


## Kompatibilitätsprobleme beheben


Soweit wir wissen, treten die Kompatibilitätsprobleme nur im Zusammenhang mit dem I<sup>2</sup>C-Interface auf. Abhilfe schafft eine neue Bibliothek, die [hier](#) als ZIP-Datei heruntergeladen werden kann.


---


Nach dem Herunterladen muss in einem nächsten Schritt die Datei entpackt werden und der gesamte Ordner

 LiquidCrystal\_I2C\_AvrI2C

mit den Dateien

 AvrI2c\_Greiman

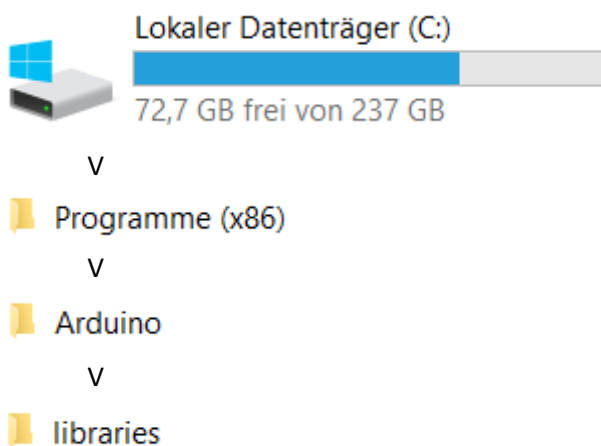
 LiquidCrystal\_I2C\_AvrI2C

 LiquidCrystal\_I2C\_AvrI2C

zu den übrigen Arduino-Bibliotheken verschoben werden.


---

Üblicherweise befinden sich diese in dem Ordner <libraries>, zu dem man sich wie folgt durchklicken kann:



---

Nun den oben bereits erwähnten Ordner

 LiquidCrystal\_I2C\_AvrI2C

mit Inhalt kopieren und in den Arduino-Ordner „libraries“ einfügen.

---

Programme auf unseren Seiten, die eine I<sup>2</sup>C-Bibliothek benötigen, zeigen gleich zu Beginn folgende Einträge:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

Diese Einträge müssen durch

```
#include <AvrI2c_Greiman.h>
#include <LiquidCrystal_I2C_AvrI2C.h>
```

ersetzt werden.

Weiterhin muss der Eintrag

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

durch

```
LiquidCrystal_I2C_AvrI2C lcd(0x27,16,2);
```

ersetzt werden.

---

Die auf diese Weise neu eingebundene Bibliothek kennt nicht alle Befehle der vorherigen. Das wird spätestens beim Kompilieren erkennbar, wenn die IDE bei „lcd.init();“ hängenbleibt. Der passende Befehl „lcd.begin();“ löst das Problem.

---

Soweit wir wissen, sollten mit diesen Veränderungen unsere Programme auch mit der zur Zeit neuesten Arduino-Version 1.6.7 ordnungsgemäß funktionieren.