

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.) مثلاً 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

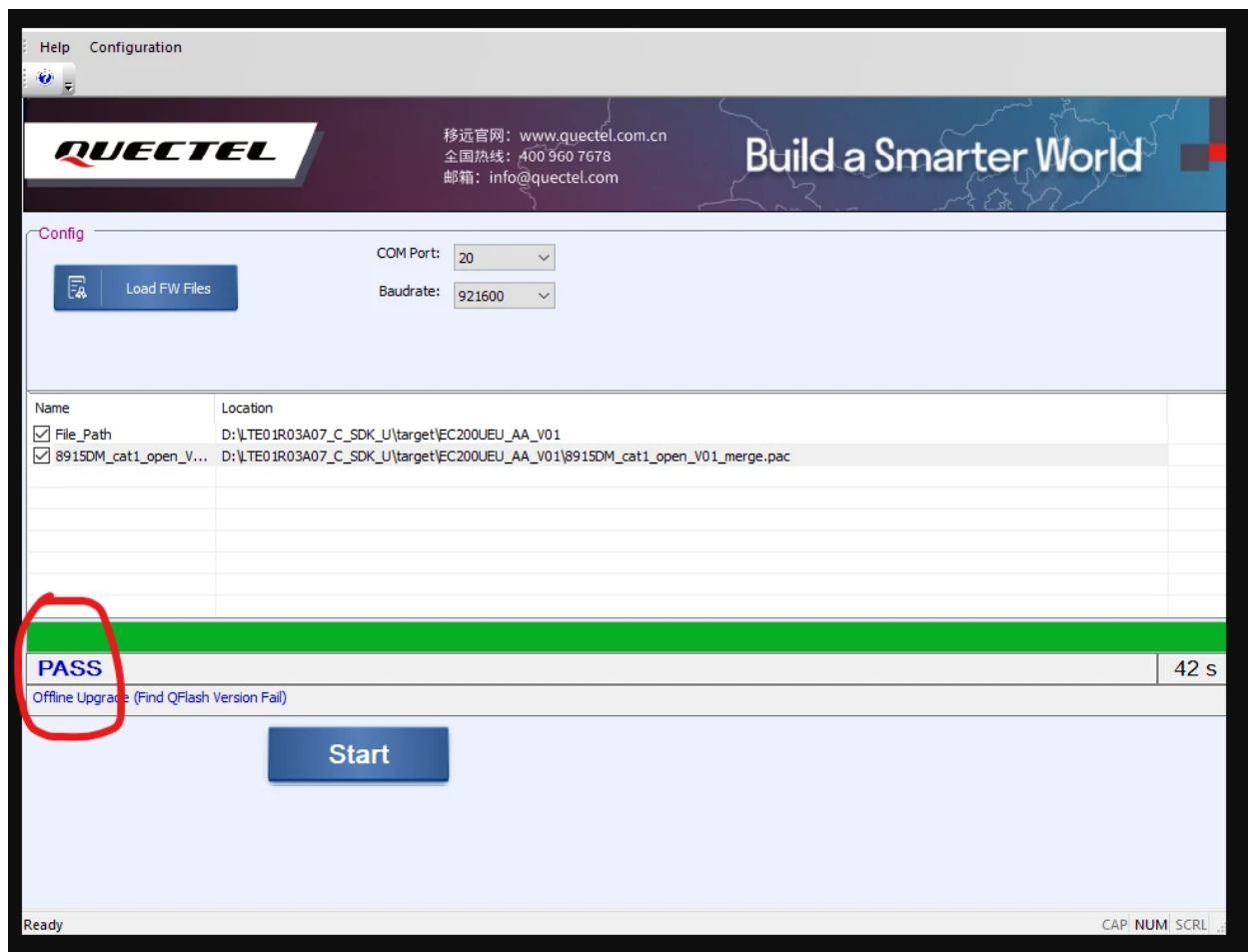
EC200UEU_AA V01 release NOVOLTE SINGLESIM

.\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 06/29/2012
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.

نکته: 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 Quetel 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

2- برو به:

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

حالا آدرس مسیر رو بده

3- 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.

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

اسکرپت فایل aMahdiRahi_build_ec200u_dev.bat میتوانید یک فایل با فرمت .bat بسازید و متن زیر رو داخلش قرار بدین و سپس فایل رو در فولدر SDK کنار فایل build_all.bat قرار بدین و استفاده نمایید:

```
@echo off
```

```
title Mahdi Rahi EC200U Easy Build System
```

```
setlocal EnableExtensions EnableDelayedExpansion
```

```
===== ::
```

```
::AUTO ADMIN ELEVATION (Robust & Argument-Preserving)
```

```
===== ::
```

```
@setlocal enableextensions
```

```
@cd /d "%~dp0"
```

```
::Admin بررسی دسترسی
```

```
net session >nul 2>&1
```

```
if %errorlevel% neq 0)
```

```
    echo Requesting Administrator privileges...
```

```
:: powershell -NoProfile -ExecutionPolicy Bypass -Command "Start-Process -FilePath '%~f0' -Verb RunAs"
```

```
    powershell -Command "Start-Process '%~f0' -Verb RunAs"
```

```
exit /b
```

```
(
```

```
:: اگر ادمین هستیم، ادامه می دهیم
```

```
@endlocal
```

```
echo Running with Administrator privileges...
```

echo.

===== ::

::AUTO DETECT SDK ROOT (DRIVE + PATH)

===== ::

::set SDK_DRIVE=%~d0

::set SDK_PATH=%~p0

::set SDK_ROOT=%SDK_DRIVE%%SDK_PATH%

set SDK_ROOT=%~dp0

% ::SDK_DRIVE%

::cd "%SDK_PATH" %

cd /d "%SDK_ROOT" %

echo SDK Root Detected:

echo %SDK_ROOT%

echo=====

echo EC200U Build System

echo=====

===== ::

::GET DATE & TIME (24H)

===== ::

```
for /f %%i in ('powershell -NoProfile -Command "Get-Date -Format yyyy.MM.dd_HH.mm.ss"') do set
DATETIME=%%i
```

```
===== ::
```

```
::PATH DEFINITIONS
```

```
===== ::
```

```
set BASE_DIR=%SDK_ROOT%MahdiRahi_BuildArchive
```

```
set BUILD_DIR=%BASE_DIR%\%DATETIME%
```

```
set OLD_DIR=%BUILD_DIR%\previousVersion
```

```
set TARGET_DIR=%SDK_ROOT%\target
```

```
===== ::
```

```
::CREATE DIRECTORIES
```

```
===== ::
```

```
if not exist "%BASE_DIR%" mkdir "%BASE_DIR"%%
```

```
if not exist "%BUILD_DIR%" mkdir "%BUILD_DIR"%%
```

```
if not exist "%OLD_DIR%" mkdir "%OLD_DIR"%%
```

```
===== ::
```

```
::BACKUP TARGET
```

```
===== ::
```

```
if exist "%TARGET_DIR) "%
```

```
echo.
```

```
echo Compressing TARGET folder...
```

```
powershell -command "Compress-Archive -Path '%TARGET_DIR%' -DestinationPath  
'%OLD_DIR%\target.zip' -Force"
```

```
echo Backup created:
```

```
echo %OLD_DIR%\target.zip
```

```
(else)
```

```
echo.
```

```
echo TARGET folder not found. Skipping backup.
```

```
(
```

```
===== ::
```

```
::CLEAN OLD BUILD FOLDERS
```

```
===== ::
```

```
echo.
```

```
echo=====
```

```
echo CLEANING BUILD ENVIRONMENT
```

```
echo=====
```

```
echo.
```

```
echo Stopping possible build processes...
```

```
for %%p in)
```

```
ninja.exe
```

cmake.exe

arm-none-eabi-ar.exe

arm-none-eabi-gcc.exe

cc1.exe

cc1plus.exe

collect2.exe

ld.exe

(do)

taskkill /f /im %%p >nul 2>&1

(

timeout /t 2 >nul

echo Cleaning folders...

echo.

if exist "%SDK_ROOT%\out) "

echo Removing OUT folder...

call :SAFE_DELETE "%SDK_ROOT%\out"

echo OUT folder removed.

(else)

echo OUT folder not found. Skipping.

(

echo.

if exist "%SDK_ROOT%\target) "

echo Removing TARGET folder...

```

    call :SAFE_DELETE "%SDK_ROOT%\target"

    echo TARGET folder removed.

(else)

    echo TARGET folder not found. Skipping.

(

echo.

echo Clean completed.

echo.

===== ::

::START BUILD

===== ::

color 0E

for /f %%i in ('powershell -NoProfile -Command "Get-Date -Format HH:mm:ss"') do set
STARTTIME=%%i

echo Start Time: %STARTTIME%

echo.

echo.

echo=====

echo Starting Firmware Build

echo=====

echo.

call build_all.bat new EC200UEU_AA V01

```

```
set BUILD_RESULT=%ERRORLEVEL%
```

```
for /f %%i in ('powershell -NoProfile -Command "Get-Date -Format HH:mm:ss"') do set ENDTIME=%%i
```

```
for /f %%i in ('powershell -NoProfile -Command  
"$s=[datetime]::Parse('%STARTTIME%');$e=[datetime]::Parse('%ENDTIME%');($e-$s).ToString()") do set  
DURATION=%%i
```

```
echo Start Time: %STARTTIME%
```

```
echo End Time: %ENDTIME%
```

```
echo Build Duration: %DURATION%
```

```
echo.
```

```
===== ::
```

```
::BUILD RESULT
```

```
===== ::
```

```
IF %BUILD_RESULT% NEQ 0)
```

```
color 0C
```

```
echo=====
```

```
echo          BUILD FAILED
```

```
echo=====
```

```
(ELSE)
```

```
color 0A
```

```
echo=====
```

```
echo          BUILD SUCCESSFUL
```

```
echo=====
```

```
echo.

echo Creating components archive... "BACKUP OF USER CODE..."

set "ARCHIVE_BASE=!BASE_DIR!\!DATETIME"!

set "COMPONENT_DIR=!ARCHIVE_BASE!\components"

echo Backup of: Folder components (else folder: ql-config)

if not exist "!COMPONENT_DIR!" mkdir "!COMPONENT_DIR"!

powershell -NoProfile -Command^

    $"items = Get-ChildItem '%SDK_ROOT%components' -Exclude 'ql-config'; Compress-Archive -Path
$items -DestinationPath '!COMPONENT_DIR!\components.zip' -Force"

echo.

echo Components archive created:

echo !COMPONENT_DIR!\components.zip

(

pause

echo 10

pause

echo 9

pause

echo 8

pause

echo 7

pause

echo 6
```

pause

```
echo 5
```

pause

echo 4

pause

echo 3

pause

```
echo 2
```

pause

```
echo 1
```

pause

```
echo please click exit window "x."
```

```
:WAIT_CLOSE
```

```
timeout /t 3600 >nul
```

```
goto WAIT_CLOSE
```

=====

===== ::

=====

===== :

===== :

=====

=====

=====

=====

=====

=====

```
===== ::
```

```
::FUNCTION: DELETE FOLDER
```

```
===== ::
```

```
:SAFE_DELETE
```

```
set "TARGET=%~1"
```

```
set RETRY=0
```

```
set MAX_RETRY=5
```

```
:DELETE_LOOP
```

```
if exist "%TARGET%" "%
```

```
    rmdir /s /q "%TARGET%" 2>nul
```

```
(
```

```
if exist "%TARGET%" "%
```

```
    set /a RETRY+=1
```

```
    if !RETRY! LEQ %MAX_RETRY) %
```

```
        echo Retry !RETRY! deleting %TARGET%
```

```
        timeout /t 3 >nul
```

```
        goto DELETE_LOOP
```

```
( else)
```

```
    color 0C
```

```
    echo=====
```

```
    echo          BUILD FAILED
```

```
    echo=====
```

```
    echo ERROR: Could not delete %TARGET%
```

```
    echo please remove manually.
```

pause

echo 4

pause

echo 3

pause

echo 2

pause

echo 1

pause

exit /b 1

(

(

exit /b 0