

Projekt: \_\_\_\_\_

Entwickler: \_\_\_\_\_

Gruppe: \_\_\_\_\_

# Checkliste für das Codereview

- **1. Es werden aussagekräftige (Variablen-, Unterfunktions-)Namen genutzt**
  - 1.1 Namen aus nur einem Buchstaben werden vermieden; diese sind i.d.R. nicht aussagekräftig.
  - 1.2 Es werden aussagekräftige Variablennamen aus einem oder mehreren Worten genutzt.
  - 1.3 Je größer der Gültigkeitsbereich einer Variable, desto aussagekräftiger sollte der Name sein.
  - 1.4 Namen von Funktionen sind aus mehreren Worten mit Verb zusammengesetzt (z.B. initLCD).
  - 1.5 Es wird bei Namen nur eine Sprache verwendet (Englisch oder Deutsch).
- **2. Variablen werden sinnvoll genutzt**
  - 2.1 Alle Variablen werden initialisiert.
  - 2.2 Der Wertebereich der Variablen wird beachtet (kein `uint8_t test = 300`).
  - 2.3 Für boole Werte wird nur 0 (Falsch), und `!=0` (Wahr) genutzt, kein 't', 'f' oder -1, 1.
  - 2.4 Variablen dienen jeweils nur einer Funktion.
  - 2.5 Die Funktion von Variablen sind nicht überladen.
- **3. Explizite Zahlenwerte im Code werden vermieden**
  - 3.1 Es werden maximal die Zahlen 0 und 1 in Funktionen genutzt
  - 3.2 (für andere sind Konstanten oder `#defines` angelegt).
  - 3.3 Länge von Strings sind nicht fest eingetragen, sondern durch `sizeof()` and `strlen()` ermittelt.
  - 3.4 Konstante und `#defines` liegen in einen header file.
- **4. Es wurde ausreichend modularisiert**
  - 4.1 `main()` ist schlank gehalten.
  - 4.2 Ähnliche Codeabfolgen sind zu Unterfunktionen, Schleifen, Arrays o.ä. zusammengefasst.
  - 4.3 Funktionen enthalten i.d.R. nicht mehr als 50 Codezeilen.
  - 4.4 Jede Funktion ist für genau eine Aufgabe zuständig.
  - 4.5 Eingabe, Verarbeitung und Ausgabe sind gut separiert.
- **5. Das Programm ist leserlich**
  - 5.1 Der beschreibende Kommentar am Anfang des Programms ist korrekt und vollständig (z.B. Angabe der verwendeten HW).
  - 5.2 Einrückungen werden für die unterschiedlichen Ebenen verwendet.
  - 5.3 Es werden nicht zu viele Leerzeilen genutzt, i.d.R. reicht eine Leerzeile.
  - 5.4 Kommentare sind verständlich; es ist kein Code auskommentiert.
  - 5.5 Kommentare enthalten keine Fehler in Grammatik und Rechtschreibung.
- **6. Verzweigungen wurden optimiert**
  - 6.1 Bei gegenseitig ausschließende Bedingungen wurde `if-then-else-if` genutzt.
  - 6.2 Die Programmteile im `if-then-else` beinhalten keine gleichen Anfangs- oder Endanweisungen.
  - 6.3 Wo es geht wird `switch-case` statt `if-then-else` genutzt.
  - 6.4 Bei `switch-case` Anweisungen ist der default Fall berücksichtigt
  - 6.5 und die korrekte Verwendung von `break` geprüft worden.
  - 6.6 `goto` wird nicht verwendet.

- **7. Programmablauf wurde optimiert**
  - 7.1 Es werden statt `_delay_ms()` / `_delay_us()` Interrupts, Timer oder Interrupt-gesteuerte Takte genutzt.
  - 7.2 ISR enthalten nur wenig Code (aufgerufene Unterfunktionen berücksichtigen!).
- **8. Das Programm läuft korrekt**
  - 8.1 Die Dateien sind vollständig und ausreichend zum kompilieren.
  - 8.2 Es gibt keine Fehler oder Warnings beim kompilieren.
  - 8.3 Das Programm ist lauffähig und tut was es soll.
  - 8.4 Das Programm ist ästhetisch und interessant umgesetzt.

## Freitext