AN14188

Windows 10 IoT企业版以太网PHY配置

第1版—2024年5月27日

应用笔记

文档信息

信息	内容
关键词	以太网PHY、ENET_QOS、ENET、Windows 10 IoT企业版BSP、调试、Windows驱动程序、ACPI表、U-Boot、Windbg
摘要	本文档描述在采用恩智浦i.MX系列SoC的Windows 10 IoT板级支持包（BSP）上配置和调试以太网物理层（PHY）的必要步骤。本文以Realtek RTL8211和TI DP83867两款PHY芯片为例，展示了如何在U-Boot、EFI以及Windows驱动程序中进行相关配置。

1 介绍

本文档描述用户如何在采用恩智浦i.MX系列SoC的Windows 10 IoT板级支持包上配置和调试以太网物理层。Windows 10 IoT BSP支持两种以太网IP模块：ENET和ENET_QOS。i.MX 8M Plus和i.MX 93中包含这两种模块，而i.MX 8M、i.MX 8MM、i.MX 8MN和i.MX 8QXP中包含ENET模块。需要关注的配置部分包括：

- U-Boot
- EFI
- Windows驱动程序

文中以两款PHY芯片为例进行配置说明：

- Realtek RTL8211
- TI DP83867

为了充分理解本文档内容，建议事先熟悉以下相关文档：
i.MX Windows 10 IoT快速入门指南
i.MX Windows 10 IoT用户指南
i.MX Windows 10 IoT版本说明

2 配置示例

本节通过两个PHY的具体案例来展示PHY的配置过程。

2.1 U-Boot中ENET和ENET_QOS的PHY配置

U-Boot已经实现了对多PHY的支持。

根据Openwrt论坛的信息，U-Boot（与Linux类似）通过遍历可用驱动列表并使用首个匹配部分PHY ID（例如，数据手册中的PHY标识寄存器1和2）的驱动来选择PHY驱动。因此，在U-Boot中为ENET或ENET_QOS配置以太网PHY没有本质区别。

要编译支持特定PHY的U-Boot，需要将其配置特性添加到U-Boot配置文件中。示例1展示了i.MX 8M Plus的配置示例，位于文件uboot-imx/configs/imx8mp_evk_nt_uuu_defconfig中。

示例1

```
CONFIG_PHY_REALTEK=y
CONFIG_PHY_ATHEROS=y
CONFIG_PHY_TI_DP83867=y
```

U-Boot中所有可用的PHY驱动可以在文件uboot-imx/drivers/net/phy/Makefile中找到：

示例2

```
obj-$(CONFIG_BITBANGMII) += miiphybb.o
obj-$(CONFIG_B53_SWITCH) += b53.o
...
obj-$(CONFIG_PHY_ATHEROS) += atheros.o
...
obj-$(CONFIG_PHY_REALTEK) += realtek.o
...
```

```
obj-$(CONFIG_PHY_TI_DP83867) += dp83867.o
...
```

PHY复位的配置位于文件uboot-imx/arch/arm/dts/imx8mp-evk-u-boot.dtsi中。

示例3

```
&ethphy0 {
 reset-gpios = <&gpio4 22 1>;
 reset-assert-us = <15000>;
 reset-deassert-us = <100000>;
};

&fec {
 phy-reset-gpios = <&gpio4 2 1>;
 phy-reset-duration = <15>;
 phy-reset-post-delay = <100>;
};
```

关于在U-Boot中更改PHY的更多信息，请参见恩智浦社区。

2.2 以太网MAC（ENET）PHY配置

本节详细介绍以太网MAC（ENET）PHY的配置。

2.2.1 EFI ENET PHY配置

EFI负责配置RGMII的管脚和时钟，而ACPI表则提供Windows ENET驱动使用的信息。

2.2.1.1 EFI ENET PHY管脚和引脚布线

RGMII接口的管脚和时钟初始化实现以下函数：

- VOID EnetInit(VOID)

该函数在每个平台的iMX8BoardInit.c文件中实现，例如：/mu_platform_ nxp/NXP/MX8M_PLUS_EVK/ Library/iMX8BoardLib/iMX8BoardInit.c。

示例4。iMX8BoardInit.c中的引脚复用设置

```
VOID EnetInit(VOID)
{
// ENET1/2 MDIO bus (both ENETs share one MDIO bus connected to the ENET1
 controller)
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD2 = IOMUXC_MUX_ALT4; // ENET1_MDC -> PAD_SAI1_RXD2
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD3 = IOMUXC_MUX_ALT4; // ENET1_MDIO ->
 ENET1_MDIO_SELECT_INPUT
…
```

2.2.1.2 ENET ACPI表配置

对于Enet驱动，PHY类型和寄存器值在Dsdt-Enet.asl文件中设置，例如：mu_platform_nxp/NXP/MX93_11X11_EVK/AcpiTables/Dsdt-Enet.asl。

可用的寄存器设置命令：

- MII_REG_WR – 写入
- MII_REG_RMW – 读取、修改、写入

2.2.1.3 RTL8211 ACPI表设置

示例5。在ACPI表`Dsdt-Enet.asl`中为Enet设置RTL8211

```
 Name (_DSD, Package () {
   ToUUID("daffd814-6eba-4d8c-8a91-bc9bbf4aa301"),
   Package () { // RTL8211FDI-VD-CG
     Package (2) {"MDIOBusController InputClk kHz", 266000},
     Package (2) {"PhyAddress",                     0x00},
     Package (2) {"PhyInterafceType",               0x00},   // RGMII, default
value
     Package (2) {"PhyMaxMDIOBusClock_kHz",         15000},
     Package (2) {"PhyMinSTAHoldTime_ns",           10},
     Package (2) {"PhyDisablePreamble",             0},
     Package (2) {"ConfigCmds", Package () {
                    MII_REG_WR (0x1F, 0x0d08),         // Select page
                    MII_REG_RMW(0x11, 0x0000, 0x0100), // Enable Tx-delay
                    MII_REG_RMW(0x15, 0x0000, 0x0008), // Enable Rx-delay
                    MII_REG_WR (0x1F, 0x0d04),         // Select page
                    MII_REG_WR (0x10, 0x617F),         // Set green LED for
Link, yellow LED for Active
                    MII_REG_WR (0x1F, 0x0000),         // Set default page
                    ENET_MII_END}}
   }
 })
```

2.2.1.4 DP83867 ACPI表设置

示例6。在ACPI表`Dsdt-Enet.asl`中为Enet设置DP83867

```
 Name (_DSD, Package () {
   ToUUID("daffd814-6eba-4d8c-8a91-bc9bbf4aa301"),
   Package () { // RTL8211FDI-VD-CG
     Package (2) {"MDIOBusController_InputClk_kHz", 266000},
     Package (2) {"PhyAddress",                     0x00},
     Package (2) {"PhyInterafceType",               0x00},   // RGMII, default
value
     Package (2) {"PhyMaxMDIOBusClock_kHz",         15000},
     Package (2) {"PhyMinSTAHoldTime_ns",           10},
     Package (2) {"PhyDisablePreamble",             0},
     Package (2) {"ConfigCmds", Package () {
                    MII_REG_RMW(0x1F, 0x0000, 0x8000), // 3 Global Software
Reset 3 Global Software Reset 3 Global Software ResetGlobal Software Reset
(CTRLCTRL)
                    MII_REG_RMW(0x32, 0x0000, 0x0003), // Enable Shift mode for
both Rx/Tx (RGMIICTL)
                    MII_REG_WR (0x86, 0x0077),         // 2.0ns for Tx/Rx-delay
(RGMIIDCTL)
                    MII_REG_RMW(0x1F, 0x0000, 0x4000), // 3 Global Software
Reset 3 Global Software Reset 3 Global Software ResetGlobal Software Restart
                    MII_REG_WR (0x18, 0x5032),         // 1000BT, Link,
Receive, Transmit
                    ENET_MII_END}}
   }
   })
```

2.2.2 Windows驱动

ENET Windows驱动从上述ACPI表中读取所有PHY寄存器设置，因此在使用不同PHY时无需更改。

2.3 以太网服务质量（ENET_QOS）PHY配置

本节详细介绍以太网QOS（ENET_QOS）PHY的配置。

2.3.1 EFI ENET_QOS PHY配置

EFI配置RGMII的引脚和时钟，ACPI表提供Windows `ENET_QOS`驱动使用的信息，但不包括PHY寄存器设置。

2.3.1.1 EFI ENET_QOS PHY管脚和引脚布线

RGMII接口的管脚和时钟初始化实现以下函数：

- VOID EnetQosInit()

该函数在每个平台的`iMX8BoardInit.c`文件中实现，例如：`/mu_platform_ nxp/NXP/MX8M_PLUS_EVK/ Library/iMX8BoardLib/iMX8BoardInit.c`。

示例7。`iMX8BoardInit.c`中`ENET_QOS`的引脚复用设置

```
VOID EnetQosInit()
{
…
  /* Tx pads */
  IOMUXC_SW_MUX_CTL_PAD_ENET_TD0 = IOMUXC_MUX_ALT0;
  IOMUXC_SW_PAD_CTL_PAD_ENET_TD0 = IOMUXC_SW_PAD_CTL_PAD_FSEL_MASK |
 IOMUXC_SW_PAD_CTL_PAD_DSE(0x03);
```

2.3.1.2 ENET_QOS ACPI表配置

`ENET_QOS PHY`的寄存器设置在其Windows驱动程序中硬编码，必须在驱动程序中调整。

2.3.2 ENET_QOS Windows驱动

ENET_QOS以太网的PHY寄存器设置在Windows驱动程序中硬编码。对于RTL 8211，设置位于`MII_Rtl8211fInit`函数中。连接的PHY的检测由`MII_PhySpecificInit`函数完成。如果需要检测其他PHY，必须扩展供应商和型号开关以包含新的PHY标识，并实现新的函数，例如`MII_DP83867fInit`。代码示例可以在第5节找到。

3 常见问题

本节列出了在调试PHY时可能遇到的常见问题及其解决方案。

3.1 MAC地址缺失

当熔丝中未写入MAC地址时，可以在U-Boot、ACPI或通过Windows注册表进行设置，以便开发使用。

3.1.1 通过Windows注册表设置MAC地址

使用`ipconfig /all`命令，可以检查以太网接口的MAC地址。

如果物理地址无效（例如"`00-00-00-00-00-00`"），可以通过注册表编辑器或命令行设置地址。

3.1.1.1 使用Regedit输入MAC地址

1. 打开注册表编辑器，找到 `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Class{4D36E972-E325-11CE-BFC1-08002BE10318}\xxxx`
2. 检查文件夹（如0000、0001等），找到目标接口（DriverDesc = i.MX Ethernet adapter）。
3. 添加新的字符串变量NetworkAddress，格式为xx-xx-xx-xx-xx-xx。必须使用本地管理地址（LAA）。了解详情，请参见MAC地址。
4. 重启板卡。

3.1.1.2 使用REG命令输入MAC地址

在命令提示符窗口输入：

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Control\Class{4d36e972-e325-11ce-bfc1-08002be10318}\0000" /V NetworkAddress /T REG_SZ /D xx-xx-xx-xx-xx-xx /F
```

您需要像之前一样确保找到正确的文件夹（0000、0001等），识别要设置 MAC 地址的接口。

可以使用一个批处理脚本通过注册表设置MAC地址，链接为 http://lallouslab.net/2016/06/20/batchography-change-mac-address-batch-script/。

3.1.2 U-Boot MAC地址设置

在U-Boot中，可以在shell中手动设置MAC地址（注意Windows以太网驱动不会使用U-Boot中的MAC地址），或者在缺少设置的情况下启用随机MAC地址。

3.1.2.1 通过U-Boot变量手动设置MAC地址

在U-Boot shell中输入以下命令：

```
setenv ethaddr xx:xx:xx:xx -> 用于ENET
setenv eth1addr xx:xx:xx:xx -> 用于ENET_QOS
saveenv
```

3.1.2.2 启用随机MAC地址

在板级defconfig文件中添加`CONFIG_NET_RANDOM_ETHADDR=y`。

如果SROM和环境变量中都没有MAC地址，系统会报错。如果定义了`CONFIG_NET_RANDOM_ETHADDR`，使用随机分配的本地MAC地址。

了解详情，请访问恩智浦社区。

3.1.3 ACPI MAC地址设置

Windows驱动程序使用_DSM 方法从ACPI表中获取MAC地址。_DSM方法使用Dsdt-Platform.asl文件中定义的MC1X和MC2X，这些定义描述了MAC字节在熔丝中的存储位置：

```
OperationRegion(FUSE, SystemMemory,0x30350400,0x900) // 0x3035_0D00
Field(FUSE, AnyAcc, Nolock, Preserve)
{
 Offset(0x240),

 MC15, 8, // 0x640 NET1 MAC address bytes 5
 MC14, 8, // 0x641 NET1 MAC address bytes 4
 MC13, 8, // 0x642 NET1 MAC address bytes 3
 MC12, 8, // 0x643 NET1 MAC address bytes 2
 Offset(0x250),
 MC11, 8, // 0x650 NET1 MAC address bytes 1
 MC10, 8, // 0x651 NET1 MAC address bytes 0
 MC25, 8, // 0x652 NET2 MAC address bytes 5
 MC24, 8, // 0x653 NET2 MAC address bytes 4
 Offset(0x260),
 MC23, 8, // 0x660 NET2 MAC address bytes 3
 MC22, 8, // 0x661 NET2 MAC address bytes 2
 MC21, 8, // 0x662 NET2 MAC address bytes 1
 MC20, 8, // 0x663 NET2 MAC address bytes 0
}
```

然后，Dsdt-Enet.asl中的_DSM方法可以在需要时返回这些值：

```
// Function 1: Return Mac Address
case (1) {
 Store (MC10, MAC0)
 Store (MC11, MAC1)
 Store (MC12, MAC2)
 Store (MC13, MAC3)
 Store (MC14, MAC4)
 Store (MC15, MAC5)
 Return (MAC)
}
```

MC2X值用于第二个以太网接口（参见Dsdt-Enet_QoS.asl）。

如果MAC地址在熔丝中存储的顺序错误，可以在此处进行修补。

3.2 发送/接收延迟

对于ENET驱动程序，延迟设置可以在ACPI表中设置；而对于ENET_QOS，必须在Windows驱动程序代码中进行更改。

3.2.1 ENET发送延迟设置示例

以下是在ACPI中为i.MX 8M Nano设置延迟的示例：
mu_platform_nxp/NXP/MX8M_NANO_EVK/AcpiTables/Dsdt-Enet.asl

```
Name (_DSD, Package () {
 ToUUID("daffd814-6eba-4d8c-8a91-bc9bbf4aa301"),
 Package () { // ATHEROS AR8031>
   Package (2) {"MDIOBusController_InputClk_kHz", 266000},
   Package (2) {"PhyAddress", 0x00},
   Package (2) {"PhyInterafceType", 0x00}, // RGMII, default value
   Package (2) {"PhyMaxMDIOBusClock_kHz", 15000},
   Package (2) {"PhyMinSTAHoldTime_ns", 10},
   Package (2) {"PhyDisablePreamble", 0},
   Package (2) {"ConfigCmds", Package () {
        // Enable GTX_CLK delay
        MII_WRITE_COMMAND(MII_REG_AR8031_DP_ADDR, 0x0005),// Choose SerDes Test
 and System Mode Control
        MII_WRITE_COMMAND(MII_REG_AR8031_DP_RW, 0x0100),// Select 1 - RGMII Tx
 Clock Delay Enable
        /// Specific
        MII_WRITE_COMMAND(MII_REG_AR8031_SS, 0x000C),// Smart speed off
        ENET_MII_END}}
   }
})
```

3.2.2 在驱动程序中设置ENET_QOS发送延迟的示例

发送/接收延迟的设置位于`MII_Rtl8211fInit`函数中。

```
// Enable TX-delay for rgmii-id and rgmii-txid
 Val = MII_Read(pAdapter, PhyAddr, 0x11);
 if (pAdapter->MiiCfg.MiiInterfaceType == RGMII) {
     // RGMII config
     Val |= 0x0100;
 } else {
     Val &= ~0x0100;
 }
 MII_Write(pAdapter, PhyAddr, 0x11, Val);
 // Enable RX-delay for rgmii-id and rgmii-rxid
 Val = MII_Read(pAdapter, PhyAddr, 0x15);
 if (pAdapter->MiiCfg.MiiInterfaceType == RGMII) {
     // RGMII config
     Val |= 0x0008;
 } else {
    Val &= ~0x0008;
 }
 MII_Write(pAdapter, PhyAddr, 0x15, Val);
```

4 调试

本节提供了在目标板上调试PHY的帮助信息。

4.1 如何在目标板上启动内核调试

当以太网尚未工作时，必须使用串行调试。要通过串行端口启动内核调试，请执行以下步骤：

1. 在提升权限的命令窗口中输入以下命令，以在目标/开发板上启用内核调试：
   ```
   bcdedit /debug on
   bcdedit /dbgsettings serial debugport:3 baudrate:921600
   ```
   请使用适合您板卡设计的端口号。波特率必须与U-Boot defconfig文件中的`CONFIG_BAUDRATE`值匹配。

2. 在开发PC上提升权限的命令窗口中输入以下命令来启动WinDBG：
   `"C:\Program Files (x86)\Windows Kits\10\Debuggers\x64\windbg.exe" -k com:port=COM3,baud=921600`

4.2 如何在WinDbg中显示调试消息

要在WinDbg窗口中查看以太网驱动程序的调试消息，需要在驱动程序源代码中取消注释相关内容，并设置WinDbg调试打印过滤器。

4.2.1 在Windows驱动程序中启用调试消息

要启用调试消息：

1. 打开iMXPlatform项目。
2. 打开文件`imxnetmini->header files->mp_dbg.h`。通过取消注释或注释定义来启用/禁用所需的日志输出，例如，`//#define DBG_MDIO_DEV`

示例8。PHY调试的建议候选项

```
// ENET PHY device-specific macros - uncomment next line for message printing
//#define DBG_PHY_DEV
// MDIO bus-specific macros - uncomment next line for message printing
//#define DBG_MDIO_BUS
// MDIO device-specific macros - uncomment next line for message printing
//#define DBG_MDIO_DEV
// MDIO device command-specific macros - uncomment next line for message
 printing
//#define DBG_MDIO_DEV_CMD
```

4.2.2 在WinDbg中启用调试消息

通过在WinDbg中输入以下命令，为当前调试会话启用调试消息：

```
ed nt!Kd_IHVDRIVER_Mask 0xFFFFFFFF
```

也可以通过以下命令在Windows注册表中永久设置：

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Debug Print
Filter" /v IHVDRIVER /t REG_DWORD /d 0xFFFFFFFF
```

5 代码示例

针对检测TI DP83867扩展的Windows驱动程序函数

```
NTSTATUS MII_PhySpecificInit(PMP_ADAPTER pAdapter)
{
    NTSTATUS Status = STATUS_SUCCESS;
    switch (pAdapter->ENETDev_PHYDevice.PhyVendor)
        { case REALTEK:
            switch (pAdapter->ENETDev_PHYDevice.PhyModel)
                { case RTL8211F:
                case RTL8211F_VD_CG:
                    DBG_PHY_DEV_PRINT_INFO("Detected Realtek RTL8211F");
                    MII_Rtl8211fInit(pAdapter);
                break;
                default:
                    DBG_PHY_DEV_PRINT_WARNING("Unknown Realtek PHY Model: 0x
%02X", pAdapter->ENETDev_PHYDevice.PhyModel);
                    break;
            }
        break;
        case TEXAS_INSTRUMENTS:
            switch (pAdapter->ENETDev_PHYDevice.PhyModel)
            { case DP83867:
                DBG_PHY_DEV_PRINT_INFO("Detected TI DP83867");
                MII_DP83867fInit(pAdapter);
                break;
            default:
                DBG_PHY_DEV_PRINT_WARNING("Unknown TI PHY Model: 0x%02X",
 pAdapter->ENETDev_PHYDevice.PhyModel);
                break;
            }
            break;
        default:
            DBG_PHY_DEV_PRINT_WARNING("Unknown PHY vendor: 0x%02X", pAdapter-
>ENETDev_PHYDevice.PhyVendor);
            break;
    }
    return Status;
}
```

RTL8211的Windows驱动程序初始化函数

```
NTSTATUS MII_Rtl8211fInit(PMP_ADAPTER pAdapter)
{
    NTSTATUS Status = STATUS_SUCCESS;
    UINT16 Val;
    UINT8 PhyAddr = pAdapter->MiiCfg.PhyAddr;
    // Select Page 0x0d08*/
    MII_Write(pAdapter, PhyAddr, 0x1F, 0x0d08);
    // Enable TX-delay for rgmii-id and rgmii-txid
    Val = MII_Read(pAdapter, PhyAddr, 0x11);
    if (pAdapter->MiiCfg.MiiInterfaceType == RGMII) {
        // RGMII config
        Val |= 0x0100;
    } else {
        Val &= ~0x0100;
    }
    MII_Write(pAdapter, PhyAddr, 0x11, Val);
    // Enable RX-delay for rgmii-id and rgmii-rxid
    Val = MII_Read(pAdapter, PhyAddr, 0x15);
    if (pAdapter->MiiCfg.MiiInterfaceType == RGMII) {
        // RGMII config
        Val |= 0x0008;
    } else {
        Val &= ~0x0008;
    }
    MII_Write(pAdapter, PhyAddr, 0x15, Val);
    // Restore to default page 0
    MII_Write(pAdapter, PhyAddr, 0x1F, 0x0000);
    // Set green LED for Link, yellow LED for Active
    MII_Write(pAdapter, PhyAddr, 0x1F, 0x0D04);
    MII_Write(pAdapter, PhyAddr, 0x10, 0x617F);
    MII_Write(pAdapter, PhyAddr, 0x1F, 0x0000);
    return Status;
}
```

TI DP83867的Windows驱动程序初始化函数

```
NTSTATUS MII_DP83867fInit(PMP_ADAPTER pAdapter)
{
    NTSTATUS Status = STATUS_SUCCESS;
    UINT16 Val;
    UINT8 PhyAddr = pAdapter->MiiCfg.PhyAddr;
    // Select Page 0x0d08*/
    MII_Write(pAdapter, PhyAddr, 0x1F, 0x0d08);
    // Enable TX-delay for rgmii-id and rgmii-txid
    Val = MII_Read(pAdapter, PhyAddr, 0x11);
    if (pAdapter->MiiCfg.MiiInterfaceType == RGMII) {
        // RGMII config
        Val |= 0x0100;
    } else {
        Val &= ~0x0100;
    }
    MII_Write(pAdapter, PhyAddr, 0x11, Val);
    // Enable RX-delay for rgmii-id and rgmii-rxid
    Val = MII_Read(pAdapter, PhyAddr, 0x15);
    if (pAdapter->MiiCfg.MiiInterfaceType == RGMII) {
        // RGMII config
        Val |= 0x0008;
    } else {
        Val &= ~0x0008;
    }
    MII_Write(pAdapter, PhyAddr, 0x15, Val);
    // Restore to default page 0
    MII_Write(pAdapter, PhyAddr, 0x1F, 0x0000);
    // Set green LED for Link, yellow LED for Active
    MII_Write(pAdapter, PhyAddr, 0x1F, 0x0D04);
    MII_Write(pAdapter, PhyAddr, 0x10, 0x617F);
    MII_Write(pAdapter, PhyAddr, 0x1F, 0x0000);
    return Status;
}
```

6 关于本文中源代码的说明

本文中所示的示例代码具有以下版权和BSD-3-Clause许可：

2024年恩智浦版权所有；在满足以下条件的情况下，可以源代码和二进制文件的形式重新分发和使用本源代码（无论是否经过修改）：

1. 重新分发源代码必须保留上述版权声明、这些条件和以下免责声明。
2. 以二进制文件形式重新分发时，必须在文档和/或随分发提供的其他材料中复制上述版权声明、这些条件和以下免责声明。
3. 未经事先书面许可，不得使用版权所有者的姓名或参与者的姓名为本软件的衍生产品进行背书或推广。

本软件由版权所有者和参与者“按原样”提供，不承担任何明示或暗示的担保责任，包括但不限于对适销性和特定用途适用性的暗示保证。在任何情况下，无论因何种原因或根据何种法律条例，版权所有者或参与者均不对因使用本软件而导致的任何直接、间接、偶然、特殊、惩戒性或后果性损害（包括但不限于采购替代商品或服务；使用损失、数据损失或利润损失或业务中断）承担责任，无论是因合同、严格责任还是侵权行为（包括疏忽或其他原因）造成的，即使事先被告知有此类损害的可能性也不例外。

7 修订历史

表1. 修订历史

文档ID	发布日期	描述
AN14188 v.1.0	2024年5月27日	初始版本

Legal information

Definitions

Draft — A draft status on a document indicates that 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 in a draft version of a document and shall have no liability for the consequences of use of such information.

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.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com.cn/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

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.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

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

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

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

NXP — wordmark and logo are trademarks of NXP B.V.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Freescale — is a trademark of NXP B.V.

i.MX — is a trademark of NXP B.V.

Microsoft, Azure, and ThreadX — are trademarks of the Microsoft group of companies.

目录

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

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

Document feedback

Date of release: 27 May 2024
Document identifier: AN14188