Modul-Design SW Watchdog

Für das studentische Projekt *Sichere Eisenbahnsteuerung*

Datum	20.05.2010
Quelle	Dokumente\02_Design\02.02_Moduldesign
Autoren	F. Geber
Version	2.1
Status	freigegeben

1 Historie

Version	Datum	Autor	Bemerkung
0.1	09.12.2009	Felix Blüml	Erste Version
0.2	22.12.2009	Felix Blüml	Kap. 5.2: Änderung der Groß-/Kleinschreibung von SW_status_array Dokumentenübergreifend: Änderung des Schnittstellennamens von sw_init() in initSW()
0.3	22.12.2009	Felix Blüml	Anpassung von Abb. 1 gemäß den Änderungen aus Version 0.2
0.4	22.12.2009	Felix Blüml	Anpassung von Abb. 2 gemäß den Änderungen aus Version 0.2
0.5	22.12.2009	Felix Blüml	Dokumentenübergreifend: Änderung von „unsigned integer 8 Bit“ in „unsigned char“
1.0	22.12.2009	Felix Blüml	(Review durch Thomas Musialslci.) Dokumentenübergreifend: Anpassungen der Formulierungen zur besseren Verständlichkeit und Rechtschreibfehlerkorrektur
1.1	12.01.2010	Felix Blüml	Dokumentenübergreifend: Änderung der Schnittstelle von „hello(module_id, status)“ in „helloModul(module_id, status)“. Hinzufügen der Schnittstelle stopSW().
1.2	12.01.2010	Felix Blüml	Kap. 3.2.1: Vervollständigung des Textes durch das Wort „freigegeben“. Kap. 6.1: Korrektur des maximalen Zählerstandes und Abhängigkeiten. Kap. 6.2.4: TR durch TR0 korrigiert.
1.3	26.01.2010	Felix Blüml	Kap. 3.2.1: Änderung des Ablaufs; Timer0 wird zu Beginn erst einmal gestoppt. Korrektur der Vergabe der höchsten Priorität. Kap 3.1: Hinzufügen des gemessenen Wertes des Zeitfensters.
1.4	04.02.2010	Felix Blüml	Kap 3.1: Entfernen des gemessenen Wertes des Zeitfensters. Berechnung für 10 MHz umgestellt.
1.5	07.02.2010	Felix Blüml	Abb.1: Korrektur des Kommentars.
1.6	07.02.2010	Felix Blüml	Korrektur der fehlerhaften Nummerierung der Kapitel.
2.0	06.05.2010	F. Geber	Layout-Anpassung; Aktualisieren der referenzierten Dokumenten (Kapitel 4), C515C ist von Infineon nicht Siemens; Text unter Bildern zu Bildbeschriftungen

			umwandeln und im Text richtig referenzieren; Umbenennen von BMV in BV (Betriebsmittelverwaltung), da es so Konvention ist; in Kapitel 5.2: SW_status_array ist 7 Byte lang, nicht 6; in Kapitel 5.3: Genaue Festlegung des Aufrufs von initSW() nach initNOTAUS(); Vollendung des letzten Satzes in Kapitel 6; in Kapitel 6.2.1: Im Quellcode wird zuerst das Array befüllt, bevor TR0 = 0 gesetzt wird;
2.1	20.05.2010	F. Geber	Fehler in der letzten Änderung (Abkürzung von Betriebsmittelverwaltung ist BMV nicht BV); Änderung in 5.4.4: Tabellenkopf falsch und Vorbedingung falsch; 6.1: Rechtschreibfehler in der ersten Formel & Variable falsch benannt

2 Inhaltsverzeichnis

3 Einleitung

Dieses Dokument beschreibt das Modul Software-Watchdog (SW) des Software-Designs.

Der SW soll Deadlocks im Programmablauf erkennen und ggf. das Modelleisenbahnsystem mit Hilfe eines weiteren Moduls, dem Not-Aus-Treiber des Software-Designs, stoppen können.

Deadlocks sollen durch zwei Prüfungen identifiziert werden:

- Zu hoher Zeitbedarf bei Bewältigung einer Aufgabe und
- Wiederholte Ausführung einer gleichen Aufgabe.

Zur Umsetzung der ersten Prüfmethode durch den SW wird der Hardwarezähler Timer/Counter 0 des Mikrocontrollers C515C verwendet, wie im Data Sheet (siehe Referenzierte Dokumente) beschrieben.

4 Referenzierte Dokumente

- Dokumente\02_Design\02.01_Subsystemdesign\Software-Design
- Dokumente\02_Design\02.02_Moduldesign\Not-Aus-Treiber-Modul-Design
- http://www.datasheetcatalog.com/datasheets_pdf/C/5/1/5/C515C.shtml (Infineon Technologies AG C515C Data Sheet Version 07.97)

5 Architektur

In den folgenden Unterkapiteln wird die Architektur des Moduls näher erläutert. Siehe dazu auch die Abbildung 1.

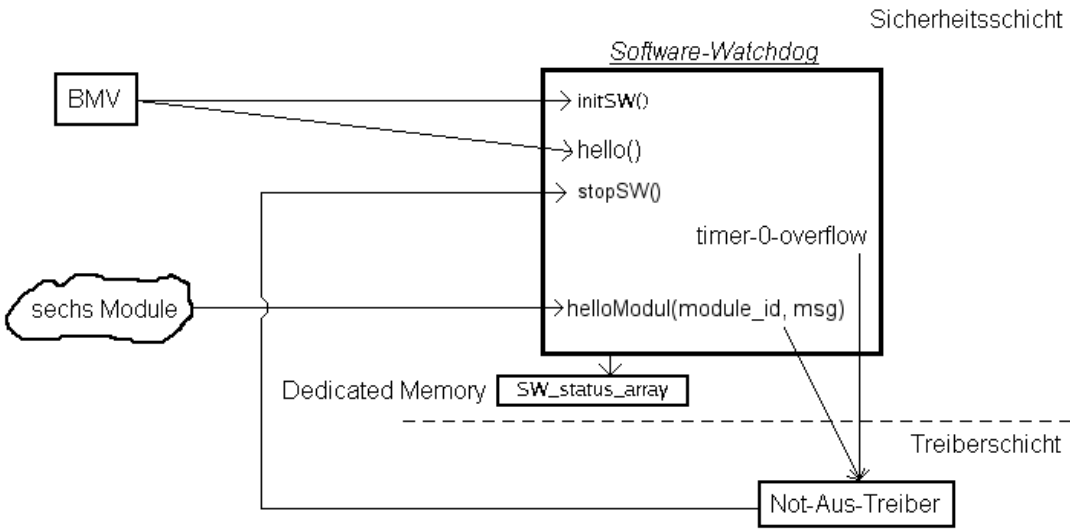

Abbildung 1: Schnittstellen und globale Variablen des SW

5.1 Funktionshierarchie

Der Software-Watchdog ist im Software-Design in der Sicherheitsschicht angesiedelt. Es ist jedoch gestattet und gefordert, dass die Schnittstellen in einigen Fällen schichtenübergreifend aufgerufen werden. Aufrufe erfolgen durch die Module selbst, u. A. auch durch die Betriebsmittelverwaltung (BMV) und den Not-Aus-Treiber (siehe dazu auch die Architektur des Software-Designs).

Es wird der Interrupt Timer-0-Overflow des Mikrocontrollers C515C verwendet. Dieser wird im Falle eines Zählerüberlaufs des Hardwarezählers Timer-0 ausgelöst und übernimmt die weitere Programmsteuerung.

5.2 Daten

Zum Speichern der Status der sechs zu überwachenden Module wird ein globales Array, bestehend aus 7 Byte, benötigt: (unsigned char) *SW_status_array* [7]. Dabei ist jedes Element einem Modul zugeordnet.

5.3 Abhängigkeiten von anderen Modulen

Für die korrekte Verwendung der Schnittstellen HelloModul *(void helloModul(byte module_id, byte status))* und Hello *(void hello())* muss das Modul Betriebsmittelverwaltung vorher (möglichst gleich zu Beginn der Laufzeit) die Schnittstelle initSW *(void initSW())* aufrufen (geschieht gleich nach *initNOTAUS())*.

Der Software-Watchdog ist so ausgelegt, dass im Fehlerfall die Schnittstelle Emergency_off *(void emergency_off())* des Moduls Not-Aus-Treiber aufgerufen wird. Dadurch soll das

Modelleisenbahnsystem abrupt ausgeschaltet werden können. Dazu muss der Not-Aus-Treiber seinerseits wiederum die Schnittstelle StopSW *(void stopSW())* kontaktieren.

5.4 Schnittstellenbeschreibung

Das Modul bietet die unten beschriebenen Schnittstellen.

5.4.1 void initSW()

Dies ist eine Schnittstelle nur für das Modul Betriebsmittelverwaltung. Die Schnittstelle dient der Initialisierung eines Speicherfeldes für die zuletzt gemeldeten Status der Module und der 'Scharfschaltung' des Hardwarezählers.

Name:	initSW
Vorbedingung:	Diese Schnittstelle muss einmal zu Beginn der Laufzeit aufgerufen werden.
Parameter:	Keine.
Rückgabe:	Keine.
Nachbedingung:	Das Speicherfeld für die zuletzt gemeldeten Status der Module wurde initialisiert und der Hardwarezähler ist aktiv.
Fehlerfall:	Wurde diese Schnittstelle zur Laufzeit erst später aufgerufen, so werden dadurch alle bisher gesammelten Status verworfen und die Vorbedingungen bzgl. Initialstatus aus void helloModul(byte module_id, byte status) und void hello() werden nicht erfüllt.

5.4.2 void hello()

Diese Schnittstelle ist ebenfalls nur für das Modul Betriebsmittelverwaltung vorgesehen. Die Schnittstelle dient der Rückmeldung innerhalb des vorgegebenen Zeitfensters (siehe dazu den festgelegten Wert in Kapitel 6.1) zur Bewältigung von Aufgaben.

Name:	hello
Vorbedingung:	*initSW()* muss vorher aufgerufen worden sein. *hello()* muss innerhalb des vorgegebenen Zeitfensters aufgerufen werden.
Parameter:	Keine.
Rückgabe:	Keine.
Nachbedingung:	Der Hardwarezähler Timer-0 wird zurückgesetzt.
Fehlerfall:	Wurde *hello()* nicht innerhalb des vorgegebenen Zeitfensters aufgerufen, so wird das Modul Not-Aus-Treiber kontaktiert.

	Wurde *initSW()* vorher nicht aufgerufen, so wird der Interrupt timer-0-overflow nicht ausgelöst.

5.4.3 void helloModul(byte module_id, byte status)

Dies ist eine Schnittstelle mit Statusmeldungen für die Module mit zugeordneter Modul-ID (für den Parameter `module_id`):

- 0 – Leitzentrale,
- 1 – Befehlsvalidierung,
- 2 – Ergebnisvalidierung,
- 3 – S88-Treiber,
- 4 – SSC-Treiber,
- 5 – RS232-Treiber und
- 6 - Auditing System.

Die Schnittstelle dient der Überwachung der Tätigkeiten der Module.

Name:	helloModul
Vorbedingung:	Die Schnittstelle *initSW()* muss vorher aufgerufen worden sein. `module_id` darf nur Modul-IDs zwischen 0 <= x <= 6 erhalten. Der Initialstatus* eines Moduls mit `status=255` ist nicht erlaubt.
Parameter:	(unsigned char) module_id, (unsigned char) status
Rückgabe:	Keine.
Nachbedingung:	Der empfangene `status` der jeweiligen `module_id` wird in einem Array vermerkt.
Fehlerfall:	Ist `module_id` > 6, so wird das Modul Not-Aus-Treiber kontaktiert. Wird als Initialstatus* eines Moduls `status=255` gesetzt, so wird das Modul Not-Aus-Treiber kontaktiert. Wenn für jeweils eine `module_id` zwei Mal hintereinander der gleiche `status` empfangen wurde (unabhängig von anderen empfangen Statusmeldungen anderer Module in der Zwischenzeit) wird das Modul Not-Aus-Treiber kontaktiert. Wurde *initSW()* vorher nicht aufgerufen, kann es sein, dass der Not-Aus-Treiber kontaktiert wird.

* Die allererste Statusmeldung eines Moduls zur Laufzeit.

5.4.4 void stopSW()

Dies ist eine Schnittstelle nur für das Modul Not-Aus-Treiber. Die Schnittstelle dient zum Abschalten des Hardwarezählers.

Name:	stopSW
Vorbedingung:	Not-Aus-Treiber wurde aufgerufen.
Parameter:	Keine.
Rückgabe:	Keine.
Nachbedingung:	Der Hardwarezähler ist inaktiv.
Fehlerfall:	Nicht bekannt.

6 Dynamisches Verhalten

Im Folgenden wird das dynamische Verhalten aller, im Modul vorhandenen, Funktionen beschrieben. Doch zunächst einige Erläuterungen zu den Zeitfensterkonfiguration des Hardwarezählers.

6.1 Zeitfensterkonfiguration des Hardwarezählers

Der verwendete Hardwarezähler Timer-0 soll auf $t_{Zeitfenster} = 20$ ms gestellt werden. Der Wert ist ein Kompromiss aus der benötigten Zeit für eine Vollbremsung der Modelleisenbahnzüge und möglichst langer Rechenzeit für die Module.

Er wird als Zeitfenster für die Rückmeldung der Module verwendet.

Zur Ermittlung des Sollwertes für Timer-0 wird die folgende Formel verwendet:

$$SWT0STARTWERT = MaxZählerstand - ceil(\frac{t_{Zeitfenster} * f_{Taktrate}}{ZyklenProTakt})$$

Also z.B.:

$$SWT0STARTWERT = 65535 - ceil(\frac{0{,}02s * 10\text{MHz}}{6}) = 32202_D = 7\text{DCA}_H$$

Daraus ergeben sich die folgenden Werte für den Zählerstand des Timer-0:

$TL0 = CA_H$ und $TH0 = 7D_H$

Die oben gezeigte Ermittlung des Zählerstandes soll nicht erst zur Laufzeit, sondern schon vorher durch z.B. Precompilerfunktionen, gebildet werden.

6.2 Funktionsbeschreibung

6.2.1 initSW()

Nach Aufruf dieser Schnittstelle wird das Speicherfeld für die zuletzt gemeldeten Status der Module in Form eines globalen Arrays mit dem Wert FF_H befüllt.

Danach wird der Hardwarezähler Timer-0 des Mikrocontrollers C515C Timer-0 gestoppt. Anschließend wird der Interrupt Timer-0-Overflow freigegeben und Timer-0 wieder aktiviert.

Zur Verdeutlichung des oben genannten Vorgehens soll das Aktivitätsdiagramm in Abbildung 2 dienen:

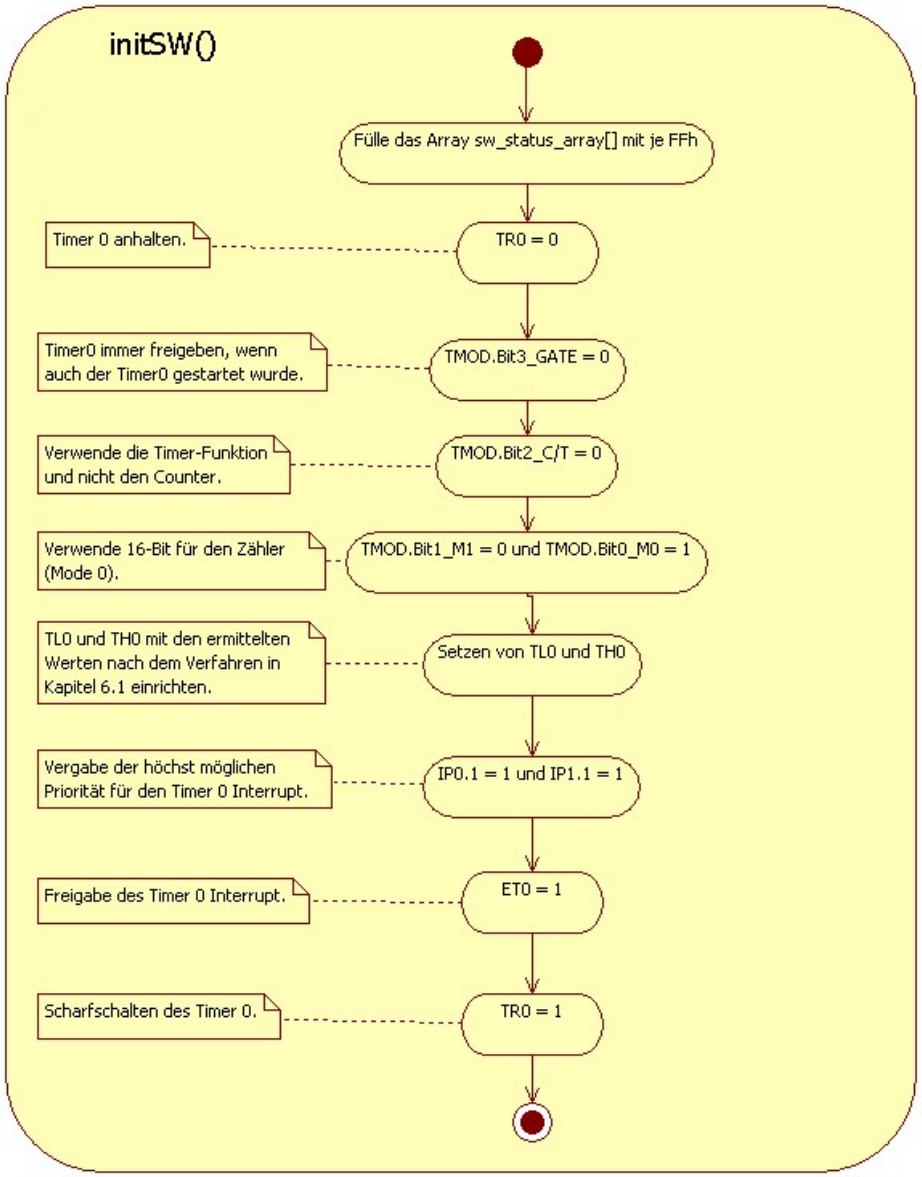

Abbildung 2: Aktivitätsdiagramm zu initSW()

6.2.2 hello()

Nach Aufruf dieser Schnittstelle wird der Hardwarezähler Timer-0 des Mikrocontrollers C515C zurückgesetzt. Siehe dazu das Aktivitätsdiagramm in Abbildung 3:

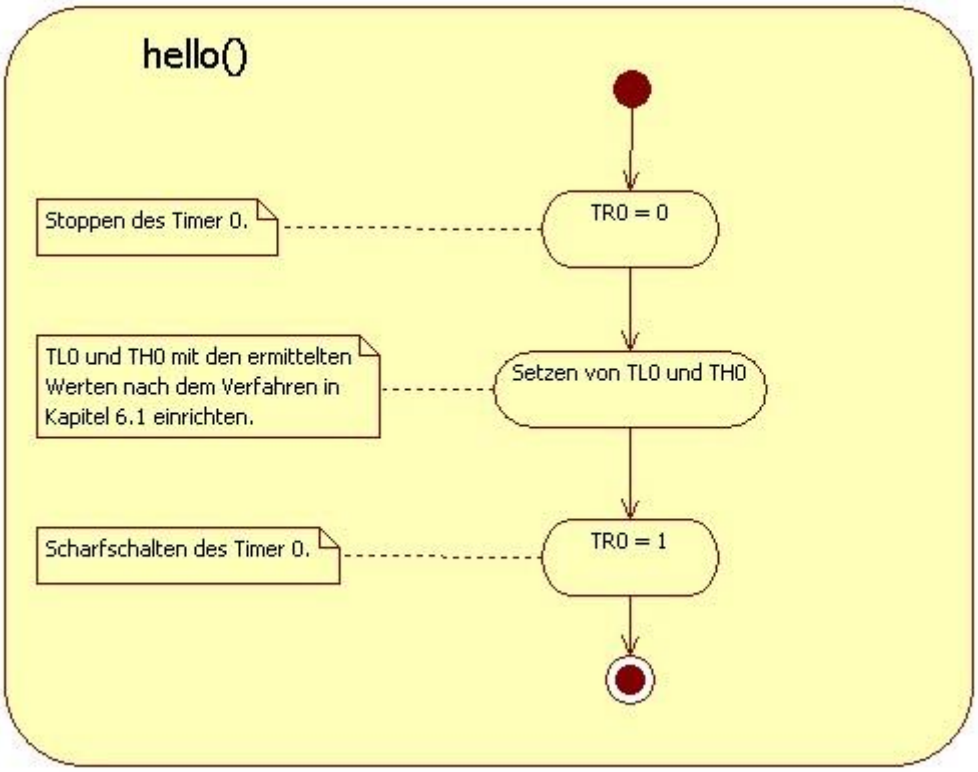

Abbildung 3: Aktivitätsdiagramm zu hello()

6.2.3 helloModul(module_id, status)

Nach Aufruf dieser Schnittstelle wird zunächst überprüft, ob die erhaltene Modul-ID ungültig ist. I.d.F. wird der Not-Aus-Treiber kontaktiert und somit der Programmfluss angehalten. Danach wird geprüft, ob der erhaltene Status bereits durch einen vorherigen Aufruf im globalen Array gesetzt wurde. I.d.F. wird ebenso der Not-Aus-Treiber kontaktiert. Sollten beide Prüfungen negativ verlaufen sein, so wird der erhaltene Status im globalen Array gespeichert. Siehe dazu das Aktivitätsdiagramm in Abbildung 4:

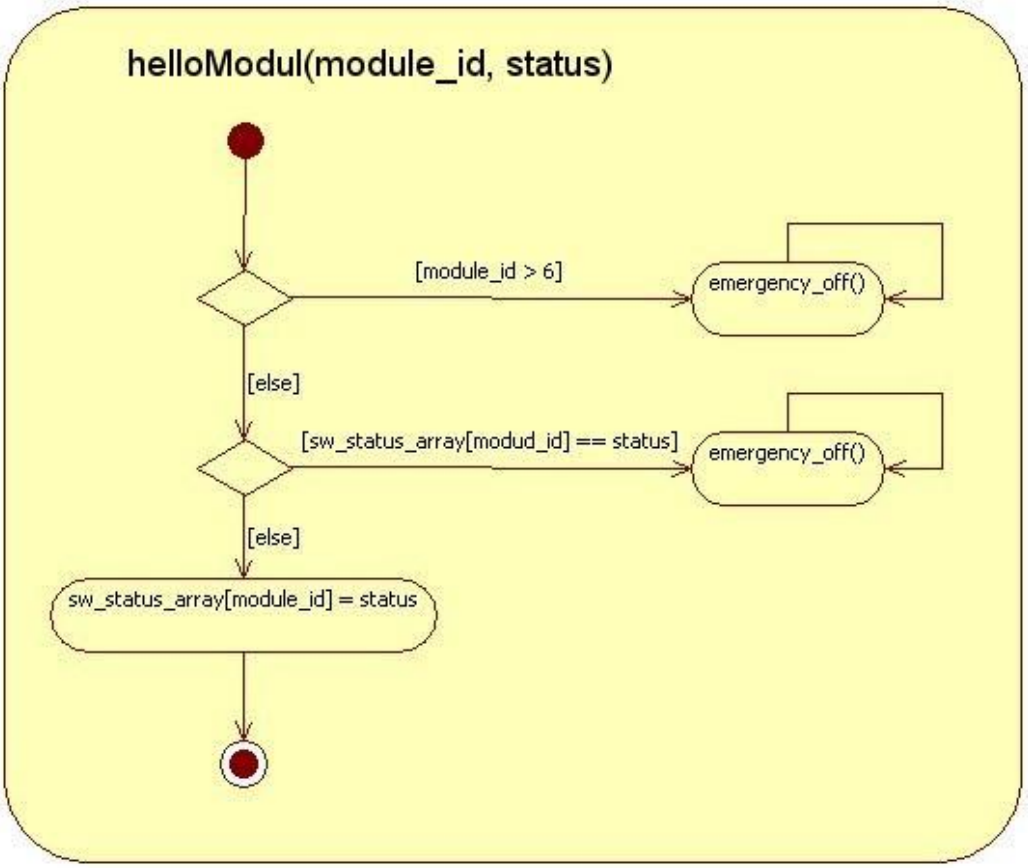

Abbildung 4: Aktivitätsdiagramm zu helloModul(module_id, status)

6.2.4 stopSW()

Nach Aufruf dieser Schnittstelle wird der Hardwarezähler Timer-0 des Mikrocontrollers C515C inaktiv geschaltet. Dazu wird TR0 = 0 gesetzt.

6.2.5 Interrupt timer-0-overflow

Wird dieser Interrupt ausgelöst, so wird der Not-Aus-Treiber kontaktiert und somit der Programmfluss angehalten. Siehe dazu das Aktivitätsdiagramm in Abbildung 5:

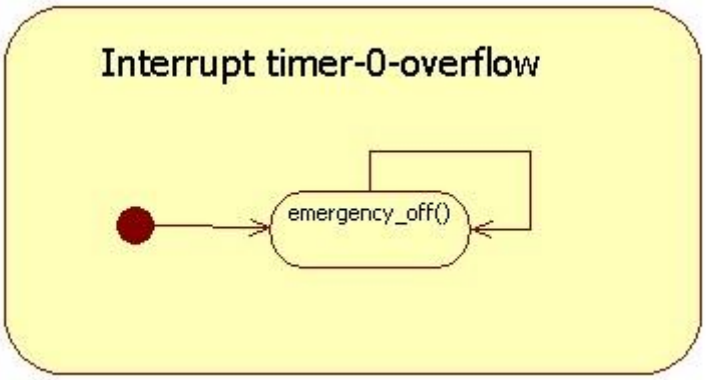

Abbildung 5: Aktivitätsdiagramm zu Interrupt timer-0-overflow