Mac OS下STM32Cube生成VSCode匹配的Python脚本

- 使用方法：在STM32CUBE生成的GCC工程目录下（Makefile所在目录）执行该脚本，生成VSCode需要的c_cpp_properties.json文件。
- 需要先安装好编译器：`arm-none-eabi-gcc`；调试器插件：`cortex-debug`，
- 实例配置lanch.json如下：

```
{
    // 使用 IntelliSense 了解相关属性。
    // 悬停以查看现有属性的描述。
    // 欲了解更多信息，请访问: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "cwd": "${workspaceRoot}",
            "executable": "${workspaceRoot}/build/TEST_VSCODE.elf",
            "servertype": "jlink",
            "type": "cortex-debug",
            "request": "launch",
            "name": "Debug (J-Link)",
            "device": "STM32F103RB",
            "interface": "swd",
            "ipAddress": null,
            "serialNumber": null,
        }
    ]
}
```

- Python脚本

```
#!/usr/bin/env python
# encoding: utf-8

"""
@version: v1.0
@author: momowang
@license: Apache Licence
@file: addHeder.py
```

```
@time: 2018/9/8 下午2:41
"""
import os,re,json

path = os.getcwd()

# 获取工具链绝对路径:arm_none_eabi_gcc
def toolChainsPath():
    pipeline = os.popen("which arm-none-eabi-gcc")
    # print(pipeline.read())
    return pipeline.read()

# 读取Makefile的文件,获取到头文件路径
def getConfigInfo():
    # print("当前文件目录是:" + path)
    headerArry = []
    defineArry=[]
    file = None
    try:
        file = open(path+"/Makefile","r")
        buff = file.read()
        # 查找两字符串之间的字符串,并替换删除多余部分
        pattern = re.compile("# C includes"+'(.*?)' + "# compile gcc fl
ags",re.S)
        m = pattern.findall(buff)[0][17:].replace("\\",'').replace(" ",
'').replace("-I",'')
        # 拼接"${workspaceFolder}/"
        headerArry = ["${workspaceFolder}/" + x for x in m.splitlines()
[:-1]]
        headerArry.append("${workspaceFolder}/")
        if  toolChainsPath().strip():
            headerArry.append(toolChainsPath().strip())

        pattern = re.compile("# C defines"+'(.*?)' + "# AS includes",re
.S)
        defineArry = pattern.findall(buff)[0][11:].replace("\\",'').rep
lace(" ",'').replace("-D",'').splitlines()[1:-2]

    except:
        print("出了点小问题,未找到Makefile文件或打开失败!!!")
    finally:
        if file:
           file.close()
    return headerArry,defineArry


def writeJsonConfig(version=4):
    filePath = path + "/.vscode/c_cpp_properties.json"
    if not filePath:
        print("还没有添加c_cpp_properties.json文件")
        return None
    data = ""
```

```
    with open(filePath,"r") as f:
        data = json.load(f,encoding='UTF-8')
        data["configurations"][0]["version"] = version
        data["configurations"][0]["includePath"],data["configurations"]
[0]["defines"] = getConfigInfo()
    with open(filePath,"w") as f:
        f.write(json.dumps(data,encoding='UTF-8'))
        print("修改c_cpp_properties.json的配置成功")

if __name__ == '__main__':
    writeJsonConfig()
```