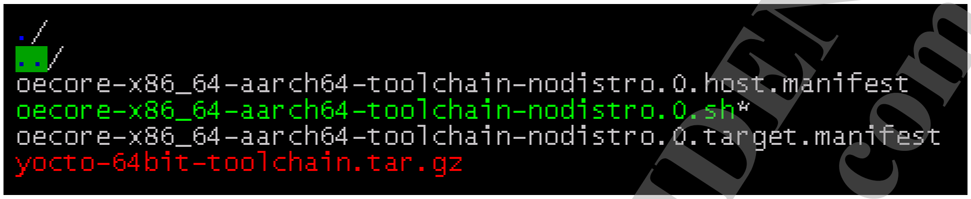
MT816 Tool Chain User Guide

- ## Step 1 :

Extract “yocto-64bit-toolchain.tar.gz” in your local folder,

Such as “/proj/myaccount/MT8516-SDK”

***[MT8516_SDK]$** ls*

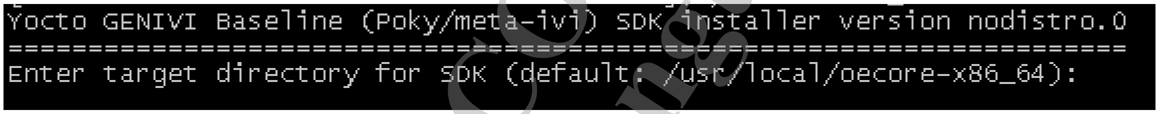


```
./
../
oecore-x86_64-aarch64-toolchain-nodistro.0.host.manifest
oecore-x86_64-aarch64-toolchain-nodistro.0.sh*
oecore-x86_64-aarch64-toolchain-nodistro.0.target.manifest
yocto-64bit-toolchain.tar.gz
```

- ## Step 2:

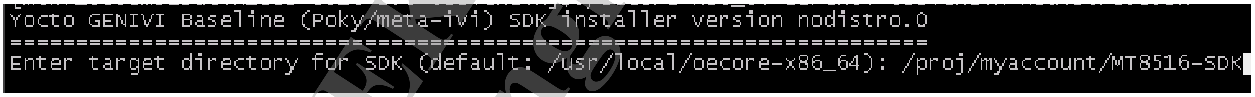
Run “oecore-x86_64-aarch64-toolchain-nodistro.0.sh” to install toolchain.

[MT8516_SDK]$ *./oecore-x86_64-aarch64-toolchain-nodistro.0.sh*

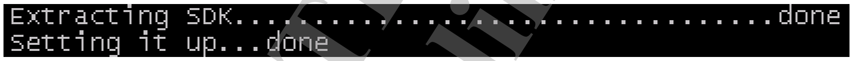


```
Yocto GENIVI Baseline (Poky/meta-ivi) SDK installer version nodistro.0
======================================================================
Enter target directory for SDK (default: /usr/local/oecore-x86_64):
```

Input target directory , such as “/proj/myaccount/MT8516-SDK”



```
Yocto GENIVI Baseline (Poky/meta-ivi) SDK installer version nodistro.0
======================================================================
Enter target directory for SDK (default: /usr/local/oecore-x86_64): /proj/myaccount/MT8516-SDK
```



```
Extracting SDK.................................done
Setting it up...done
```

- ## Step 3:

Configuration environment parameter.

You can run “environment-setup-aarch64-poky-linux” script to configuration environment. This script file is under the directory which you set in step 2.

***[MT8516_SDK]$** source environment-setup-aarch64-poky-linux*

- ## Step 4:

Now you can use this toolchain to compile your program

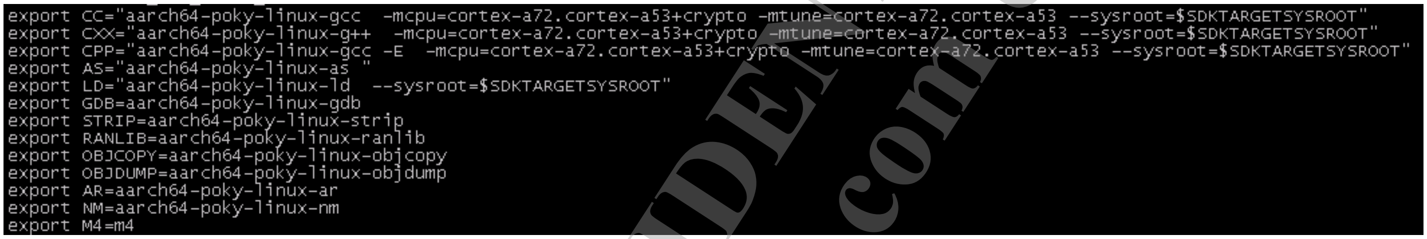
[MT8516_SDK]$ *vi test.c*

Edit your test.c.

[MT8516_SDK]$ $CC test.c -o test

For environment variables, you can refer to “environment-setup-aarch64-poky-linux”

[MT8516_SDK]$ *vi environment-setup-aarch64-poky-linux*



```
export CC="aarch64-poky-linux-gcc  -mcpu=cortex-a72.cortex-a53+crypto -mtune=cortex-a72.cortex-a53 --sysroot=$SDKTARGETSYSROOT"
export CXX="aarch64-poky-linux-g++  -mcpu=cortex-a72.cortex-a53+crypto -mtune=cortex-a72.cortex-a53 --sysroot=$SDKTARGETSYSROOT"
export CPP="aarch64-poky-linux-gcc -E  -mcpu=cortex-a72.cortex-a53+crypto -mtune=cortex-a72.cortex-a53 --sysroot=$SDKTARGETSYSROOT"
export AS="aarch64-poky-linux-as "
export LD="aarch64-poky-linux-ld  --sysroot=$SDKTARGETSYSROOT"
export GDB=aarch64-poky-linux-gdb
export STRIP=aarch64-poky-linux-strip
export RANLIB=aarch64-poky-linux-ranlib
export OBJCOPY=aarch64-poky-linux-objcopy
export OBJDUMP=aarch64-poky-linux-objdump
export AR=aarch64-poky-linux-ar
export NM=aarch64-poky-linux-nm
export M4=m4
```

… …

- references

https://gcc.gnu.org/onlinedocs/gcc/AArch64-Options.html#AArch64-Options