mbed mit Visual Studio Code

# Benötigte Tools:

GNU ARM Embedded Toolchain: <https://developer.arm.com/open-source/gnu-toolchain/gnu-rm>

OpenOCD: <http://openocd.org/>

Visual Studio Code: https://code.visualstudio.com/  
inclusive **C/C++ Extension**

Mbed CLI: <https://github.com/ARMmbed/mbed-cli>  
erfordert **Python 2.7** und **PIP**

# Neues Projekt

In mbed (unter Windows z.B. in PowerShell)

$ mbed new <project\_name> --scm git

Toolchain einstellen in mbed\_settings.py  
*set GCC\_ARM\_PATH to the* binary *directory of your GCC Arm installation (example: C:\Program Files\GNU Tools ARM Embedded\6 2017q2\bin). Use version 6 of GCC Arm Embedded; version 5.0 or any older version might be incompatible with the tools.*

# Projekt kompilieren:

mbed compile -t <Toolchain> -m <MCU>

z.B.   
mbed compile -t GCC\_ARM -m NUCLEO\_L152RE

# Projekt zu Visual Studio Code exportieren

mbed export -i vscode\_gcc\_arm -m <MCU>

# Debugger konfigurieren

1. Open the folder in Visual Studio Code.
2. Open the ‘.vscode/launch.json’ file.
3. update the MIDebuggerPath to the full path of your copy of arm-none-eabi-gdb
4. If you’re using pyOCD as your debug server, verify that debugServerPath is set to the location of pyocd-gdbserver.
5. If you’re using OpenOCD as your debug server:
   1. Change debugServerArgs to include your OpenOCD arguments. For more info, read our [toolchain document](https://docs.mbed.com/docs/mbed-os-handbook/en/latest/debugging/toolchain/).
6. "debugServerArgs": "-f interface/stlink-v2-1.cfg -f board/st\_nucleo\_l1.cfg -c init ",

Launch.json:

{

"version": "0.2.0",

"configurations": [

{

"name": "C++ Launch",

"type": "cppdbg",

"request": "launch",

"program": "${workspaceRoot}/BUILD/${workspaceRootFolderName}.elf",

"args": [],

"stopAtEntry": true,

"cwd": "${workspaceRoot}",

"environment": [],

"externalConsole": false,

"debugServerArgs": "-f interface/stlink-v2-1.cfg -f board/st\_nucleo\_l1.cfg -c init ",

"serverLaunchTimeout": 20000,

"filterStderr": true,

"filterStdout": false,

"serverStarted": "GDB\\ server\\ started",

"preLaunchTask": "make",

"setupCommands": [

{ "text": "-target-select remote localhost:3333", "description": "connect to target", "ignoreFailures": false },

{ "text": "-file-exec-and-symbols ${workspaceRoot}/BUILD/${workspaceRootFolderName}.elf", "description": "load file", "ignoreFailures": false},

{ "text": "-interpreter-exec console \"monitor endian little\"", "ignoreFailures": false },

{ "text": "-interpreter-exec console \"monitor reset\"", "ignoreFailures": false },

{ "text": "-interpreter-exec console \"monitor halt\"", "ignoreFailures": false },

{ "text": "-interpreter-exec console \"monitor arm semihosting enable\"", "ignoreFailures": false },

{ "text": "-target-download", "description": "flash target", "ignoreFailures": false }

],

"logging": {

"moduleLoad": true,

"trace": true,

"engineLogging": true,

"programOutput": true,

"exceptions": true

},

"linux": {

"MIMode": "gdb",

"MIDebuggerPath": "/usr/bin/arm-none-eabi-gdb",

"debugServerPath": "pyocd-gdbserver"

},

"osx": {

"MIMode": "gdb",

"MIDebuggerPath": "/usr/local/bin/arm-none-eabi-gdb",

"debugServerPath": "pyocd-gdbserver"

},

"windows": {

"preLaunchTask": "make.exe",

"MIMode": "gdb",

"MIDebuggerPath": "C:\\Program Files (x86)\\GNU Tools ARM Embedded\\5.4 2016q3\\bin\\arm-none-eabi-gdb.exe",

//"debugServerPath": "C:\\Program Files (x86)\\GNU MCU Eclipse\\OpenOCD\\0.10.0-5-20171110-1117\\bin\\openocd.exe",

"setupCommands": [

{ "text": "-environment-cd ${workspaceRoot}\\BUILD" },

{ "text": "-target-select remote localhost:3333", "description": "connect to target", "ignoreFailures": false },

{ "text": "-file-exec-and-symbols ${workspaceRootFolderName}.elf", "description": "load file", "ignoreFailures": false},

{ "text": "-interpreter-exec console \"monitor endian little\"", "ignoreFailures": false },

{ "text": "-interpreter-exec console \"monitor reset\"", "ignoreFailures": false },

{ "text": "-interpreter-exec console \"monitor halt\"", "ignoreFailures": false },

{ "text": "-interpreter-exec console \"monitor arm semihosting enable\"", "ignoreFailures": false },

{ "text": "-target-download", "description": "flash target", "ignoreFailures": false }

]

}

}

]

}

# Debug Server

OpenOCD Debug Server muss manuell gestartet werden. Unter Windows z.B. mit bat Datei:

cd c:\Program Files (x86)\GNU MCU Eclipse\OpenOCD\0.10.0-5-20171110-1117\bin  
start openocd.exe -f interface/stlink-v2-1.cfg -f board/st\_nucleo\_l1.cfg -c init  
pause

# Projekt Debuggen

1. Debug Server manuell starten
2. Im Debug-Tag auf Play Button klicken
3. Projekt wird erstellt und Debugging startet, falls Build OK
4. To see warnings or errors, select View > Problems.
5. Click on the Debug Console button to see the debug output. (This is not activated automatically).

# Tutorials:

<https://docs.mbed.com/docs/mbed-os-handbook/en/latest/debugging/toolchain/>

<https://docs.mbed.com/docs/mbed-os-handbook/en/latest/debugging/vscode/>