
Ford Motor (China) Company

SDL Develop Environment

Version <1.0>

免责声明

本档中的内容仅供参考。福特汽车中国对本服务内容的错误或遗漏概不负责。在任何情况下，福特汽车中国均不对因使用本档而产生或与之相关的任何特殊，直接，间接，间接或偶然的损害赔偿或任何损害负责，无论是在合同，疏忽，其他侵权行为中服务或服务的内容。福特汽车中国保留随时对本档内容进行补充，删除或修改的权利，恕不另行通知。

Disclaimer

The contents contained in this document are for general information purposes only. Ford Motor China assumes no responsibility for errors or omissions in the contents on the Service.

In no event shall Ford Motor China be liable for any special, direct, indirect, consequential, or incidental damages or any damages whatsoever, whether in an action of contract, negligence or other tort, arising out of or in connection with the use of the Document or the contents of the Document.

Ford Motor China reserves the right to make additions, deletions, or modification to the contents on the Service at any time without prior notice.



本作品采用[知识共享署名-相同方式共享 4.0 国际许可协议](https://creativecommons.org/licenses/by-sa/4.0/)进行许可。

This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

编制	Beyondsoft	日期	2017-11-14	版权	署名-相同方式共享 4.0 国际
审核	Ford	日期	2017-11-14	管理	Ford

Project Name: sdl_core	Version: <1.0>
SDL_Develop_Environment	Date: <14/11/2017>
<document identifier>	

修改历史

版本	日期	说明
1.0.0	2017-11-14	初版，完成文档架构，填写文档内容

Project Name: sdl_core	Version: <1.0>
SDL_Develop_Environment	Date: <14/11/2017>
<document identifier>	

目录

1	SYSTEM REQUIREMENT	2
2	INSTALL COMPILER AND DEVELOP TOOLS.....	2
2.1	For x86 desktop:.....	2
2.2	For armhf desktop:	2
3	INSTALL SUPPORT PACKAGES FOR 32-BIT ARCHITECTURE.....	2
4	INSTALL ALL THE PACKAGES FOR SDL DEPENDENCE	2
4.1	For x86 desktop:.....	2
4.2	For armhf desktop:	2
5	BUILD AND RUN.....	3
5.1	Get source code.	3
5.2	cmake && make && make install	3
5.2.1	For x86 desktop:.....	3
5.2.2	For armhf desktop:	3



Project Name: sdl_core	Version: <1.0>
SDL_Develop_Environment	Date: <14/11/2017>
<document identifier>	

1 System requirement

Your operating system should be Ubuntu16.04 x64, other OS may be not compatible.

2 Install compiler and develop tools.

2.1 For x86 desktop:

```
$sudo apt install git cmake pkg-config build-essential
```

2.2 For armhf desktop:

```
$sudo apt install git cmake pkg-config gcc-arm-linux-gnueabi  
g++-arm-linux-gnueabi
```

3 Install support packages for 32-bit architecture

```
$sudo apt-get install lib32ncurses5 lib32z1 lib32stdc++6
```

4 Install all the packages for sdl dependence

4.1 For x86 desktop:

```
$sudo apt install libudev-dev libsqlite3-dev libssl-dev libbluetooth-dev libplist-dev  
libusbmuxd-dev
```

4.2 For armhf desktop:

We saved all the packages on the github, you just need download and install as follow.

```
$git clone https://github.com/APCVSRepo/sdl_implementation_reference.git  
$cd sdl_implementation_reference  
$git checkout sdl_rk3288_dependency_library  
$cd sdl_dependence  
$sudo ./install.sh
```



本作品采用[知识共享署名-相同方式共享 4.0 国际许可协议](https://creativecommons.org/licenses/by-sa/4.0/)进行许可

Project Name: sdl_core	Version: <1.0>
SDL_Develop_Environment	Date: <14/11/2017>
<document identifier>	

And you can also uninstall all the packages by uninstall.sh command.

```
$sudo ./uninstall.sh
```

5 Build and Run

5.1 Get source code.

```
$git clone https://github.com/APCVSRepo/sdl_implementation_reference.git
$cd sdl_implementation_reference
$git checkout develop
```

5.2 cmake && make && make install

5.2.1 For x86 desktop:

```
$cd sdl_implementation_reference
$mkdir build
$cd build
$cmake -DCMAKE_BUILD_TYPE="Release" ..
$make
$make install
```

5.2.2 For armhf desktop:

```
$cd sdl_implementation_reference
$mkdir build
$cd build
$cmake \
-DCMAKE_BUILD_TYPE="Release" \
-DCMAKE_CXX_COMPILER="arm-linux-gnueabi-g++" \
-DCMAKE_C_COMPILER="arm-linux-gnueabi-gcc" \
-DUDEV_PATH_INCLUDES="/usr/arm-linux-gnueabi/include" \
```



本作品采用[知识共享署名-相同方式共享 4.0 国际许可协议](https://creativecommons.org/licenses/by-sa/4.0/)进行许可

Project Name: sdl_core	Version: <1.0>
SDL_Develop_Environment	Date: <14/11/2017>
<document identifier>	

```

-DUDEV_PATH_LIB="/usr/arm-linux-gnueabi/lib" \
-DOPENSSL_ROOT_DIR="/usr/arm-linux-gnueabi/lib" \
-DOPENSSL_LIBRARIES="/usr/arm-linux-gnueabi/lib" \
-DOPENSSL_INCLUDE_DIR="/usr/arm-linux-gnueabi/include" \
-DSQLITE3_LIBRARIES="/usr/arm-linux-gnueabi/lib" \
-DSQLITE3_INCLUDE_DIRS="/usr/arm-linux-gnueabi/include" ..

$make

$make install

```

