

Generating Standalone SDK Toolchain

MA35XX

江天文

2024/03/22

Joy of innovation
nuvoTon

| Introduction

- For third-party developers, standalone SDK toolchain is must be have on hand.
- By distributing standalone SDK tool to independent software vendors, OEM manufacturers need not to disclose proprietary source code to the public.
- Resort to the local makefile local.mk, SDK toolchain can be generated without user intervention by the extensions of build instruction.

| Generating standalone SDK toolchain

- Change directory to *the root of Buildroot* (**`${BR2_DIR}`**)

`$ cd ${BR2_DIR}`

- Fetch and put the file ***local.mk*** into the root of Buildroot.

LINK: <https://raw.githubusercontent.com/symfund/ma35d1-portal/master/mk/local.mk>

- Override the source directories of TF-A, U-Boot, Linux and optee-os in file ***local.mk***

`ARM_TRUSTED_FIRMWARE_OVERRIDE_SRCDIR`=/path/to/actual/source/directory/of/tf-a

`UBOOT_OVERRIDE_SRCDIR`=/path/to/actual/source/directory/of/uboot

`LINUX_OVERRIDE_SRCDIR`=/path/to/actual/source/directory/of/linux

`OTEE_OS_OVERRIDE_SRCDIR`=/path/to/actual/source/directory/of/optee-os

Toolchain Requirements

- Before generating SDK tool, Buildroot must be correctly configured. That means toolchain options are tailored to meet actual requirements, some mandatory packages are selected in mind.

→Toolchain

[*] Build cross gdb for the host

Python support (Python 3)

→Target packages →Debugging, profiling and ...

[*] gdb

→Target packages →Networking applications

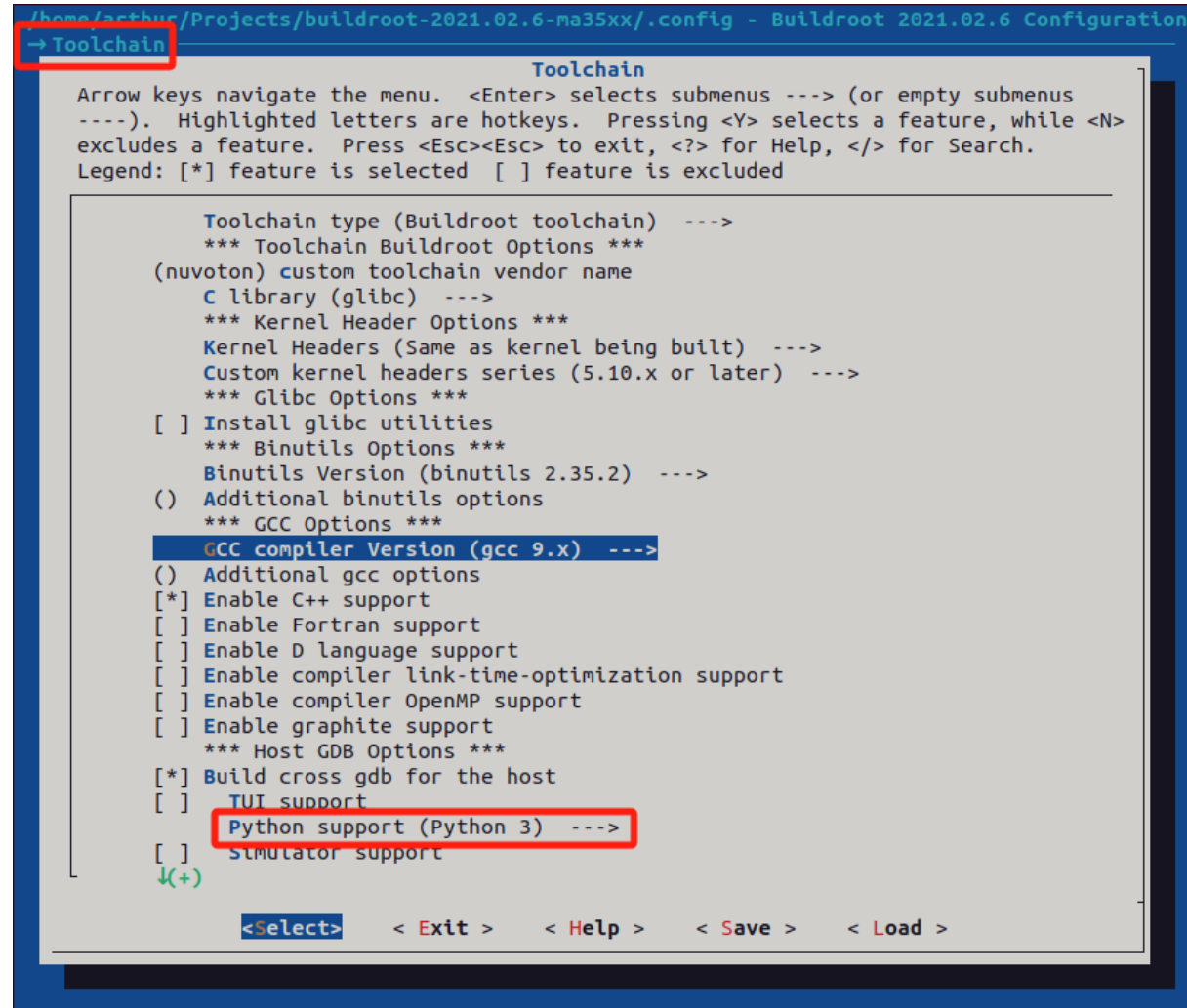
[*] openssh

[*] client

[*] server

[*] key utilities

[*] rsync



```
Buildroot 2021.02.6 Configuration
Toolchain
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenu
----). Highlighted letters are hotkeys. Pressing <Y> selects a feature, while <N>
excludes a feature. Press <Esc><Esc> to exit, <?> for Help, </> for Search.
Legend: [*] feature is selected [ ] feature is excluded

Toolchain type (Buildroot toolchain) --->
*** Toolchain Buildroot Options ***
(nuvoton) custom toolchain vendor name
C library (glibc) --->
*** Kernel Header Options ***
Kernel Headers (Same as kernel being built) --->
Custom kernel headers series (5.10.x or later) --->
*** Glibc Options ***
[ ] Install glibc utilities
*** Binutils Options ***
Binutils Version (binutils 2.35.2) --->
() Additional binutils options
*** GCC Options ***
GCC compiler Version (gcc 9.x) --->
() Additional gcc options
[*] Enable C++ support
[ ] Enable Fortran support
[ ] Enable D language support
[ ] Enable compiler link-time-optimization support
[ ] Enable compiler OpenMP support
[ ] Enable graphite support
*** Host GDB Options ***
[*] Build cross gdb for the host
[ ] TUI support
Python support (Python 3) --->
[ ] Simulator support
jk(+)

<Select> < Exit > < Help > < Save > < Load >
```

| Generating standalone SDK toolchain

- Begin building SDK toolchain, the action *make clean* is optional, if want to save the build time.

\$ make clean ; make sdk-tool

- When complete generating SDK tool, the SDK tool is located at ***output/images/aarch64-nuvoton-linux-gnu_sdk-buildroot_installer***

- To install the SDK tool on local computer in which the SDK tool is built or another computer, launch the SDK tool installer show below.

\$ sudo output/images/aarch64-nuvoton-linux-gnu_sdk-buildroot_installer

- By default, the SDK tool is installed in ***/opt/aarch64-nuvoton-linux-gnu_sdk-buildroot***
- Before using the SDK tool, open a terminal window and set up the build environment for the new terminal.

\$ source /opt/aarch64-nuvoton-linux-gnu_sdk-buildroot/environment-setup

Joy of innovation
nuvoTon

谢谢

謝謝

Děkuji

Bedankt

Thank you

Kiitos

Merci

Danke

Grazie

ありがとう

감사합니다

Dziękujemy

Obrigado

Спасибо

Gracias

Teşekkür ederim

Cảm ơn