

بسم الله الرحمن الرحيم

هدیه به امام زمان عج و نائب بر حقش امام خامنه ای

نحوه آماده سازی Toolchain: QuecOpen برای SDK مازول EC200U و کامپایل SDK

ابزار ها:

1- SDK مخصوص EC200U برای برنامه نویسی

2- Qflash برای دانلود فایل کامپایل شده روی مازول در ویندوز (و Qloader در لینوکس)

کامپایل کردن SDK:

روش اول: استفاده از فایل aMahdiRahi_build_ec200u_dev.bat که در کنار فایل build_all.bat قرار دارد. کافی است که روی آن دبل کلیک کنید. یا راست کلیک کنید و گزینه Run as Administrator را انتخاب کنید.

توجه: این فایل به غیر از کامپایل (بیلد) sdk اقدام به فشردن سازی و کپی گرفتن از فولدر target که حاصل بیلد قدیمی است میکند و همینطور از محتویات فولدر Components به غیر از فولدر ql-config فایل فشردن تهیه میکند. فایل های کپی شده و فشردن شده همگی درون فولدر MahdiRahi_BuildArchive ذخیره میشوند. که برای هر بار اقدام به بیلد کردن یک فولدر با نامی که شامل تاریخ و زمان به ثانیه است میسازد و در آن قرار میدهد.

روش دوم:

0- همیشه در ابتدا فولدر out که از دفعه قبلی بیلد کردن sdk بوجود آمده است را پاک کنید. (پاک کردنش باعث کندتر شدن زمان ساخت پروژه نمیشود و تاثیری ندارد)

00- در نام فولدرهای مسیر SDK هیچ اسپیدی وجود نداشته باشد.

1- اگر از ویندوز استفاده میکنید لازم است CMD را با دسترسی Administrator باز کرده و به آدرس SDK بروید. (اگر خواستید از PowerShell استفاده کنید در انتها توضیح دادم)

- در CMD برای رفتن به آدرس از دستور "cd" استفاده میکنیم. (بعد از cd باید فقط یک اسپیس بزاریم و بعد آدرس فولدر را از Explorer ویندوز کپی و در cmd پیست میکنیم.) مثلا به آدرس زیر میریم:

```
cd D:\SDK\LTE01R03A07_C_SDK_U
```

اگر جواب نداد، ابتدا فقط آدرس درایو مد نظر با علامت ":" رو بنویسید و بعد آدرس رو با دستور "cd" و بدون اسم درایو وارد کنید:

D:

```
cd SDK\LTE01R03A07_C_SDK_U
```

2- سپس با دستور زیر کار بیلد کردن را انجام دهید.

build_all.bat new EC200UEU_AA V01

چنانچه به درستی انجام شده باشد خروجی به شکل زیر خواهید داشت

```
kernel total size:0x240000|2304.00KB,used:0x23e520|2297.28KB,remain:0x1ae0|6.72KB
app total size:0x120000|1152.00KB,used:0x5c00|23.00KB,remain:0x11a400|1129.00KB

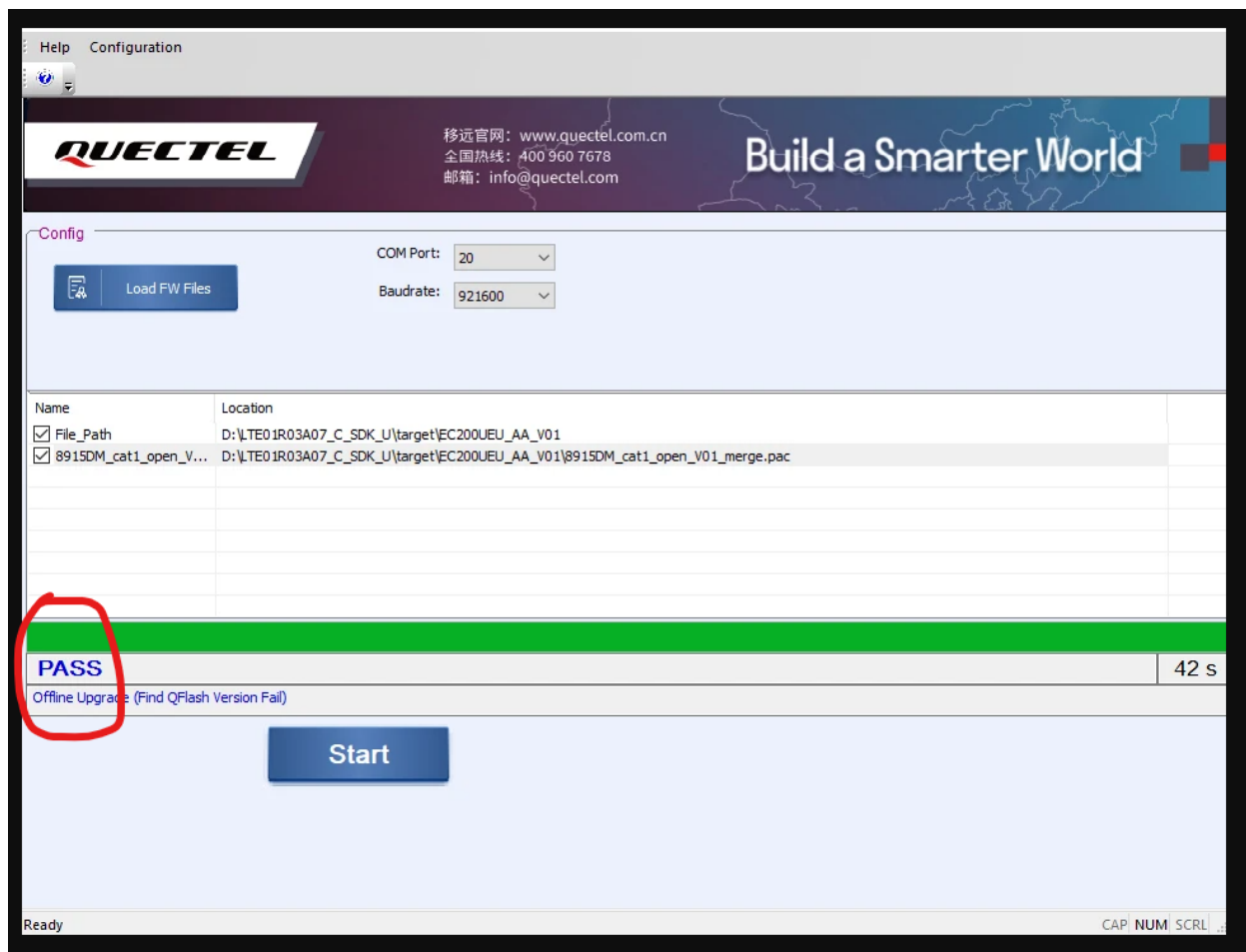
EC200UEU_AA V01 release NOVOLTE SINGLESIM

.\hex\8915DM_cat1_open-cat1_UIS8915DM_BB_RF_SS_NoVolte_cus-release.pac
.\hex\8915DM_cat1_open-cat1_UIS8915DM_BB_RF_SS_NoVolte_cus-release.pac.d
.\hex\8915DM_cat1_open.elf
.\hex\8915DM_cat1_open.img
.\hex\8915DM_cat1_open.map
.\hex\8915DM_cat1_open.sign.img
.\hex\appimg_flash_delete.pac
.\hex\boot.elf
.\hex\boot.img
.\hex\boot.map
.\hex\boot.sign.img
.\hex\boot2.elf
.\hex\boot2.img
.\hex\boot2.img.lzmar
.\hex\boot2.map
.\hex\cat1_UIS8915DM_BB_RF_SS_NoVolte_cus.img
.\hex\cat1_UIS8915DM_BB_RF_SS_NoVolte_cus_nvitem.bin
.\hex\cat1_UIS8915DM_BB_RF_SS_NoVolte_cus_prepack.cpio
.\hex\fdl1.elf
.\hex\fdl1.img
.\hex\fdl1.map
.\hex\fdl1.sign.img
.\hex\fdl2.elf
.\hex\fdl2.img
.\hex\fdl2.map
.\hex\fdl2.sign.img
.\hex\flash_test_fdl.elf
.\hex\flash_test_fdl.img
.\hex\flash_test_fdl.map
.\hex\partinfo.bin
30 File(s) copied
      1 file(s) moved.
      1 file(s) moved.
D:\LTE01R03A07_C_SDK_U\target\EC200UEU_AA_V01\prepack\8915DM_cat1_open-cat1_UIS8915DM_BB_RF_SS_NoVolte_cus-release.pac
      1 file(s) moved.
.\target.cmake
1 File(s) copied
.\hex\examples\V01_APP.elf
.\hex\examples\V01_APP.img
.\hex\examples\V01_APP.map
.\hex\examples\V01_APP.pac
.\hex\examples\V01_APP.sign.img
.\hex\examples\V01_APP_prepack.cpio
6 File(s) copied
      1 file(s) moved.
      1 file(s) moved.
      1 file(s) moved.

Sun 06/29/2017
START TIME: 12:57:26.84
END TIME: 12:58:49.43
```

اگر پروژه شما به درستی بیلد شد در دایرکتوری target خروجی چیزی را که ساختید مشاهده میکنید.

3- نرم افزار Qflash را باز کنید و آدرس فایلی که ساختیم (8915DM_cat1_open_V01_merge.pac) را به آن بدهید و دکمه استارت را بزنید. چنانچه کار با موفقیت انجام شود باید خروجی مانند زیر داشته باشیم.



- کامپایل SDK توسط PowerShell

توجه شود که اولویت با CMD است.

نگته: تفاوت PowerShell با CMD در دستور فایل بیلد است. (باید ابتدای دستور بیلد کردن علامت "\" را بزاریم)

1- PowerShell رو با دسترسی Administrator باز کرده و به آدرس SDK بروید.

2- سپس با دستور زیر کار بیلد کردن را انجام دهید.

.\build_all.bat new EC200UEU_AA V01

نکته: اگر همه موارد بالا رو بدرستی رفتید اما با خطا در زمان بیلد کردن مواجه شدید ، فولدر sdk رو در آنتی ویروس مثلا Windows Defender را exclude کنید. (چون: آنتی ویروس فایل را قفل کرده و اجازه نوشتن نمی دهد. برای SDK های Quectel این خیلی رایج است چون در زمان build صدها فایل a. سریع ساخته و rename می شوند. بهترین کار این است که کل مسیر SDK را در **Windows Defender Exclusion** بگذاری.)

1- باز کردن Windows Security

در Start جستجو کن:

Windows Security

برو به:

Virus & threat protection -> Manage settings -> Exclusions -> Add or remove exclusions -> Add an exclusion -> Folder

حالا آدرس مسیر فولدر های زیر رو بده (برای هر فولدر باید مرحله 1 را تکرار کنید و یک فولدر اضافه کنید):

اول مسیر فولدر اصلی پروژه و سپس فولدر های زیر در فولدر اصلی پروژه:

out

prebuilts

target

build

2- برو به:

Virus & threat protection -> Manage settings -> Exclusions -> Add or remove exclusions -> Add an exclusion -> Process

و سپس نام فرایند زیر رو وارد کنید (برای هر پروسس باید مرحله 2 را تکرار کنید و یک process اضافه کنید):

arm-none-eabi-ar.exe

arm-none-eabi-gcc.exe

arm-none-eabi-g++.exe

ninja.exe

cmake.exe

make.exe

3- میتوانید این مرحله 3 را ازش عبور کنید و اگر بعد از اتمام مراحل بعد باز هم مشکل وجود داشت ، آنگاه این مرحله 3 را نیز انجام دهید. مرحله 3 ، در windows security defender برو به:

Virus & threat protection -> Manage settings -> Exclusions -> Add or remove exclusions -> Add an exclusion -> File Type

و سپس نام فرایند زیر رو وارد کنید(برای هر File Type باید مرحله 3 را تکرار کنید و یک File Type اضافه کنید):

.a

.o

.elf

.bin

4- فولدر out در sdk که از بیلد ناقص قبلی باقی مانده را حذف کنید

- همیشه در ابتدا فولدر out که از دفعه قبلی بیلد کردن sdk بوجود آمده است را پاک کنید. (پاک کردنش باعث کندتر شدن زمان ساخت پروژه نمیشود و تاثیری ندارد)

- در نام فولدرهای مسیر SDK هیچ اسپیدی وجود نداشته باشد.

5- cmd اگر باز هست را ببند و از ابتدا عملیات کامپایل (بیلد) کردن رو تکرار کنید.

نکته: اگر همه موارد بالا رو بدرستی رفتید اما با خطا در زمان بیلد کردن مواجه شدید ، راه دیگه غیرفعال کردن UAC ویندوز هست. مراحل زیر رو دنبال کنید:

وارد Control Panel و سپس به User Accounts شوید و سپس روی گزینه

Change User Account Control settings

کلیک کنید و در پنجره باز شده نوار کنترل رو روی پایین ترین سطح یعنی Never notify قرار بدین و سپس روی دکمه Ok کلیک کنید و سپس روی گزینه Yes کلیک کنید. حالا به احتمال بالا پروژه بیلد خواهد شد. البته توجه کنید چون امنیت سیستم رو کاهش میده حتما بعد از بیلد یا بعد از اتمام کار با پروژه در آن روز کاری مجددا به حالت قبل برگردانید البته تا زمانی که با برنامه ای با منبع ناشناس یا برنامه ای که کرک نشده باشد کار کنید ، غیر فعال بودن این گزینه معمولا مشکلی ایجاد نمیکند.

نکته: اگر همه موارد بالا رو بدرستی رفتید اما با خطا در زمان بیلد کردن مواجه شدید ، راه دیگه غیرفعال کردن Real-time protection در Windows Security Defender است. مراحل زیر رو دنبال کنید:

در ویندوز در منوی Start به قسمت Setting بروید سپس به Update & Security و بعد به Windows Security بروید.

سپس روی Virus & threat protection کلیک کنید در پنجره باز شده در بخش Virus & threat protection settings بخش Manage settings کلیک کنید سپس وضعیت نوار Real-time protection را روی off بزارید. حال، sdk را بیلد کنید.

نکته: ممکن است پروژه SDK در ابتدا و در حالتی که هیچ تغییری بهش ندادین یا بسیار تغییرات کم بود ، بدرستی بیلد شود اما بعد از تغییرات زیاد و کد نویسی زیاد و اضافه کردن دیفاین و کتابخانه های مختلف ، در زمان بیلد کردن با خطا های عدم دسترسی به فایل هایی که قبلا دسترسی داشته (فایل های خود sdk) یا عدم امکان دسترسی به تغییر نام فایل های a. و از این قبیل که ربطی به کد شما ندارد روبرو شوید. راه حل:

در شاخه اصلی SDK فایل build_all.bat را با ادیتور متنی باز کنید. (یک فایل اسکریپت است) تنها در یکجای کدهای اسکریپت متن G Ninja وجود دارد (حدود خط 274) بعد از آن دو if و کدهای داخلش هست که در انتهای if دوم دستور ninja وجود دارد:

```
build_all.bat
263
264 if not exist %PROJECT_OUT% mkdir %PROJECT_OUT%
265 if not exist %PROJECT_OUT%\include mkdir %PROJECT_OUT%\include
266 cd /d out\%BUILD_TARGET%_%BUILD_RELEASE_TYPE%
267
268 echo=
269
270 echo %PATH%
271
272 echo=
273
274 cmake ../../ -G Ninja
275
276 echo=
277 if not "%build_tasknum%"==" " (
278     set /a tasknumCheck=%build_tasknum%
279     if "!tasknumCheck!" == "%build_tasknum%" (
280         ninja -j !tasknumCheck!
281     ) else (
282         cd /d %PROJECT_ROOT%
283         echo.
284         echo *****
285         echo xxxxxxxxxxxxxxxxxxxx ninja tasknum error xxxxxxxxxxxxxxxxxxxx
286         echo *****
287         set ret=1
288         goto doneseg
289     )
290 ) else (
291     ninja
292 )
293
294 if not %errorlevel% == 0 (
295     cd /d %PROJECT_ROOT%
296     echo.
297     echo.
298     echo ***** ERROR *****
299     echo ***** ERROR *****
300     echo *****
301     echo xxxxxxxxxxxxxxxxxxxx build ended error xxxxxxxxxxxxxxxxxxxx
```

این کلمه ninja را به

ninja-j1

یا

ninja-j2

تغییر بدین. درواقع این sdk خیلی قدیمی است احتمالا قبل از 2018 و این دستور ninja باعث میشود در ویندوز بطور پیش فرض بصورت موازی اقدام به بیلد sdk کند و چون خیلی قدیمی و ویندوز های امروزی جدید هستند ، باعث میشود بیلد موازی بدرستی انجام نشود و بعضی از دستورات قبل از اینکه دستورات مورد نیازشان اجرا شوند و نیاز های اولیه آنها ساخته شود ، اجرا شوند که باعث میشود فایل های لازم را نداشته باشند اما با تغییر به ninja-j1 یا ninja-j2 میزان موازی انجام شدن را کاهش میدهم و این باعث میشود که بدرستی انجام شود. توجه: ninja-j2 موازی با تعداد کم انجام میشود و ninja-j1 بصورت تکی هر دستور به نوبت انجام میشود. ninja-j1 از اطمینان بیشتری برخوردار است اما زمان بیلد کمی بیشتر میشود. (مثلا بجای 1 دقیقه میشود حدود 1:40 ثانیه)