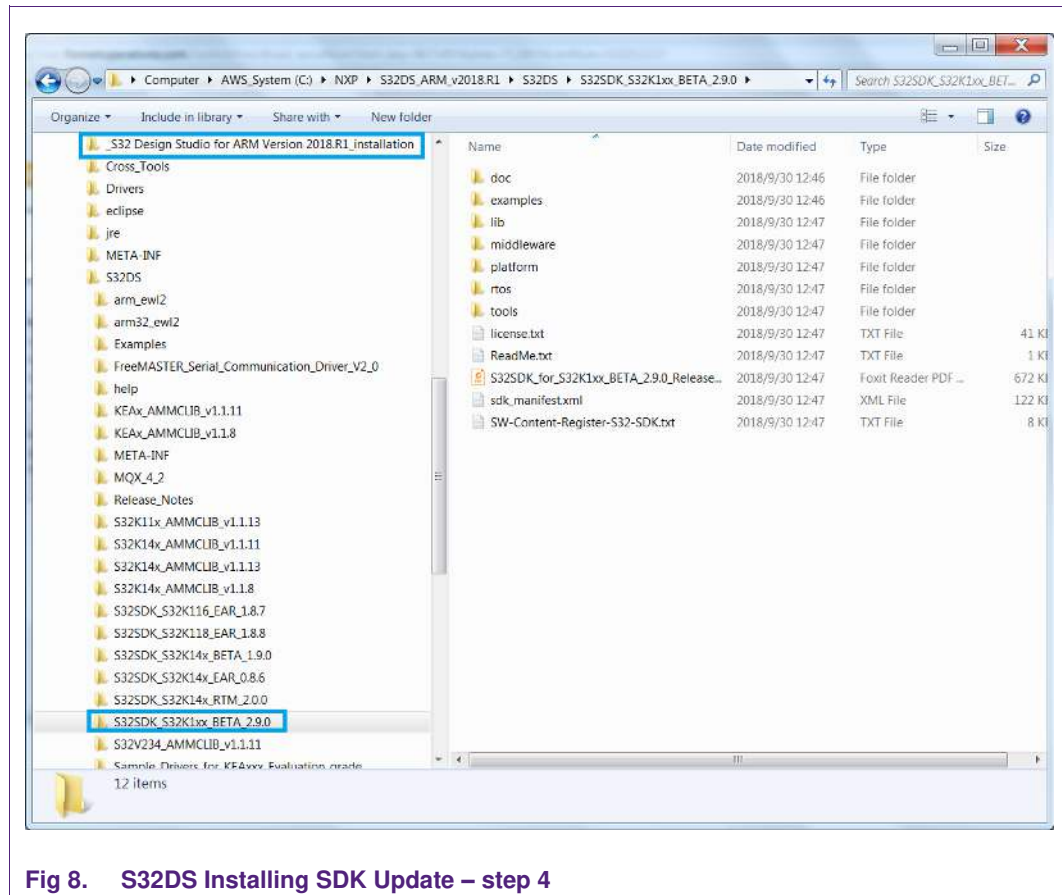

Fig 8.   S32DS Installing SDK Update – step 4

3.4.2 Import example project files and build an executable file

The following step-by-step guidance is just one way how to create an executable .elf file from the example project library, which is an known way of working, but there may be faster alternatives.

- Choose your workspace – usually just an (empty) directory in the user's part of the file system

- Start S32 Design Studio for ARM and select this workspace

- An empty Project Explorer pane (usually upper left corner) appears

- Load the example project with "File -> New -> S32DS Project from Example" and choose "S32K1xx BETA SDK v2.9.0 Example Project -> S32K148 -> driver_examples -> communication -> phy_tja1101_s32k148". Then click "Finish" to create a new project from the example project, which is just inside the workspace.
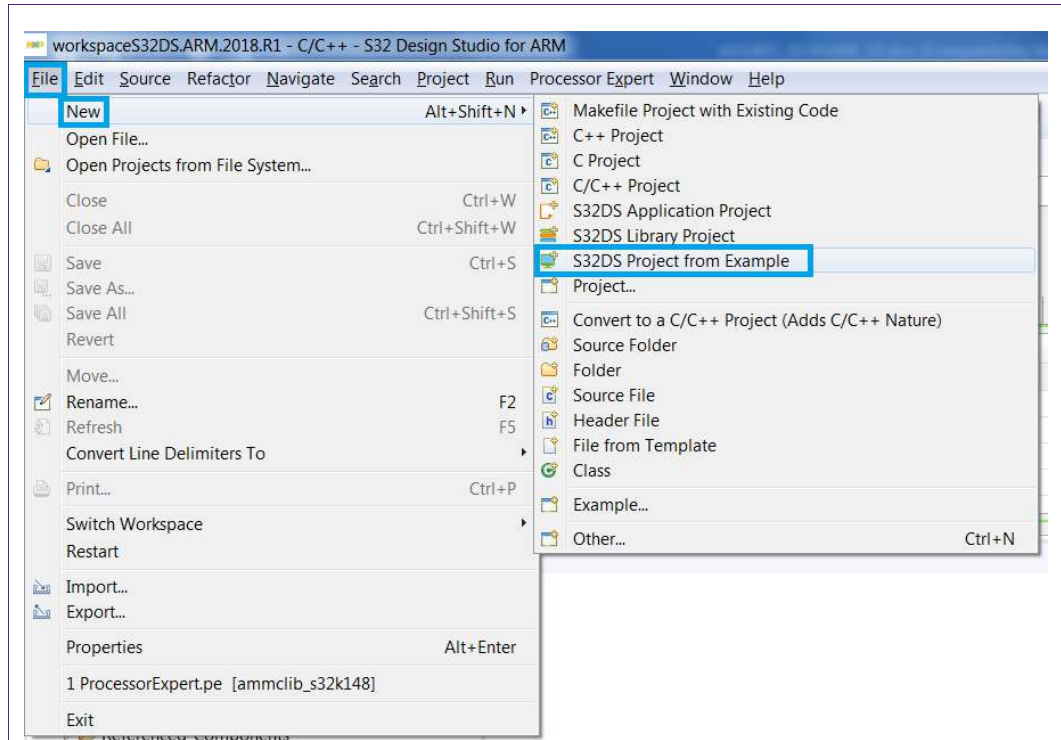
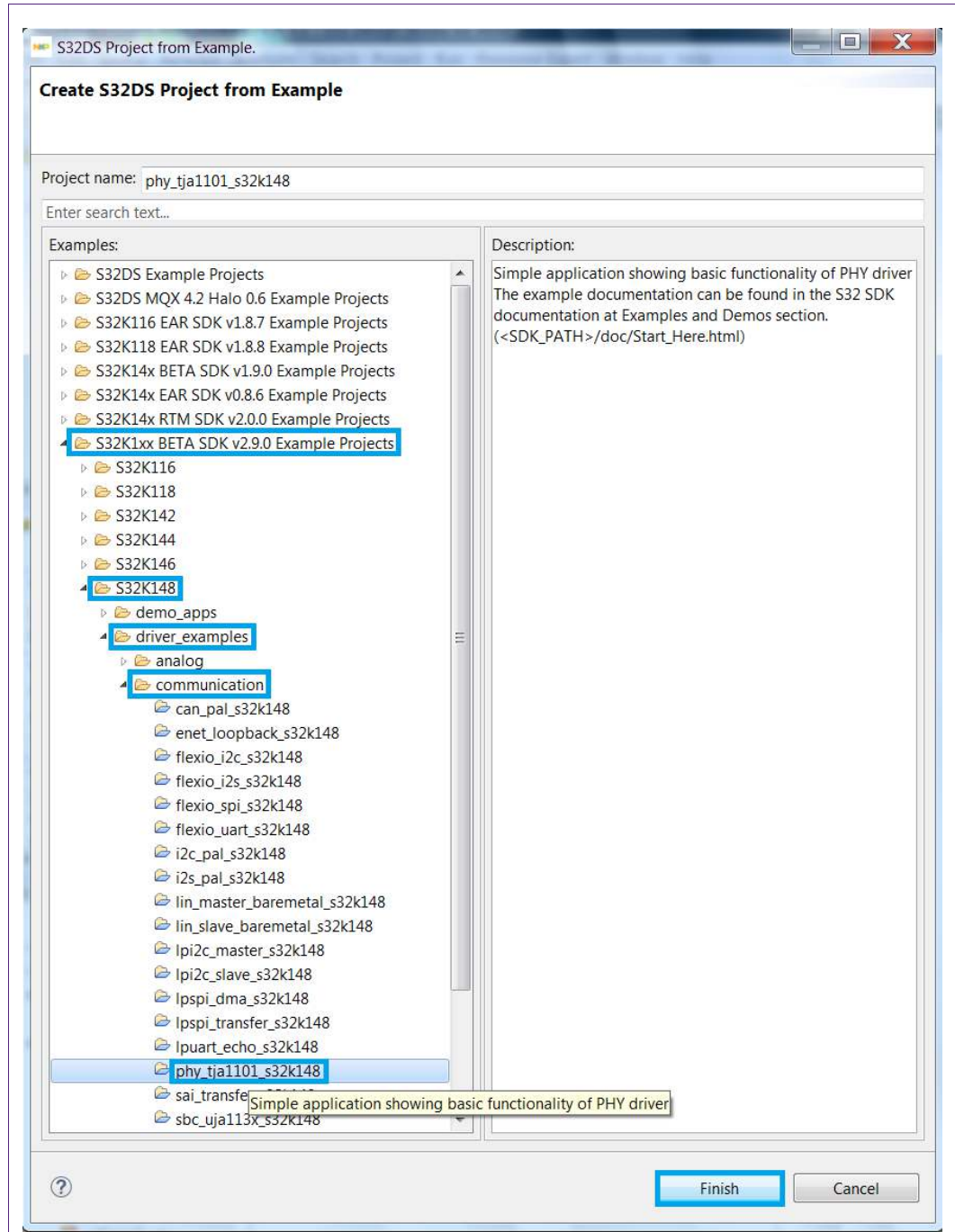**Fig 9.   S32DS Example Project Loading – step 1**

Fig 10. S32DS Example Project Loading – step 2

- A new project with the name "phy_tja1101_s32k148" appears in the project explorer. Now you can modify the example codes by your own. Fig 11 shows how to modify the Master/Slave configuration of the TJA1101 in S32DS: double click the "phy1: phy" (usually lower left corner), then select the Master/Slave/Automatic (pin strapping) in the drop-down list of the "PHY Role" box.
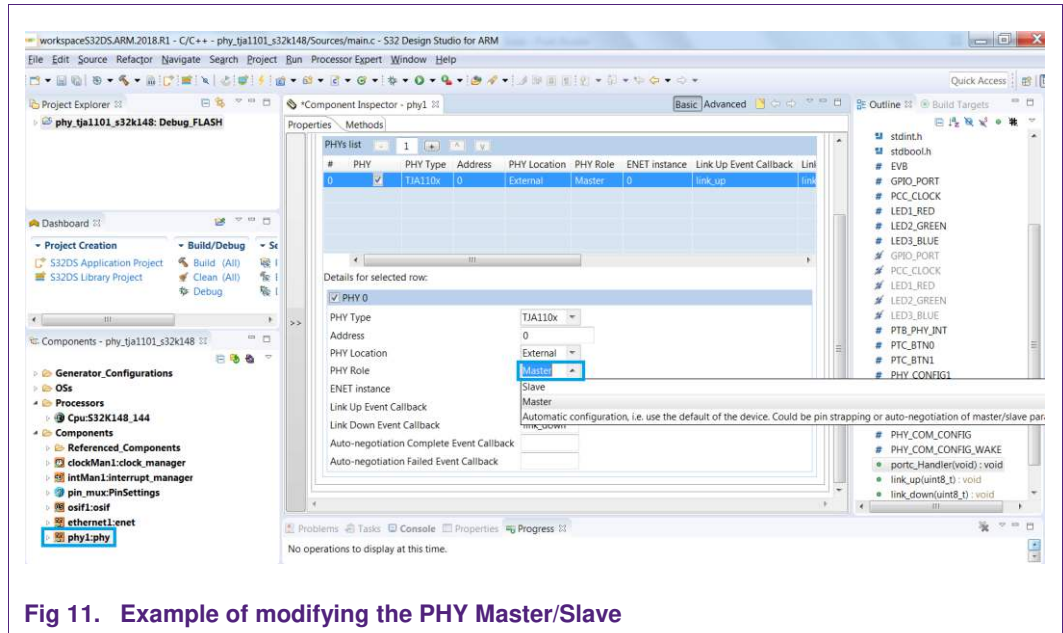
Fig 11. Example of modifying the PHY Master/Slave

- Generate the code after modification by clicking the marked symbol in Fig 12 (or by Project -> Generate Processor Expert Code) and watch the progress in the pop up progress bar or the Console window on the lower right pane.
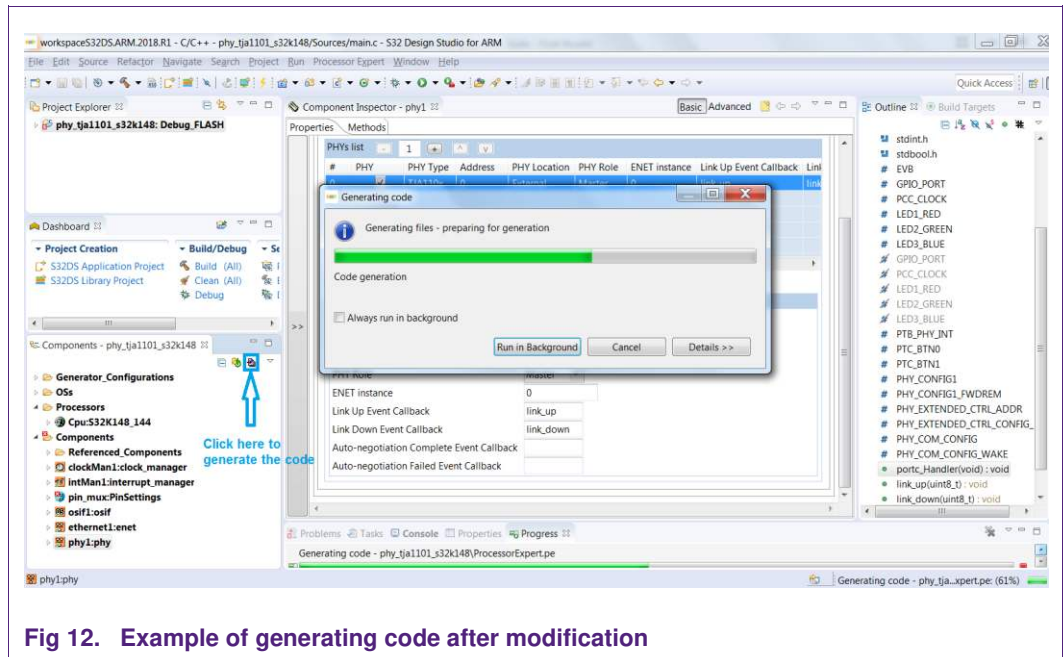

Fig 12. Example of generating code after modification

- Build the project by clicking the hammer symbol (or by Project -> Build All), using the build configuration "Debug_FLASH", and watch the progress in the pop-up progress bar or the Console window on the lower right pane
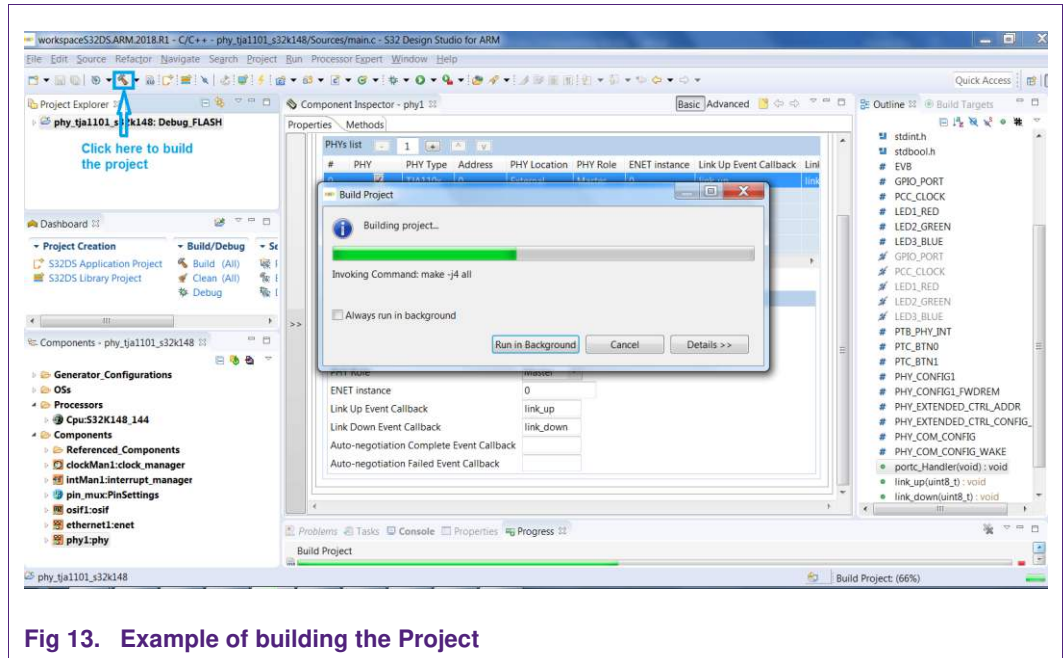
Fig 13.   Example of building the Project

3.4.3   Download and run the firmware

To download and debug your project with OpenSDA, it's necessary to follow below steps:

- Configure the debugger by clicking "Run -> DebugConfigurations" (also available under the beetle symbol). If you have chosen the example project as described in chapter 3.4.2, you will see a predefined debug configuration under "GDB PEMicro Interface Debugging". Make sure the "phy_tja1101_s32k148_debug_flash_pemicro" is appeared as default. Then click "Close" at the lower right pane.
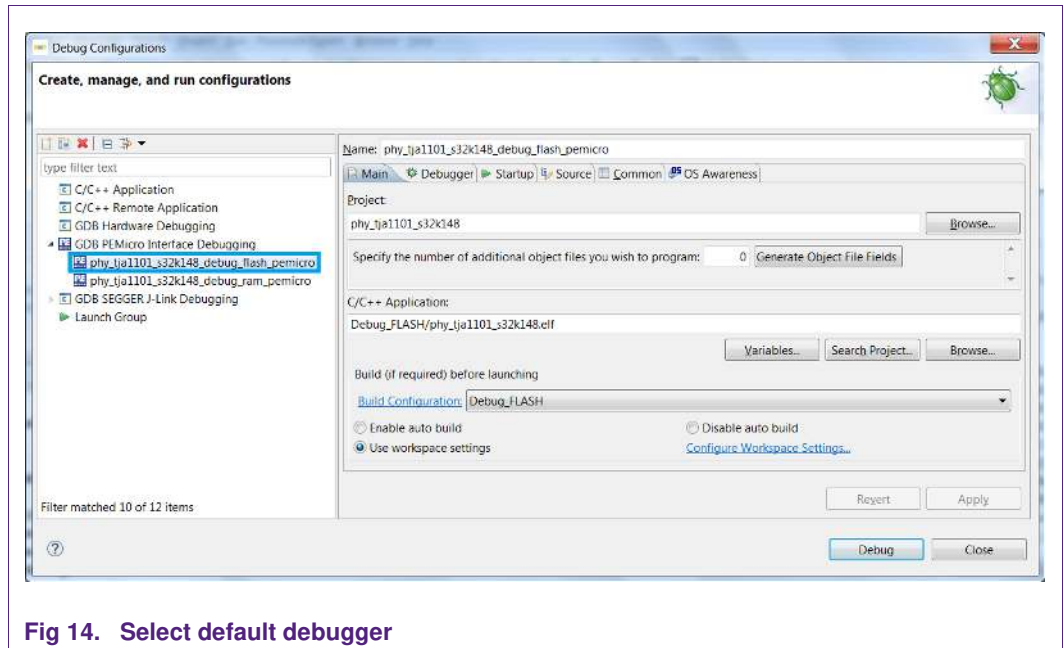

Fig 14.   Select default debugger

- Make sure the S32K148EVB is connected to the PC via Micro USB cable, and the board is powered up (two red LEDs will be on, please refer to the user manual of the board) before clicking "Run -> Debug" (or hit F11) or clicking on pull down arrow for debug icon and select…elf target.
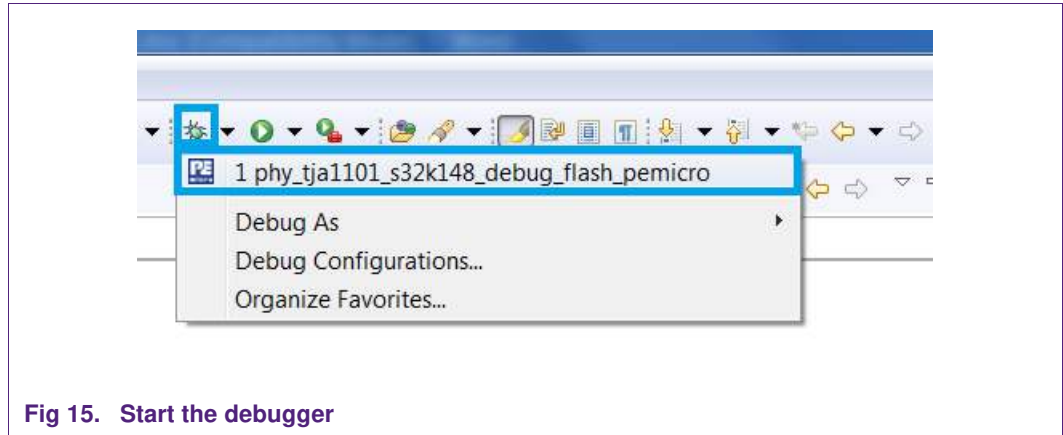

Fig 15.   Start the debugger

- If everything goes well, you will see the debug view, and the main window will show, that the program has stopped on "main()". Then you can select "Run -> Run" or use a shortcut key Ctrl + F11. The program continues, and the sleep/wake LED will be switched on (see Fig 17 for location of the sleep/wake LED, and Table 1 for description of the sleep/wake LED).

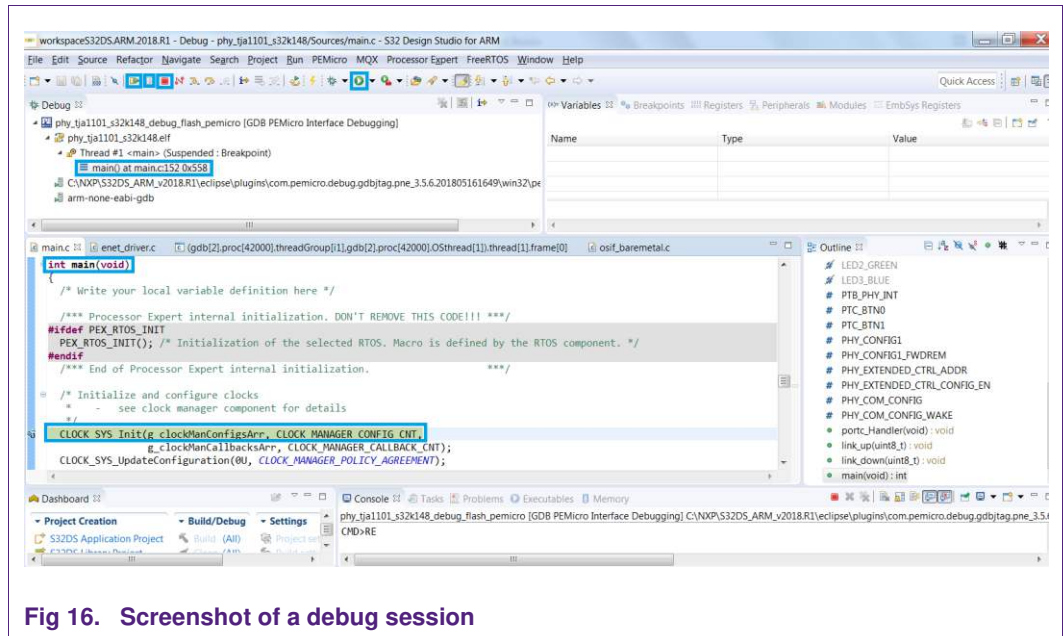 Please find more details of Step, Run, Suspend, Resume of the debug session in the board user manual.

Fig 16.   Screenshot of a debug session

AH1805

© NXP Semiconductors N.V. 2018. All rights reserved.

User Manual **Rev.1.0 — 8. Nov 2018** **16 of 26**

3.5 Firmware Installation on SJA1105SMBEVM

Please refer to the AH1721 about how to install the S32DS for the SJA1105SMBEVM, load the example project from a zip file, compile and download the firmware.

The example project zip file can be downloaded from the www.nxp.docstore.com. Please refer to chapter 3.1 of where to download the zip file.

3.6 Operation

3.6.1 Buttons and LEDs on S32K148EVB

On the S32K148EVB side, the wakeup button is located on the ADTJA1101-RMII daughter board, while for going to sleep, one of the general-purpose buttons (BTN1, see Fig 17) is used. The Sleep Request SMI command will be sent to the TJA1101 when pushing the sleep button on the S32K148EVB board. A wakeup request will be sent to the TJA1101 when pushing the wakeup button on the ADTJA1101-RMII board (refer to the TJA1101 datasheet for more details).

There are 3 options to change the Master/Slave configuration of the TJA1101:

- Push the two push buttons (BTN0 & BTN1, see Fig 17) simultaneously. The PHY configuration will be swapped: Master will become Slave and vice versa.

- Configure the Master/Slave in S32DS. Please refer to chapter 3.4.2 for more details

- Configure the Master/Slave during startup of the TJA1101 via the jumper "SEL_MAST/SLV" (Open=Master, Closed=Slave) on the ADTJA1101-RMII board, in case the TJA1101 is configured as "Automatic configuration…" in S32DS (see Fig 11). Please refer to the user manual of the ADTJA1101-RMII regarding the jumper settings

The 3-color LED on the S32K148EVB which is marked as "Wake/Sleep LED" in the Fig 17 is used to indicate the TJA1101 sleep/link up status:

Table 1. Sleep/Wake LED of S32K148EVB

LED label	Color	Description
Sleep/Wake	Blue	the TJA1101 in Sleep mode
	Red	the link of the TJA1101 is not established
	Green	the link of the TJA1101 is established already

Please note that the connection to ports A1 and B2 in Fig 17 is exemplarity. It can be connected to any other 100BASE-T1 port of the SJA1105SMBEVM, such as B3, B0 and B1 depending on the specified use case.
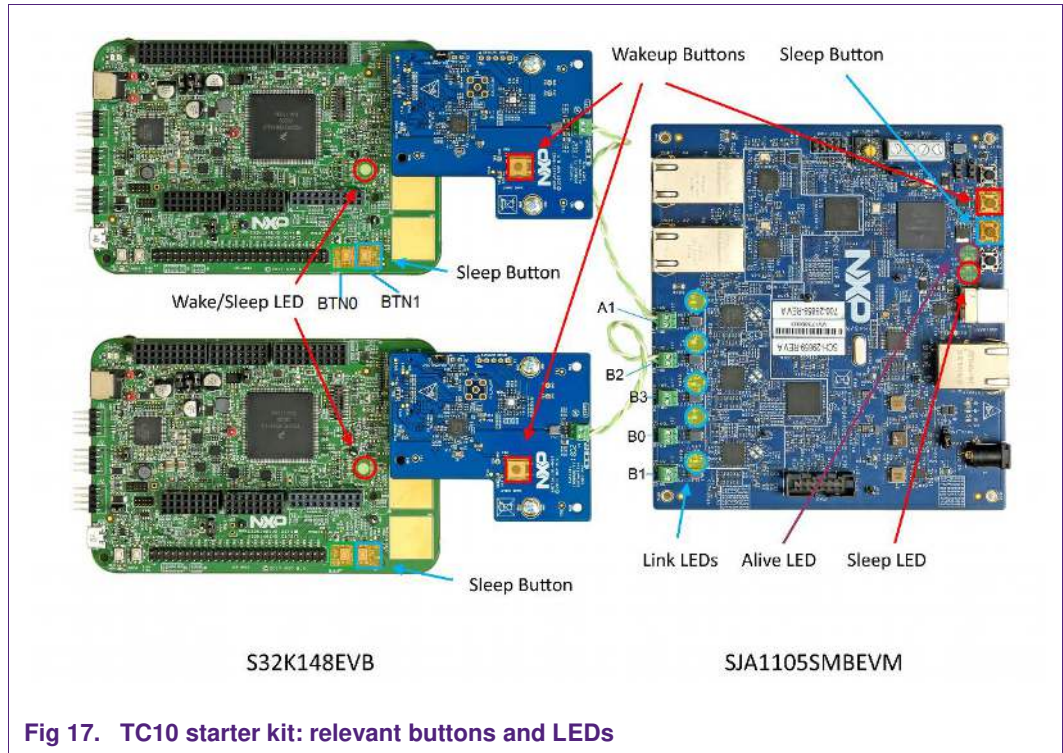
Fig 17. TC10 starter kit: relevant buttons and LEDs

3.6.2 Buttons and LEDs on SJA1105SMBEVM

The meanings of two push buttons and three status LEDs which shown in Fig 17 are described in below tables.

Table 2. Buttons of SJA1105SMBEVM

Button name/label	Description
Wakeup/SW1	Connects the PHYs' wakeup line with VCC – this is for local wake
Button 0/SW3	Sleep request SMI commands will be sent to all PHYs when pushing this button

Table 3. LEDs of SJA1105SMBEVM

LED name/label	Color	Description
Link LEDs/D2, D4, D5, D15, D16	Green	Link-up indication LEDs for the 100BASE-T1 ethernet ports. Each LED is controlled by the on-board uC. and is set/cleared by the LinkUp/LinkFail interrupt of the associated port's PHY. D2: Port A1 D4: Port B0 D5: Port B1 D15: Port B2 D16: Port B3
Active LED/D3	Green	This LED should continuously blink if the example SW is running
Sleep LED/D17	Green	This LED will on in case all active links are sleeping

As shown in Fig 18, the Master/Slave settings of the 5 100BASE-T1 ports (Fig 17) can be re-configured by the S32DS GUI driver configuration. The default settings can be found in Table 4.
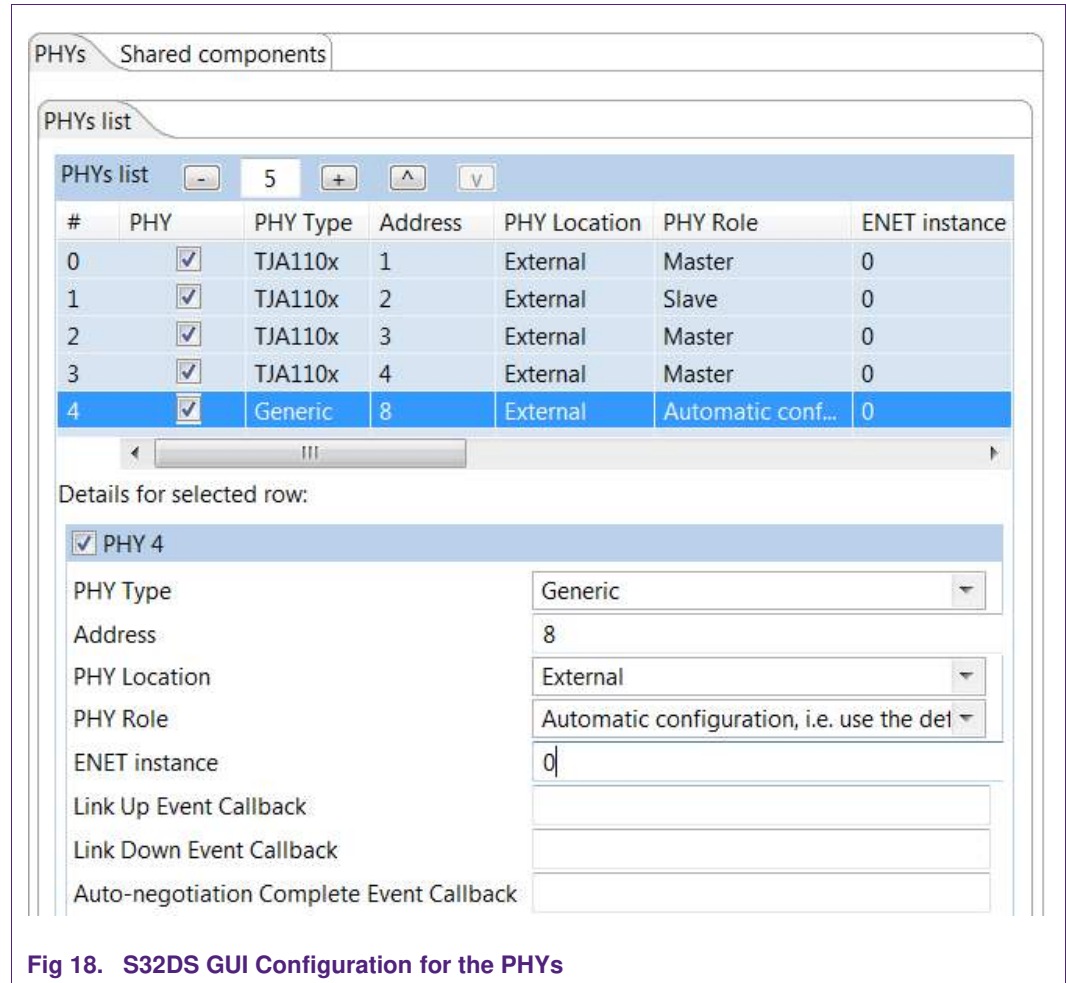

Fig 18.   S32DS GUI Configuration for the PHYs

Table 4.   Default PHY settings of SJA1105SMBEVM

Port	SMI Address	Master/Slave
A1	0x04	Master
B2	0x08	Master
B3	0x09	Master
B0	0x06	Master
B1	0x07	Master

3.6.3   Wakeup Forwarding Configuration

The TJA1101/TJA1102 PHY can be configured with various wakeup forward configurations. The following figure shows the logic behind the possible options:
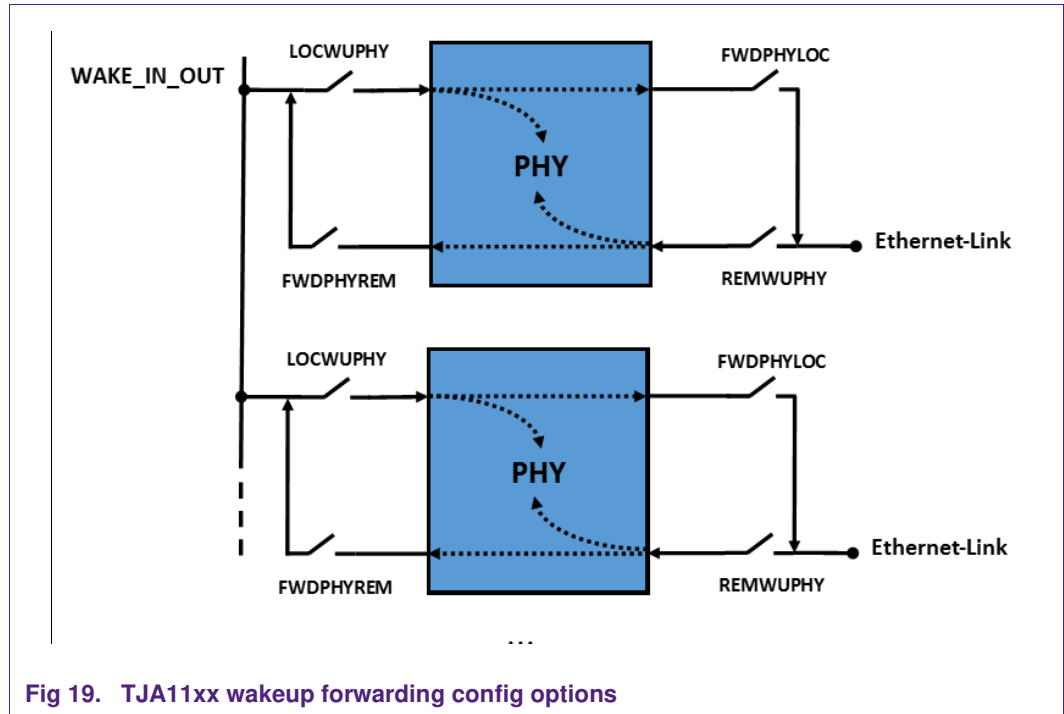
Fig 19. TJA11xx wakeup forwarding config options

The SJA1105SMBEVM software configures the TJA1102(S) PHYs on the SJA1105SMBEVM for wakeup forwarding in the following way:

Table 5.   Wake-up & forwarding configuration of the TJA1102s on SJA1105SMBEVM

Link name	PHY	Address	WAKE_IN_OUT	Remote Wake Up
A1	TJA1102S	3	PHY + Forward	PHY + Forward
B0	TJA1102 ch0	4	PHY + Forward	PHY only
B1	TJA1102 ch1	5	PHY + Forward	PHY + Forward
B2	TJA1102 ch0	6	PHY + Forward	Ignore
B3	TJA1102 ch1	7	PHY only	PHY + Forward

Note that table 5 only shows the settings included in the provided software project. It can be changed by modifying the supplied source files.

Table 6.   Wake-up & forwarding configuration of the TJA1101 on S32K148EVM

Link name	PHY	Address	WAKE_IN_OUT	Remote Wake Up
	TJA1101	0	PHY + Forward	PHY + Forward

The column "WAKE_IN_OUT" refers to wakeup signal sent to the PHY via a rising pulse on the WAKE_IN_OUT line, which connects the WAKE_IN_OUT pin of all PHYs on the SJA1105SMBEVM. The pulse may be generated by pushing the wakeup button which is shown in Fig 17.

The column " Remote Wake Up" shows the configured behavior of the PHY when a WUP (Wake Up Pulse) or WUR (Wake Up Request) is received from the PHY on the other end of the ethernet link. Please refer to the TJA1102 DS/AH for more details of WUP and WUR.

"Ignore" means, that the PHY doesn't react to a wake up at the WAKE_IN_OUT (LOCWUPHY = 0) or remote wake up (REMWUPHY = 0) - ignored.

"PHY only" means, that a wake up at the WAKE_IN_OUT (LOCWUPHY = 1, FWDPHYREM = 0) or remote wake up (REMWUPHY = 1, FWDPHYLOC = 0) wakes up the PHY on the SJA1105SMBEVM only, without wake up forwarding function.

"PHY + Forward" means, that a wakeup request is also forwarded to "the other side", i.e. from the remote wake up to the WAKE_IN_OUT (REMWUPHY = 1, FWDPHYLOC = 1) or from WAKE_IN_OUT to the ethernet link (LOCWUPHY = 1, FWDPHYREM = 1).

3.6.4 Typical sequences

3.6.4.1 Master/Slave setup

When powering up the boards of the TC10 Starter Kit, links between S32K148EVBs and the SJA1105SMBEVM might not come up, because there is a role conflict of either Master-Master or Slave-Slave.

Please refer to chapter 3.6.1 regarding how to change the Master/Slave configurations of the TJA1101 on the ADTJA1101-RMII. If a link is successfully established, the link LED on the SJA1105SMBEVM goes on, and the Sleep/Wake LED on the S32K148EVB goes from red to green (see Table 1).

3.6.4.2 Send all active PHYs to sleep

Pressing " "Sleep button" (Fig 17) on the SJA1105SMBEVM writes a sleep request to each PHY. This results in sending a sleep request over the link to the link partner. All PHYs on the SJA1105SMBEVM and the TJA1101 on the ADTJA1101-RMII will go to sleep if no data is detected on MII or MDI during the intermediate state (Sleep Request mode) before Sleep mode. Please refer to the TJA1102 application hint (AH1508) for more details of the process from active link to sleep.

3.6.4.3 Wake up and wake up forwarding evaluation

Please note that the below behaviors are based on default configurations in Table 5 & Table 6, and the connection shown in Fig 17**Error! Reference source not found.**. The behavior may be different with different wake up and wake up forwarding configurations.

- There are two options to wake up the slept PHYs after sleep operation described in chapter 3.6.4.2.Option 1: press the "Wakeup button" on the SJA1105SMBEVM board (Fig 17). A rising wake up signal will be sent to the WAKE_IN_OUT pin of the PHYs on the SJA1105SMBEVM. All PHYs on the board will be waked up.

- As described in Table 5, A1, B0, B1 and B2 will forward the local wake up request to the link partner by sending wake up request over the link. Both TJA1101 will be waked up since they are configured as "react on remote wake up". All Link LEDs on the SJA1105SBEVM will be switched on after a while.

- Option 2: press the "Wakeup button" on the ADTJA1101-RMII which is connected to the A1 port of the SJA1105SMBEVM via Ethernet cable (Fig 17). The TJA1101 will be waked up by local wake up signal and send out wake up request over the link. The TJA1102S (A1 port) will be waked up by the remote wake up signal, and then forward the wake-up request to the WAKE_IN_OUT pin by generating a rising edge (configured as "react on remote wake up, and forward remote wake up to local"). All PHYs on the SJA1105SMBEVM will be waked up since all of them are configured as "react on local wake up".

- B2 port will forward the local wake up to the connected ADTJA1101-RMII by sending out wake up request over the link. All Link LEDs on the SJA1105SBEVM will be switched on after a while.

The wake-up forwarding can be evaluated by following steps:

- Assume all PHYs are in Sleep mode after sleep operation described in chapter 3.6.4.2.

- Press the "Wakeup button" on the ADTJA1101-RMII which is connected to the B2 port of the SJA1105SMBEVM (Fig 17). The TJA1101 will be waked up by local wake up signal, and send out wake up request to B2 over the link. B2 will stay in Sleep mode since it's configured as "no react on remote wake up".

- Disconnect the Ethernet cable between the ADTJA1101-RMII and the B2 port, and connect the ADTJA1101-RMII to the B0 port. Repeat above two steps, and the TJA1102 (B0 port) will be waked up but all other PHYs on the SJA1105SBEVM will stay in Sleep mode, since B0 is configured as "react on remote wake up, but no forward remote wake up to local". The link LED of B0 will be switched on.

- Another alternative solution is to keep the Ethernet connection between the ADTJA1101-RMII and the B2 port, but reconfigure the wake up and wake up forwarding of the TJA1102 (B2), such as configure as "react on remote wake up, but no forward remote wake up to local (REMWUPHY = 1, FWDPHYLOC = 0)"

4. Legal information

4.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

4.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

4.3 Licenses

Purchase of NXP <xxx> components
<License statement text>

4.4 Patents

Notice is herewith given that the subject device uses one or more of the following patents and that each of these patents may have corresponding patents in other jurisdictions.

<Patent ID> — owned by <Company name>

4.5 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

<Name> — is a trademark of NXP Semiconductors N.V.

AH1805

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 20184. All rights reserved.

User Manual   **Rev.1.0 — 8. Nov 2018**   **23 of 26**

5. List of figures

6. List of tables

AH1805

© NXP Semiconductors N.V. 2018. All rights reserved.

User Manual **Rev.1.0 — 8. Nov 2018** **25 of 26**

7. Contents

AH1805

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2018. All rights reserved.

User Manual | **Rev.1.0 — 8. Nov 2018** | **26 of 26**