

# Inhaltsverzeichnis

<b>Arduino mit Visual Studio Code (VSCode)</b> .....	3
<b><i>Die initiale Erzeugung der c_cpp_properties.json</i></b> .....	3
<b><i>externe Libraries immer über den Library- Manager</i></b> .....	3
<b><i>Das heißt dann</i></b> .....	3



# Arduino mit Visual Studio Code (VSCode)

Das Ecosystem von Arduino bietet viele spannende Projekte. Die Standard- IDE von Arduino bringt einen aber ziemlich an den Rand der Verzweiflung, weil sie einerseits verheimlicht, wie der aktuelle Compiler denn nun wirklich parametrisiert ist, und bei der Fehlersuche kommt ausser einem pauschalen „Geht irgendwie nicht“ kaum eine vernünftige Fehlermeldung, von der aktbackenen Bedienung mal ganz zu schweigen.

Deutlich mehr Spaß macht es da, seine Arduino- Projekte mit der [VSCode Arduino](#) Extension in VSCode zu machen, aber auch da liegt der Teufel im Detail. Die [Installation](#) geht ja noch einigermaßen vonstatten, auch wenn die meisten Anleitungen zusätzlich noch PlatformIO verlangen und man damit von der einen (Arduino-) Abhängigkeit in die nächste rüberwechselt, darum sei hier die Installation ohne PlatformIO empfohlen.

Es gibt aber noch mindestens zwei weitere Stolpersteine:

## Die initiale Erzeugung der `c_cpp_properties.json`

Die Extension erzeugt sich die initiale Konfiguration selber, sobald man das erste Mal compiliert. Klingt gut, geht aber schrecklich in die Hose, wenn man nicht definitiv ein lauffähiges (Beispiel-) Programm geladen hat. Dann gibt's so'ne Art Phantasie- Konfiguration, die auch weiterhin gültig bleibt, während man versucht, die Fehler zu finden.

Ausserdem will der Konfigurator, dass die \*.ino- Datei genau so heißt wie der Ordner, in dem sie sich befindet.

In den `File/Preferences/Extensions` sollte man für die Arduino- Extension den Log- Level auf `Verbose` setzen, damit man im Output- Window miterleben kann, was Compiler und esptool da so alles anstellen. Ohne dies ist das Output- Window sehr langweilig...

## externe Libraries immer über den Library- Manager

Man ist versucht, der Einfachheit halber über die Include- Pfade bereits bestehende lokale Libraries seiner bisherigen Arduino- Installation einzubinden. Das klappt auch scheinbar, solange bis diese dann wieder andere Includes einbinden, dann weiß der Compiler scheinbar nicht mehr, wo er suchen soll.

Darum Libraries besser nochmal für VSCode über dessen Library- Manager neu installieren, das geht wohl besser.

## Das heißt dann

Darum bei einem neuen Projekt:

- mit einem leeren Verzeichnis anfangen
- erstmal Board & Board- Config ausfüllen
- Libraries installieren, insofern schon bekannt
- mit etwas Lauffähigem wie „Hello World“ starten, damit die Konfiguration initial richtig erzeugt wird
- und dann erst das Programm in kleinen Schritten erweitern und immer wieder mal mit „Verify“ prüfen, ob der Compiler noch nachkommt.

From:

<http://koehlers.de/wiki/> - **Steffen Köhlers Online- Bastelbuch**

Permanent link:

<http://koehlers.de/wiki/doku.php?id=smarhome:arduinoovscode>

Last update: **2023/04/16 11:16**

