



<h1>Software Review</h1> <p>Projektname:</p> <p>Modul:</p>	Version:	
	Reviewer:	
	Datum:	

1 SW-Review Zusammenfassung

Notwendige Änderungen:

2 Allgemein

- | | | |
|-----------------------|--|-----------------------------------|
| 2.1 (advisory) | Ist der Header der Files und jeder Funktion beschreibend genug | Approved <input type="checkbox"/> |
| 2.2 (advisory) | Jedes File kompiliert alleine ohne Fehler/Warnings (auch Header-Files) | Approved <input type="checkbox"/> |
| 2.3 (advisory) | Revision History enthält alle Änderungen | Approved <input type="checkbox"/> |



2.4 (advisory)	Gibt es unerreichbare Codeabschnitte (auskommentierter Code oder nicht erreichbarer Code) welche entfernt werden sollten	Approved <input type="checkbox"/>
----------------	--	-----------------------------------

2.5 (advisory)	Musste der Autor gefragt werden was der Code macht (Code sollte selbsterklärend sein)	Approved <input type="checkbox"/>
----------------	---	-----------------------------------

3 Comments

3.1 (advisory)	Entspricht der Kommentar dem Code	Approved <input type="checkbox"/>
----------------	-----------------------------------	-----------------------------------

3.2 (advisory)	Jede Funktion beschreibt die notwendigen Paramter (vor allem jene welche verändert werden) und mögliche Funktions-Abhängigkeiten	Approved <input type="checkbox"/>
----------------	--	-----------------------------------

3.3 (advisory)	Kein überflüssiger Kommentar (z.B. <code>i++;</code> <code>// increment i</code>)	Approved <input type="checkbox"/>
----------------	--	-----------------------------------

4 Coding Standards

4.1 (advisory)	Konstanten und Literale sind nicht hard coded	Approved <input type="checkbox"/>
----------------	---	-----------------------------------

4.2 (advisory)	Debug-Fehlermeldungen sind verständlich und komplett	Approved <input type="checkbox"/>
----------------	--	-----------------------------------

4.3 (advisory)	Die Struktur des Codes wird durch Einrücken verdeutlicht (keine Hard-Tabs)	Approved <input type="checkbox"/>
----------------	--	-----------------------------------

4.4 (advisory)	Klammern schaffen Klarheit und werden grosszügig eingesetzt	Approved <input type="checkbox"/>
----------------	---	-----------------------------------

4.5 (advisory)	Der Code-Style ist innerhalb des Moduls konsistent und gut strukturiert	Approved <input type="checkbox"/>
----------------	---	-----------------------------------

4.6 (advisory)	Assembler-Code wird geeignet gekapselt (innerhalb Macro/Funktion oder seperatem Assembler-File)	Approved <input type="checkbox"/>
----------------	---	-----------------------------------

4.7 (advisory)	Es gibt keine auskommentierten Codeabschnitte (es werden <code>#if</code> or <code>#ifdef</code> Preprocessorbefehle eingesetzt)	Approved <input type="checkbox"/>
----------------	--	-----------------------------------

4.8 (advisory)	<code>goto</code> oder <code>continue</code> Befehle werden nicht verwendet	Approved <input type="checkbox"/>
----------------	---	-----------------------------------



4.9 (advisory)	Makros mit Argumenten wurden durch inline Funktionen ersetzt	Approved <input type="checkbox"/>
----------------	--	-----------------------------------

5 Kontrollstrukturen

5.1 (required)	Korrekte Klammersetzung bei Blöcken von Schleifen und Verzweigungen	Approved <input type="checkbox"/>
----------------	---	-----------------------------------

5.2 (required)	Schleifen: <ul style="list-style-type: none">• Endbedingung von Schleifen überprüft• Schalufen-Zähler werden korrekt initialisiert (vor der Schleife)• Alle Schleifen-Variablen werden vor der Schleife initialisiert	Approved <input type="checkbox"/>
----------------	---	-----------------------------------

5.3 (required)	Switch Verzweigung: <ul style="list-style-type: none">• Gibt es einen default case• Jeder non-empty switch case wird mit einem break abgeschlossen	Approved <input type="checkbox"/>
----------------	---	-----------------------------------

5.4 (required)	if ... else if Verzweigung: <ul style="list-style-type: none">• Werden die meist eintreffenden Fälle zuerst getestet• Wird der else Fall behandelt	Approved <input type="checkbox"/>
----------------	---	-----------------------------------

5.5 (advisory)	Der Geltungsbereich von Kontrollstrukturen (switch, while, do ... while, for, if, else if und else) wird durch geschweifte Klammern verdeutlicht	Approved <input type="checkbox"/>
----------------	---	-----------------------------------

6 Variablen

6.1 (required)	Alle Variablen werden vor der Verwendung initialisiert	Approved <input type="checkbox"/>
----------------	--	-----------------------------------

6.2 (required)	Haben alle Variablen den korrekten Typ oder Cast	Approved <input type="checkbox"/>
----------------	--	-----------------------------------

6.3 (required)	Geltungsbereich von Variablen: <ul style="list-style-type: none">• Wurden globale Variablen minimal eingesetzt• Haben alle Variablen den kleinst möglichen Geltungsbereich	Approved <input type="checkbox"/>
----------------	---	-----------------------------------



6.4 (advisory)	Werden globale Variablen korrekt initialisiert (behalten diese bei einem Reset den notwendigen Wert)	Approved <input type="checkbox"/>
6.5 (advisory)	Gibt es redundante oder unbenutzte Variablen	Approved <input type="checkbox"/>
6.6 (advisory)	Alle Variablen haben einen klaren und beschreibenden Namen	Approved <input type="checkbox"/>

7 Datentypen

7.1 (required)	Verwendung von portablen Datentypen (z.B. uint32_t)	Approved <input type="checkbox"/>
7.2 (required)	Wurde das Vorzeichen des Datentyps bei der Deklaration spezifiziert (Verwendung von unsigned/signed)	Approved <input type="checkbox"/>
7.3 (required)	Wurden Datentypen mit typedef deklariert	Approved <input type="checkbox"/>

8 volatile

9.1 (required)	Alle Memory-Mapped Peripherie Register sind als volatile deklariert	Approved <input type="checkbox"/>
9.2 (required)	Globale Variablen welche in ISR verwendet werden sind als volatile deklariert	Approved <input type="checkbox"/>
9.3 (required)	Globale Variablen welche aus unterschiedlichen Tasks verwendet werden sind als volatile deklariert (auch wenn durch Mutex/Semaphore geschützt)	Approved <input type="checkbox"/>

9 Array

10.1 (required)	Array Indizes sind innerhalb der definierten Grenzen	Approved <input type="checkbox"/>
10.2 (required)	Wird das Verlassen der Bereichsgrenzen von Array-Indizes überprüft (dies gilt auch für Pointer)	Approved <input type="checkbox"/>
10.3 (required)	Auf μ Controller keine grossen lokalen Arrays (Stackoverflow)	Approved <input type="checkbox"/>



10 Logic/Arithmetic

- | | | |
|-----------------|--|-----------------------------------|
| 11.1 (required) | Division durch Null abfangen/überprüft | Approved <input type="checkbox"/> |
| 11.2 (required) | Kein Vergleich von Floating-Point Zahlen auf Gleichheit | Approved <input type="checkbox"/> |
| 11.3 (required) | Wird dynamisch allozierter Speicher wieder korrekt freigegeben (nur allozierter Speicher freigegeben) | Approved <input type="checkbox"/> |
| 11.4 (required) | Der Zugriff auf externe Devices wird durch Timeouts geschützt | Approved <input type="checkbox"/> |
| 11.5 (required) | Gibt es eine Chance auf einen mathematischen Overflow / Underflow (ist dies abgefangen) | Approved <input type="checkbox"/> |
| 11.6 (required) | Ist der Einsatz von Bitweise, Relationale und Logischen Operatoren korrekt (korrekter Einsatz von ==, =, &&, &, etc.; korrekt geklammert (a & 0x01) == 0 → Gleich bindet mehr als &) | Approved <input type="checkbox"/> |

11 Devensive Programmierweise

- | | | |
|-----------------|---|-----------------------------------|
| 12.1 (required) | Werden Parameter beim Funktionseintritt auf ihre Gültigkeit überprüft (sanity checking) | Approved <input type="checkbox"/> |
| 12.2 (required) | Rückgabewerte von Funktionen (vor allem Fehlermeldungen) werden überprüft | Approved <input type="checkbox"/> |
| 12.3 (required) | Wird auf mögliche NULL-Pointer hin getestet | Approved <input type="checkbox"/> |

12 Funktionen

- | | | |
|-----------------|---|-----------------------------------|
| 13.1 (required) | Werden alle lokalen Funktionsvariablen zu Beginn initialisiert | Approved <input type="checkbox"/> |
| 13.2 (required) | Werden Fehlerwerte generiert und der aufrufenden Funktion zurückgegeben | Approved <input type="checkbox"/> |
| 13.3 (required) | Ist die Funktion im minimalen Geltungsbereich definiert | Approved <input type="checkbox"/> |



13.4 (advisory)	Alle Funktionen haben einen klaren und beschreibenden Namen	Approved <input type="checkbox"/>
13.5 (advisory)	Gibt es nicht verwendete Funktionen welche gelöscht werden könnten	Approved <input type="checkbox"/>
13.6 (advisory)	Sind Funktionen nicht zu komplex (allenfalls Aufteilung in mehrere Funktionen; Richtwert: ausdrückt max. 1 A4-Seite)	Approved <input type="checkbox"/>
13.7 (advisory)	Gibt es redundanten Code welcher in einer Funktion gekapselt werden kann	Approved <input type="checkbox"/>
13.8 (advisory)	Jede Funktion hat nur einen Exit-Punkt (keine return Statement in der Mitte der Funktion)	Approved <input type="checkbox"/>

13 Multithreading

14.1 (required)	Sind Task-Prioritäten sinnvoll gewählt	Approved <input type="checkbox"/>
14.2 (required)	Kontrolle der Stackauslastung der einzelnen Tasks	Approved <input type="checkbox"/>
14.3 (required)	Semaphoren/Mutex: <ul style="list-style-type: none">• Zugriff auf kritische Abschnitte ist geschützt• Jede angeforderte Semaphore/Mutex wird wieder zurückgegeben• Alle Semaphoren/Mutex werden in der selben Reihenfolge angefordert und in umgekehrter Reihenfolge zurückgegeben (Vermeidung von Deadlocks)• Zum Schutz von kritischen Abschnitten werden Mutexe verwendet (Vermeidung von Priority Inversion)• Es wird kein Task gelöscht welcher eine Semaphore/Mutex belegt• Rekursive Mutexe werden korrekt eingesetzt (Verschachtelung)	Approved <input type="checkbox"/>
14.4 (required)	Es werden keine non-reentrant Funktionen aus den Tasks aufgerufen (Vorsicht bei Library-Funktionen)	Approved <input type="checkbox"/>



14 Interrupt Service Routine

- | | | |
|-----------------|---|-----------------------------------|
| 15.1 (required) | Innerhalb von ISR werden nur geeignete RTOS-API Funktionen aufgerufen | Approved <input type="checkbox"/> |
| 15.2 (required) | Es werden keine non-reentrant Funktionen aus der ISR aufgerufen (Vorsicht bei Library-Funktionen) | Approved <input type="checkbox"/> |
| 15.3 (advisory) | ISR ist so kurz als möglich gehalten | Approved <input type="checkbox"/> |

15 μ Controller-Peripherie

- | | | |
|-----------------|--|-----------------------------------|
| 16.1 (required) | Watchdog: <ul style="list-style-type: none">• Sinnvolles Timeout gewählt• Reset des Watchdogs an geeigneten Orten (nicht zu tief verschachtelt)• Initialisierungsvorgang durch Watchdog geschützt• Spricht der Watchdog an (Test durch forcieren eines Watchdog-Resets an einigen Code-Stellen) | Approved <input type="checkbox"/> |
|-----------------|--|-----------------------------------|