文档编号：RTEYOCTOUG
第2.0版，2021年7月29日

实时边缘Yocto Project用户指南

第1章
概述

本文档介绍如何使用Yocto Project构建环境为i.MX和QorIQ(Layerscape)板构建实时边缘镜像。它介绍了实时边缘软件Yocto层及其用法。

Yocto Project是一个专注于嵌入式Linux®操作系统开发的开源协作项目。有关Yocto Project的更多信息，请参见Yocto Project页面：www.yoctoproject.org。Yocto Project主页上有几个文档详细描述了如何使用该系统。要使用没有实时边缘发布层的基本Yocto Project，请按照www.yoctoproject.org/docs/current/yocto-project-qs/yocto-project-qs.html中的《Yocto Project快速入门》中的说明进行操作。

实时边缘层基于i.MX Yocto Project和LSDK Yocto版本。

- i.MX Yocto Project提供i.MX板支持。有关更多信息，请参见i.MX Yocto Project用户指南.pdf。
- LSDK Yocto Project提供Layerscape板支持。有关更多信息，请参见https://www.nxp.com.cn/ docs/en/user-guide /LSDKYOCTOUG.pdf。

用于构建镜像的文件存储在层中。层包含不同类型的自定义，并且来自不同的来源。层中的一些文件称为要素。Yocto Project要素包含检索源代码、构建和打包组件的机制。以下列表显示了此版本中使用的层。

实时边缘层

- **动态层：**包括i.MX和Layerscape相关要素的更新。

```
├── imx-layer
└── qoriq-layer
```

- **要素-扩展：**包括实时网络、实时系统和工业的要素。
- **要素-恩智浦：**实时边缘镜像要素

注意

1.1 最终用户许可协议

在实时边缘Yocto Project社区BSP的设置环境过程中，会显示恩智浦最终用户许可协议(EULA)。要继续使用实时边缘专有软件，用户必须同意本许可的条件。同意条款即允许Yocto Project构建解压包。

1.2 相关文档

- 有关i.MX Yocto Project的更多信息，请参见i.MX Yocto Project用户指南.pdf。
- 有关LSDK Yocto Project的更多信息，请参见https://www.nxp.com.cn/docs/en/user-guide/LSDKYOCTOUG.pdf。
- 有关实时功能的更多信息，请参见URL http:// nww.preview.nxp.com/design/software/development-software/real-time-edge-software:REALTIME-EDGE-SOFTWARE上的*实时边缘软件用户指南*。
- 有关烧写I.MX板的SD卡镜像，请参见i.MX Linux®用户指南(IMXLUG)。

第2章
功能

实时边缘Yocto Project具有以下特点：

- Linux内核：
  - 内核要素包含两个文件夹：
    - `dynamic-layers/imx-layer/recipes-kernel`：用于i.MX板的Linux
    - `dynamic-layers/qoriq-layer/recipes-kernel`：用于Layerscape板的Linux
  - `Linux 5.10.9-rt24`是为Yocto Project发布的Linux内核的基础。
- U-Boot
  - U-Boot要素有两个文件夹：
    - `dynamic-layers/imx-layer/recipes-bsp/u-boot/`：用于i.MX板的U-Boot。
    - `dynamic-layers/qoriq-layer/recipes-bsp/u-boot/`：用于Layerscape板的U-Boot。
  - `U-Boot 2020.04`是为Yocto Project发布的U-Boot基础。
  - `U-boot-script-distro boot`要素为普通和裸机镜像提供了发行版引导脚本。
- 实时网络要素：
  - `avahi`
  - `iproute2`
  - `genavb-tsn`
  - `libredblack`
  - `libyang`
  - `lldpd`
  - `linuxptp`
  - `netopeer2-cli`
  - `netopeer2-keystored`
  - `netopeer2-server`
  - `real-time-edge-nodejs-lbt`
  - `real-time-edge-prl`
  - `real-time-edge-sysrepo`
  - `sysrepo`
  - `tsn-scripts`
  - `tsntool`
- 实时系统要素：
  - `real-time-edge-baremetal`：实时边缘裸机要素位于`recipes-extended`目录中。它提供在Slave内核上运行的裸机二进制文件。
  - `real-time-edge-icc`：icc要素位于`recipes-extended`目录中。它为Master/Slave和Slave/Slave内核之间的社区提供了一个工具。
  - `jailhouse`

- 协议要素：
  - canfestival
  - igh-ethercat
  - libnfc-nci
  - libopen62541
  - real-time-edge-libbee
  - real-time-edge-libblep

第3章
主机设置

要在Linux主机上设置具有预期行为的Yocto Project，必须安装以下部分中所述的包和实用程序。

一个重要的考量因素是主机所需的硬盘空间。例如，在运行Ubuntu的机器上构建时，所需的最小硬盘空间约为50 GB。建议至少提供120 GB的磁盘空间，这才足以将所有后端编译在一起。

建议的最低Ubuntu版本为18.04或更高版本。Chromium版本74需要Ubuntu 18.04。最新版本支持Chromium V74，需要将ulimit（打开文件数）增加到4098。如果不使用Chromium，18.04应该可以工作。

注意

16.04之前的早期版本可能会导致Yocto Project构建设置失败，因为它需要从Ubuntu 12.04开始才可用的python版本。有关更多信息，请参见Yocto Project参考手册。

Ubuntu 16.04用户注释了为SDL构建期间发生的错误。要解决此问题，请在`local.conf`中注释掉以下几行，例如添加`#`字符：

```
#PACKAGECONFIG_append_pn-qemu-native = " sdl"
#PACKAGECONFIG_append_pn-nativesdk-qemu = " sdl"
```

3.1 主机包

Yocto Project构建需要为构建安装一些包。这些包记录在Yocto Project下。转至Yocto Project快速入门，查看必须为您的构建机器安装的包。

基本的Yocto Project主机包是：

```
$ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib \
build-essential chrpath socat cpio python python3 python3-pip python3-pexpect \
xz-utils debianutils iputils-ping python3-git python3-jinja2 libegl1-mesa \
libsdl1.2-dev pylint3 xterm rsnc curl
```

配置工具使用构建机器上的默认grep版本。如果您的路径中有不同版本的grep，则可能会导致构建失败。一种解决方法是将特殊版本重命名为不包含“grep”的名称。

3.2 设置repo实用程序

“Repo”是一个基于Git的工具，可以更轻松地管理包含多个存储库的项目，这些存储卡无需位于同一服务器上。Repo很好地补充了Yocto Project的分层特性，使用户可以更轻松地将自己的层添加到板支持包(BSP)。

要安装“repo”实用程序，请执行以下步骤：

1. 在主目录中创建一个`bin`文件夹。

```
$ mkdir ~/bin (this step may not be needed if the bin folder already exists)
$ curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
$ chmod a+x ~/bin/repo
```

2. 将以下行添加到`.bashrc`文件，以确保`~/bin`文件夹位于您的`PATH`变量中。

```
$ export PATH=~/bin:$PATH
```

第4章
Yocto Project设置

首先，确保使用以下命令正确设置了git：

```
$ git config --global user.name "Your Name"
$ git config --global user.email "Your Email"
$ git config –list
```

实时边缘Yocto Project版本目录包含一个源目录。该目录包含用于构建一个或多个构建目录的要素，以及一组用于设置环境的脚本。

用于构建项目的要素来自社区和实时边缘。Yocto Project层下载到sources目录。在此目录中，设置用于构建项目的要素。

以下示例显示如何下载实时边缘要素层。对于此示例，为项目创建了一个名为yocto-real-time-edge的目录。也可以使用任何其他名称来代替此名称。

```
$ mkdir yocto-real-time-edge
$ cd yocto-real-time-edge
$ repo init -u https://github.com/real-time-edge-sw/yocto-real-time-edge.git -b real-time-edge-
gatesgarth -m real-time-edge-2.0.0.xml
$ repo sync
```

此过程完成后，将源代码迁出到目录yocto-real-time-edge/sources中。您可以使用命令repo sync定期执行repo同步，以更新到最新的代码。

如果在repo初始化过程中出现错误，请尝试删除.repo目录并再次运行repo初始化命令。

repo init针对该行中的最新补丁进行了配置。

第5章
镜像构建

本节提供了详细信息以及构建镜像的过程。

5.1 构建配置

实时边缘提供了脚本real-time-edge-setup-env.sh，简化了i.MX和Layerscape机器的设置。要使用该脚本，必须指定要对其进行构建的特定机器的名称和所需的发行版。该脚本为指定的机器和发行版设置目录和配置文件。

实时边缘支持以下恩智浦硬件平台。

- imx6ull14x14evk
- imx8mmevk
- imx8mpevk
- ls1028ardb
- ls1012ardb
- ls1021atwr
- ls1021atsn
- ls1021aiot
- ls1043ardb
- ls1046ardb
- ls1046afrwy
- lx2160ardb
- lx2160ardb-rev2

每个构建文件夹都必须配置为仅使用一个发行版。每次更改DISTRO_FEATURES变量时，都需要一个干净的构建文件夹。发行版配置保存在DISTRO设置的local.conf文件中，并在bitbake运行时显示。以下是DISTRO配置列表：

- nxp-real-time-edge——正常镜像包括实时和工业包，不支持裸机。
- nxp-real-time-baremetal——裸机镜像（某些板不支持此发行版）。

real-time-edge-setup-env.sh脚本的语法如下所示：

```
$ DISTRO=<distro name> MACHINE=<machine name> source real-time-edge-setup-env.sh -b <build dir>
```

DISTRO=<发行版配置名称>是发行版，它配置构建环境，存储在meta-real-time- edge/conf/distro中。

MACHINE=<机器配置名称>是机器名称，它指向meta-freescale和meta-imx中conf/machine中的配置文件。

-b <build dir>指定由real-time-edge-setup-env.sh脚本创建的构建目录的名称。

当脚本运行时，它会提示用户接受最终用户许可协议(EULA)。接受EULA后，接受函存储在每个构建文件夹内的local.conf中，并且不再显示该构建文件夹的EULA接受函查询。

脚本运行后，工作目录就是脚本刚刚创建的目录，使用-b选项指定。创建一个conf文件夹，其中包含文件bblayers.conf和local.conf。

<build dir>/conf/bblayers.conf文件包含实时边缘Yocto Project版本中使用的所有元层。local.conf文件包含机器和发行版规范：

```
MACHINE ??= 'imx8mpevk'
DISTRO ?= 'nxp-real-time-edge'

ACCEPT_FSL_EULA = "1"
```

如有必要，可以通过编辑此文件来更改MACHINE配置。

local.conf文件中的ACCEPT_FSL_EULA表示您已接受EULA的条件。

5.2 选择实时边缘Yocto Project镜像

Yocto Project提供了在不同层上可用的镜像。

实时边缘提供了nxp-image-real-time-edge镜像，其中包含Real-time Networking、Real-time System和Protocols包。

5.3 构建镜像

Yocto Project构建使用bitbake命令。例如，bitbake <component>构建指定的组件。每个组件构建都有多个任务，例如获取、配置、编译、打包和部署到目标rootfs。bitbake镜像构建收集镜像所需的所有组件，并按照每个任务的依赖关系顺序构建。第一个构建是工具链以及构建组件所需的工具。

以下命令是如何构建镜像的示例：

```
$ bitbake nxp-image-real-time-edge
```

5.4 Bitbake选项

用于构建镜像的bitbake命令是bitbake <image name>。附加参数可用于下述特定活动。Bitbake为开发单个组件提供了各种有用的选项。如需使用bitbake参数运行，要使用的命令如下：

```
$ bitbake <parameter> <component>
```

<component>是所需的构建包。

下表提供了一些bitbake选项。

表1. Bitbake选项

Bitbake参数	说明
-c fetch	如果下载状态未标记为完成，则获取。
-c cleanall	清理整个组件构建目录。构建目录中的所有更改都丢失。组件的根文件系统和状态也被清除。该组件也会从下载目录中移除。
-c deploy	将镜像或组件部署到根文件系统。
-k	即使发生构建中断，也会继续构建组件。
-c compile -f	不建议更改临时目录下的源代码。但是，如果已更改，除非使用此选项，否则Yocto Project可能不会重建它。使用此选项可在部署镜像后强制重新汇编。

表格接下页……

表1. Bitbake选项（续）

-g	列出镜像或组件的依赖关系树。
-DDD	打开3级深度调试。每个D都增加一个调试级别。

5.5 构建方案

以下是各种配置的构建设置方案。

使用以下命令设置清单并填充Yocto Project层源：

```
$ mkdir yocto-real-time-edge
$ cd yocto-real-time-edge
$ repo init -u https://github.com/real-time-edge-sw/yocto-real-time-edge.git -m default.xml
$ repo sync
```

以下部分给出了一些具体的例子。替换指定的机器名称和后端，以自定义命令。

5.5.1 i.MX 8M Plus EVK上的实时边缘镜像

这将使用实时边缘包构建多媒体镜像。可以使用以下命令：

```
$ DISTRO=nxp-real-time-edge MACHINE=imx8mpevk source real-time-edge-setup-env.sh -b build-imx-real-time-edge
$ bitbake nxp-image-real-time-edge
```

5.5.2 i.MX 8M Plus EVK上的实时边缘裸机镜像

```
$ DISTRO=nxp-real-time-edge-baremetal MACHINE=imx8mpevk source real-time-edge-setup-env.sh -b build-imx-baremetal
$ bitbake nxp-image-real-time-edge
```

1. 重新启动构建环境

有时，会打开一个新的终端窗口或在设置构建目录后重新启动机器。在这些情况下，应该使用设置环境脚本来设置环境变量并再次运行构建。不需要完整的real-time-edge-setup-release.sh。

```
$ source setup-environment <build-dir>
```

第6章
镜像部署

完整的文件系统镜像部署到<build directory>/tmp/deploy/images。在大多数情况下，镜像特定于环境设置中设置的机器。每个镜像构建都会根据机器配置文件中定义的IMAGE_FSTYPES创建一个U-Boot、一个内核和一个镜像类型。大多数机器配置都提供SD卡镜像(.wic)和根文件系统镜像(.tar)。SD卡镜像包含适合引导相应硬件的分区镜像（带有U-Boot、内核、根文件系统和其他此类文件）。

6.1 烧写SD卡镜像

SD卡镜像文件“.wic”包含适合引导相应硬件的分区镜像（带有U-Boot、内核、根文件系统和其他文件）。要烧写SD卡镜像，请运行以下命令：

```
$ bunzip2 -dk -f ~<image_name>.wic.bz2
$ sudo dd if=<image name>.wic of=/dev/sd<partition> bs=1M conv=fsync
```

对于i.MX，请参见i.MX Linux®用户指南(IMXLUG)中的“准备要启动的SD/MMC卡”部分。

对于Layerscape，请参见LSDKYOCTOUG.pdf。

第7章
基于i.MX Yocto版本构建包

本章介绍如何将meta-real-time-edge包添加到i.MX Yocto Project中。

表2. i.MX Yocto Project上的选定包

包	要素	实时边缘Linux	i.MX Linux
IGH EtherCAT主栈	igh-ethercat	是	是
LinuxPTP	linuxptp	是	是
OPC-UA	libopen62541	是	是
real-time-edge-sysrepo	real-time-edge-sysrepo	是	否
Jailhouse	jailhouse	是	是
实时边缘裸机		是	否
抢占RT Linux		是	否

7.1 下载i.MX Yocto版本和实时边缘Yocto层

安装i.MX Yocto Project，参见用户指南：

1. 下载i.MX Yocto版本：

```
$ mkdir imx-yocto-bsp
$ cd imx-yocto-bsp
$ repo init -u https://source.codeaurora.org/external/imx/imx-manifest -b imx-linux-gatesgarth
-m imx-5.10.9-1.0.0.xml
$ repo sync
```

2. 下载实时边缘Yocto层：

```
$ cd sources
$ git clone https://github.com/real-time-edge-sw/meta-real-time-edge.git
```

在i.MX镜像构建中使能meta-real-time-edge层

1. 设置构建环境：

```
$ cd imx-yocto-bsp (The top directory of repo)
$ DISTRO=fsl-imx-wayland MACHINE=<machine name> source imx-setup-release.sh -b build-real-time-edge
```

2. 将meta-real-time-edge添加到特定构建文件夹下的bblayers.conf中

```
$ cd build-real-time-edge (The build-directory)
$ vim conf/bblayers.conf
# Add the below setting
BBLAYERS += "${BSPDIR}/sources/meta-real-time-edge"
```

7.2 选择实时边缘Yocto层的包

7.2.1 来自实时边缘Yocto层的包

来自实时边缘Yocto层的包可以单独添加到i.MX镜像中。这些包如下：

- igh_ethercat
- real-time-edge-sysrepo
- libopen62541 (OPC UA)

要选择包，请按如下所示将其添加到`local.conf`上的`IMAGE_INSTALL`中：

例如，使用以下命令添加`igh-ethercat`包：

添加igh-ethercat：

```
$ vim conf/local.conf

# Add package
IGH_ETHERCAT_imx8mmevk = " fec "
IGH_ETHERCAT ??= " "
PACKAGECONFIG_append_pn-igh-ethercat = " ${IGH_ETHERCAT} "
IMAGE_INSTALL += " igh_ethercat "
```

注意

对于i.MX Yocto版本，FEC以太网驱动程序内置在内核中，只能使用EtherCAT通用模块。为了在i.MX 8M Mini平台上使用原生支持EtherCAT的模块，用户需要通过在内核配置中设置“`CONFIG_FEC=m`”将FEC以太网驱动程序汇编为模块，并将`DEVICE_MODULES`设置为“`fec`”，如《实时边缘软件用户指南》的第`5.1.5.2章 IGH EtherCAT`设置中所述。

添加OPC UA

```
$ vim conf/local.conf
# Add package
# Select OPC UA example application
include ${BSPDIR}/sources/meta-real-time-edge/conf/distro/include/libopen62541.inc
LIBOPEN62541_LOGLEVE = "300"
IMAGE_INSTALL += " libopen62541"
```

7.2.2 i.MX Yocto层中的包

添加`meta-real-time-edge`层时，将覆盖i.MX Yocto层中的包。如果需要保留原始包而不是使用实时边缘包，则必须将这些包添加到`bblayer.conf`中的“`BBMASK`”中，如下所示。

- avahi
- ethtool
- iproute2
- jailhouse
- linuxptp
- lldpd
- tsntool

以下是可用于屏蔽`bblayer.conf`中的包的配置：

```
$ vim conf/bblayers.conf
# Add the below setting
```

```
BBMASK += "meta-real-time-edge/recipes-extended/jailhouse/*.bbappend"
BBMASK += "meta-real-time-edge/recipes-extended/tsntool/tsntool_%.bbappend"
BBMASK += "meta-real-time-edge/recipes-extended/ethtool/ethtool_%.bbappend"
BBMASK += "meta-real-time-edge/recipes-extended/linuxptp/linuxptp_3.1.bbappend"
BBMASK += "meta-real-time-edge/recipes-extended/avahi/avahi_%.bbappend"
BBMASK += "meta-real-time-edge/recipes-extended/iproute2/iproute2_%.bbappend"
BBMASK += "meta-real-time-edge/recipes-extended/lldpd/lldpd_%.bbappend"
```

可以选择以上包在i.MX Yocto层上运行。

例如：

```
$ vim conf/local.conf

# Add package
IMAGE_INSTALL += " \
     linuxptp \
     jailhouse \
     iproute2 \
     lldpd \
     avahi-daemon \
     avahi-utils \
"
```

7.3 构建镜像

添加包后，用户可以开始使用选定的实时边缘包构建镜像。

使用以下命令构建i.MX镜像：

```
$ bitbake imx-image-multimedia
```

7.4 在i.MX版本上运行包

1. 运行Jailhouse

   请参见《实时边缘软件用户指南》中的第3.3.2章 在Inmate中运行PREEMPT_RT Linux和第3.3.3章 在Inmate中运行Jailhouse示例。

2. 运行LinuxPTP

   请参见《实时边缘软件用户指南》中的第4.3.5章 IEEE 1588的快速入门和第4.3.6章 IEEE 802.1AS的快速入门。

3. 运行IGH-EtherCAT

   请参见《实时边缘软件用户指南》中的第5.1.5.2章 IGH EtherCAT设置。

4. 运行OPC-UA

   请参见《实时边缘软件用户指南》中的第5.3章 OPC UA。

第8章
基于Layerscape Yocto版本构建包

本章介绍如何将meta-real-time-edge包添加Layerscape Yocto Project中。

表3. Layerscape Yocto Project上的选定包

	要素	实时边缘Linux	Layerscape Linux
IGH EtherCAT主堆栈	igh-ethercat	是	是
LinuxPTP	linuxptp	是	是
OPC-UA	libopen62541	是	是
real-time-edge-sysrepo	real-time-edge-sysrepo	是	是
Jailhouse	jailhouse	是	是
实时边缘裸机		是	是
抢占RT Linux		是	是

8.1 下载LSDK Yocto版本和实时边缘Yocto层

1. 使用以下命令下载LSDK Yocto版本：

```
$ mkdir yocto-sdk
$ cd yocto-sdk
$ repo init -u https://source.codeaurora.org/external/qoriq/qoriq-components/yocto-sdk -b
gatesgarth
$ repo sync
```

2. 使用以下命令下载实时边缘Yocto层：

```
$ cd sources
$ git clone https://github.com/real-time-edge-sw/meta-real-time-edge.git
```

8.2 在Layerscape镜像构建中使能meta-real-time-edge层

1. 设置构建环境：

```
$ . ./setup-env -m ls1028ardb
```

2. 将`meta-real-time-edge`添加到特定构建文件夹下的`bblayers.conf`中。

```
$ vim conf/bblayers.conf
# Add the below setting
BBLAYERS += " ${TOPDIR}/../sources/meta-real-time-edge"
```

8.3 选择实时边缘Yocto层的包

8.3.1 来自实时边缘Yocto层的包

实时边缘Yocto层中包含的包可以单独添加到Layerscape镜像中。这些包是：

- igh_ethercat
- real-time-edge-sysrepo
- libopen62541 (OPC UA)

要选择包，请将它们添加到local.conf上的IMAGE_INSTALL中，如下所示：

添加igh-ethercat：

```
$ vim conf/local.conf

# Add package
IGH_ETHERCAT  ??=  " "
IMAGE_INSTALLL_append = " igh_ethercat "
```

添加real-time-edge-sysrepo

```
$ vim conf/local.conf
# Add package
REAL_TIME_EDGE_SYSREPO_ls1028ardb = ""
REAL_TIME_EDGE_SYSREPO_ls1021atsn = "real-time-edge-sysrepo-tc"
PACAGECONFIG_append_pn-real-time-edge-sysrepo = "${REAL_TIME_EDGE_SYSREPO}" IMAGE_INSTALL_append = "
real-time-edge-sysrepo "
```

添加OPC UA

```
$ vim conf/local.conf
# Add package
# Select OPC UA example application
include ../sources/meta-real-time-edge/conf/distro/include/libopen62541.inc
LIBOPEN62541_LOGLEVE = "300" IMAGE_INSTALL_append = " libopen62541"
```

8.3.2 Layerscape Yocto层中的包

添加meta-real-time-edge层时，将覆盖Layerscape Yocto层中的以下包。

- avahi
- ethtool
- iproute2
- jailhouse
- linuxptp
- lldpd
- tsntool

如果您需要保留原始包而不是使用实时边缘包，请将这些包添加到bblayer.conf文件中的“BBMASK”中，如下所示。

```
$ vim conf/bblayers.conf
# Add the below setting
BBMASK += "meta-real-time-edge/recipes-extended/jailhouse/*.bbappend"
BBMASK += "meta-real-time-edge/recipes-extended/tsntool/tsntool_%.bbappend"
BBMASK += "meta-real-time-edge/recipes-extended/ethtool/ethtool_%.bbappend"
BBMASK += "meta-real-time-edge/recipes-extended/linuxptp/linuxptp_3.1.bbappend"
BBMASK += "meta-real-time-edge/recipes-extended/avahi/avahi_%.bbappend"
BBMASK += "meta-real-time-edge/recipes-extended/iproute2/iproute2_%.bbappend"
BBMASK += "meta-real-time-edge/recipes-extended/lldpd/lldpd_%.bbappend"
```

基于Layerscape Yocto版本构建包

可以选择以上包在Layerscape Yocto层上运行。要选择包，需要将包名称添加到“IMAGE_INSTALL”中。

例如：

```
$ vim conf/local.conf

# Add package
IMAGE_INSTALL_append = " \
     linuxptp \
     jailhouse \
     iproute2 \
     lldpd \
     avahi-daemon \
     avahi-utils \
"
```

8.4 构建镜像

添加包后，可以开始使用选定的实时边缘包构建镜像。

使用以下命令构建Layerscape镜像：

```
$ bitbake fsl-image-networking
```

8.5 在Layerscape上运行包

1. 运行Jailhouse

   以下流程以LS1028ARDB为例。

   a. 从实时边缘版本中获取fsl-ls1028a-rdb-jailhouse.dtb。 其他镜像按照上述流程构建。

   b. 在U-Boot下运行以下命令来启动板。

   ```
   # setenv bootargs "root=/dev/ram0 rw earlycon=uart8250,0x21c0500
   console=ttyS0,115200 ramdisk_size=0x10000000"
   # tftp 0x82000000 Image; tftp 0xa0000000 fsl-image-networking-ls1028ardb.ext2.gz.uboot;
   tftp 0x90000000 fsl-ls1028a-jailhouse-rdb.dtb;
   # booti 0x82000000 0xa0000000 0x90000000;
   ```

   c. 将Linux内核二进制“镜像”传输到文件夹/usr/share/jailhouse/inmates/kernel/。

   有关其他步骤，请参见《实时边缘软件用户指南》中的第3.3.2章 在Inmate中运行PREEMPT_RT Linux和第3.3.3章 在Inmate中运行Jailhouse示例。

2. 运行LinuxPTP

   请参见《实时边缘软件用户指南》的第4.1.5章 IEEE 1588的快速入门和第4.1.6章 IEEE 802.1AS的快速入门。

3. 运行IGH-EtherCAT

   请参见《实时边缘软件用户指南》的第5.1.5.2章 IGH EtherCAT设置。

4. 运行OPC-UA

   请参见《实时边缘软件用户指南》的第5.3章 OPC UA。

第9章
修订记录

下表显示了对本文档的修订。

表4. 修订记录

日期	版本	对照检索	更改
2021年7月30日	2.0	-	实时边缘软件第2.0版的首版

How To Reach Us

Home Page:
nxp.com

Web Support:
nxp.com/support

Limited warranty and liability — Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

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.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. 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.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 29-Jul-2021
Document identifier: RTEYOCTOUG

arm