

بسم الله الرحمن الرحيم

آماده سازی vscode برای کار کردن با sdk QuecOpen

توجه داشته باشید که sdk QuecOpen نیازی به ide ندارد، و به کامپایلر نیز نیازی ندارد و همه چیز را درون خودش دارد.

برای کدنویسی درون این sdk کافیهست که فایل های کد خودمان را باتوجه به آموزش های قبل ایجاد و شروع به نوشتن کنیم و سپس همانند آموزش های قبل اقدام به بیلد کردن sdk کنیم.

پس در کل ما نیاز به یک ادیتور داریم.

با اینحال برای اینکه راحتتر اقدام به برنامه نویسی کنیم و دستورات زبان c را نیز با کمک یک ادیتور حرفه ای استفاده کنیم یا از قابلیت های زبان c بهره ببریم بهتره که از یک ادیتور حرفه ای مثل vscode استفاده کنیم و آن را برای این sdk که زبان c است تنظیم کنیم.

1- دانلود و نصب vscode

2- نصب اکتشن های زبان c بر روی vscode

3- ساخت یک پروفایل زبان c روی vscode و انتخاب این پروفایل برای برنامه نویسی

4- ایجاد workspace

5- برنامه نویسی و استفاده از قابلیت های vscode

توجه شود که ما نیازی نداریم از کامپایلر هایی که میشه بر روی vscode نصب کنیم ، استفاده کنیم پس تنظیمات آنها را نادیده میگیریم.

از بین مراحل 1 تا 5 تنها مرحله 4 سات که نیاز به کمی کار با تنظیمات vscode دارد که به شکل زیر پیش میبریم:

1- در یک شاخه عقب تر از فولدر sdk یک فولدر بنام زیر میسازیم:

workspace_vsCode

2- vscode رو باز کنید و در File->save workspace as.. کلیک کنید و فایل workspace را در فولدر workspace_vsCode با نام دلخواه مثل pymb.code-workspace ذخیره کنید.

3- فایل pymb.code-workspace را با محتوایی مثل زیر تنظیم و ذخیره کن:

```
}
" folders] :{"
    }
    "name": "user_code,"
    "path": "../pymb/components/user_code_folder"
{
    },//
" // name": "pymb,"
" // path": "../pymb"
{//
,[
" settings} :{"
" C_Cpp.default.configurationProvider": "ms-vscode.cpp-tools"
{
{
```

اینجوری در Explorer محیز vscode بخشی بنام user_code ایجاد میشه که به ادرس داده شده اشاره میکنه تا دسترسی راحت داشته باشیم. (درواقع عمل map کردن پوشه ها را انجام میدهیم.)

4- باید settings.json سراسری را باز و تنظیم کنیم برای اینکار، vscode را اجرا کن و دکمه های ctrl+shift+p را بزن و دستور زیر را وارد کن:

Preferences: Open User Settings (JSON)

سپس Enter را بزن.

با محتویات زیر کامل جایگزینش کن:

```
{
  "explorer.confirmDelete": false,
  "workbench.colorTheme": "Experimental Light",
  "C_Cpp.default.includePath": [
    "D:/prj/sdk_pymb/pymb/components/**",
    "D:/prj/sdk_pymb/pymb/components/user_code_folder/include/user_code",
    "D:/prj/sdk_pymb/pymb/components/ql-kernel/inc"
  ],
  "C_Cpp.default.defines": [
    "QUECTEL",
    "QUECOPEN_SDK",
    "EC200U"
  ],
  "C_Cpp.default.compilerPath": "D:/prj/pymb/prebuilts/win32/gcc-arm-none-eabi/bin/arm-none-eabi-gcc.exe",
  "C_Cpp.default.cStandard": "c11",
  "C_Cpp.default.cppStandard": "c++14",
  "C_Cpp.default.intelliSenseMode": "windows-gcc-arm",
  "files.associations": {
    "*.c": "c",
    "*.h": "c"
  },
  "editor.formatOnSave": true,
  "editor.defaultFormatter": "ms-vscode.cpptools"
}
```

ذخیرش کن و ببندش.

توجه شود که برای تشخیص فایل های include شده باید آدرس آنها در فایل بالا به شکل مثال زیر:

```
"C_Cpp.default.includePath": [  
  "D:/prj/sdk_pymb/pymb/components/**",  
  "D:/prj/sdk_pymb/pymb/components/user_code_folder/include/user_code",  
  "D:/prj/sdk_pymb/pymb/components/ql-kernel/inc"  
],
```

تنظیم شود.

تمام شد.