

# 天贝物联 ODM 项目 docker 镜像使用说明书

## 目录

一、开发流程.....	2
二、docker 镜像部署.....	3
三、 进行项目开发.....	4
四、示例.....	4

# 一、开发流程

开发流程包括：准备 docker 镜像运行环境（详见附录一）；部署 docker 镜像；进行项目开发。具体流程参见图 1。



图 1 用户开发流程图

## 二、docker 镜像部署

镜像的导入和使用：

1、下载 tenbay 提供的镜像

```
sudo docker pull tenbaydocker/ubuntu
```

```
sudo docker images
```

查看 **tenbaydocker/ubuntu** 是否已经在本地镜像列表里面(详见示例)

2、使用镜像： **sudo docker run (或者也可以使用 docker-compose)**

例 如： **sudo docker run --name ubuntu.image -p 5802:22 -dit -v /data1/share/image:/work tenbaydocker/ubuntu:latest /usr/sbin/sshd -D**

--name 为容器指定一个名称，本例中名称为 Ubuntu.image

-p 选项进行端口映射，本例中将容器的端口 5802 映射到主机的 22 端口

-d: 后台运行容器，并返回容器 ID；

-i: 以交互模式运行容器，通常与 -t 同时使用；

-t: 为容器重新分配一个伪输入终端，通常与 -i 同时使用；

-v 选项将主机的目录映射到容器的目录，本例中将主机的目录/data1/share/image 映射到容器的 work 目录

3、通过 **sudo docker exec -it ubuntu.image bash** 进入镜像

进入镜像后，就是环境的构建

**环境的构建：**

1、进入开发主机工作目录；

2、执行命令 **sh -x /usr/sbin/tools.sh** --目的是安装编译环境需要的工具和命令，需要时间比较长，大概需要一个小时；

3、**git clone [-b] [branch] url** 克隆自己的编译环境

(如果需要基础代码上进行进一步定制，请新建分支，在这个分支上开发)

注意事项：如果 clone 时不能拉取下来代码，请确认 ①gitlab 账号是否是报备给天贝的账号，必须使用报备给天贝公司的账号才能获取代码；②网络环境是否能够访问 [www.gitlab.com](http://www.gitlab.com) 网站，如果不能访问 gitlab 网站，也是不能获取代码的；

4、cd 进入编译根目录；

```
cd lede-source
```

5、下载安装 feeds 扩展程序代码

```
./script/feeds update -a
```

```
./script/feeds install -a
```

```
sh update_wireless.sh
```

注意事项：如果 **update** 时不能更新下来代码，请确认

①gitlab 账号是否是报备给天贝的账号,必须使用报备给天贝公司的账号才能获取代码;

②网络环境是否能够访问 [www.gitlab.com](http://www.gitlab.com) 网站,如果不能访问 gitlab 网站,也是不能获

6、**make menuconfig** 选择自己硬件对应的架构，并选择自己需要的软件，保存，退出

**make V=s** 开始编译;

注意事项:

① 如果遇到如图的错误，请在 `/etc/profile` 中添加一行命令：`export`

FORCE\_UNSAFE\_CONFIGURE=1, 然后保存执行 `source /etc/profile`, 操作完后

再次编译：

```
checking whether mkfifo rejects trailing slashes... yes
checking whether mknod can create fifo without root privileges... configure: error: in `/stuffs/lk/ava Beijing/lede-source/build dir/h
```

②如果遇到下载 dl 的软件包时出现下载失败，请联系天贝

③天贝提供一个基础的 `xxx.config` 可以成功编译出一个基础包;此 `xxx.config` 在第一

次执行 `update_wireless.sh` 脚本时会被覆盖为 `.config`，可以直接开始 `make`，如果开

发人员手动 `make menuconfig` 并保存后, 会覆盖该 `.config`, 不过可以再次从 `git` 仓库

直接获取该文件;

7、最终的固件包在 **bin/target/**目录下；

8、一切解释权，归 Tenbay 所有；

### 三、进行项目开发

修改源代码进行开发。

#### 四、示例

小明需要用到我司镜像完成自己项目软件的开发，编译等；当小明拿到了 **Tenbay** 提供的镜像后，在一个安装了 **docker** 的环境（我们认为是一个 **ubuntu** 的虚拟机）中进行了如下几步操作后，就可以正常开始自己的工作：

1、在 docker hub 上下载 docker 镜像；

2、**sudo docker images** 查看 tenbay 镜像已经存在本机镜像中；

3、使用 tenbay 镜像运行一个容器，例如使用命令：**sudo docker run --name ubuntu.test -p 5801:22 -d -v /data/share:/work tenbaydocker/ubuntu:latest env LANG=C.UTF-8 /usr/sbin/sshd -D** 此时可以使用 ssh 命令通过 5801 端口进入容器开始工作；

4、**ssh root@192.168.0.1 5801** 进入容器（默认密码是 123456），**cd /work**

5、执行 tool.sh 脚本完成必要的命令安装：**sh -x /usr/sbin/tool.sh**

6、**git clone -b master https://gitlab.com/lede-source.git;**

7、**cd lede-source**

8、**./script/feeds update -a**

**./script/feeds install -a**

**Sh update\_wireless.sh**

9、到此，完成了使用 tenbay 镜像构建编译环境的工作，小明可以按照自己的需求开始自己的操作了；

10、如果直接使用我司提供的代码即可满足自己的需求，那么直接使用该代码进行编译出包即可，如果需要进一步修改我们的代码，则使用我司提供的代码仓库，fork 一份在进行开发自己的功能。