

# CT Praktikum: Interrupt Latency

## 1 Einleitung

In diesem Praktikum messen Sie die Latenz eines Interrupts. Die Latenz ist die Zeit vom Auslösen des Interrupts bis zum Start der dazugehörigen ISR (Interrupt Service Routine).

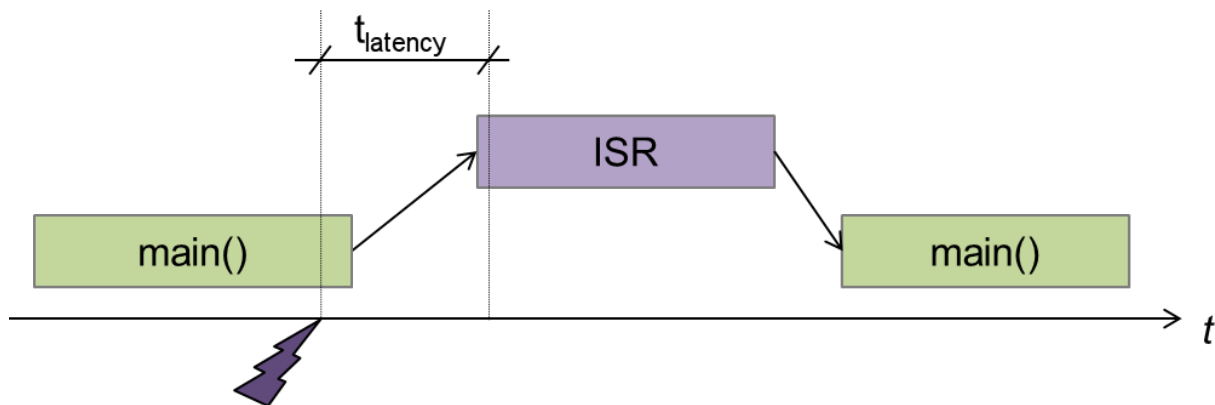


Abbildung 1: Latenz eines Interrupts

Die Latenz lässt sich einfach mit einem Timer messen. Der Timer löst beim Überlauf einen Interrupt aus und läuft dabei konstant weiter. Wird das Zählregister direkt zu Beginn der Timer-ISR ausgelesen, kann damit die Latenz gemessen werden. Das Zählregister enthält dann die Anzahl Ticks, die seit dem Auslösen des Interrupts gezählt wurden. (Dabei wird hier vernachlässigt, dass nicht nur die Interrupt Latenz gemessen wird, sondern auch noch die Zeit, die benötigt wird um den Zähler auszulesen.)

## 2 Lernziele

- Sie können den Begriff der Interrupt Latenz erklären und mögliche Ursachen nennen.
- Sie sind in der Lage, ein Programm zu realisieren um die Interrupt-Latenz zu messen.
- Sie verstehen wie andere, höher priorisierte Interrupts die Latenz eines Interrupts verlängern können.

## 3 Aufgaben

In diesem Praktikum werden zwei Interrupt-Quellen verwendet:

- Quelle 1: Timer 2 ist als Upcounter konfiguriert und wird bei Erreichen des Reload-Wertes auf null zurückgesetzt. Er ist so konfiguriert, dass er alle 1 ms zurückgesetzt wird und einen periodischen Interrupt erzeugt.
- Quelle 2: Timer 3 dient zur Erzeugung einer Lastsituation. Mittels der „defines“ zu Beginn des Programmcodes kann die Last eingestellt werden (z.B. `#define LOAD_WITH_10KHZ`). Wird keine Last ausgewählt (`#define LOAD_NONE`), so wird Timer 3 nicht gestartet.

Timer 2 und Timer 3 sind bereits konfiguriert. Sie können die Konfigurationen im vorgegebenen Quelltext finden.

### 3.1 Interruptroutine zur Messung der Latenz

Zunächst sollen Sie die Latenz des Timer Interrupts messen. Timer 2 erzeugt bei Überlauf einen Interrupt. Dazu soll **zu Beginn** der entsprechenden ISR der Zählerstand des Timer 2 ausgelesen werden. Um eventuelle Schwankungen zu eliminieren, sollen mehrere Durchläufe gemittelt werden.

- a) Ergänzen Sie die gegebene ISR für den Interrupt des Timers 2. Am Anfang der ISR soll der Zählerstand ausgelesen werden. Danach sollen der Durchschnittswert, der Minimalwert und der Maximalwert der Latenz berechnet werden. Verwenden Sie dazu die vordefinierten Variablen: `min_latency`, `max_latency`, `avg_latency`. Sobald die definierte Anzahl Durchläufe erreicht ist (vordefinierte Konstante `NUMBER_OF_TIMER_2_INTERRUPTS`) soll die zugehörige Boolesche Variable auf `TRUE` gesetzt werden. Denken Sie daran, vor der Berechnung, den IRQ des Timers 2 zurückzusetzen, `hal_timer_irq_clear(TIM2, HAL_TIMER_IRQ_UE)` ;
- b) Testen Sie die Funktionalität Ihres Codes mit dem Debugger und lesen Sie die drei Werte aus. Wie groß ist die Latenz minimal, maximal und im Durchschnitt? Tragen Sie den abgelesenen Zählerstand in der unten stehenden Tabelle ein.

	Minimum	Maximum	Durchschnitt
Latenz	21	28	21

### 3.2 Ausgabe der Werte auf dem Display

Geben Sie die Werte für Durchschnitt, Minimum und Maximum auf dem LCD des CT-Boards aus. Ergänzen Sie dazu die vorbereitete Routine `print_results()`. Achten Sie auf die Kommentare in der Funktion.

### 3.3 Messung der Latenz unter Last

Nun wird Quelle 2 (Timer 3) verwendet um eine zusätzliche Last zu erzeugen. Dazu erzeugt der Timer 3 ebenfalls einen Interrupt, der je nach Einstellung (`#define`) mit einer bestimmten Frequenz ausgelöst wird. Die dazugehörige ISR ist bereits implementiert. Darin wird lediglich ein Counter erhöht.

- a) Führen Sie nun die Messungen der Latenz mit verschiedenen Frequenzen durch (Siehe unten: Tabelle) und tragen Sie die Werte (Anzahl Ticks und Zeit) ein. Ermitteln Sie dazu den Takt des Timer 2 aus dem gegebenen Quelltext und berechnen Sie die Latenzzeit.

Interrupt-Frequenz des Timer 3	Minimum Latenz Timer 2	Maximum Latenz Timer 2	Durchschnitt Latenz Timer 2
Timer 3 nicht gestartet entspricht A 3.1b)	21	28	21
10 kHz	21	24	21
100 kHz	17	59	22
500 kHz	17	75	29
1 MHz	17	75	38

- b) Erklären Sie, warum die durchschnittliche Latenz des Interrupts für Timer 2 mit zunehmender Frequenz des Timer 3 Interrupts zunimmt.

Die überlappende Interrupts erhöht sich und der Timer 2 Interrupt zieht den Kürzeren

- c) Wie können die **beiden** Interrupts konfiguriert werden, um diesen Effekt auf die Latenz des Timer 2 zu vermeiden? Ändern sie die Konfiguration und wiederholen Sie die Messung (Eintragen von Anzahl Ticks und Zeit).

Interrupt-Frequenz des Timer 3	Minimum Latenz Timer 2	Maximum Latenz Timer 2	Durchschnitt Latenz Timer 2
1 MHz	17	27	20

**Hinweis:** Die Resultate in der Tabelle können variieren. Sie sind mehr als Anhaltspunkte zu verstehen

### 3.4 Bewertung

Die lauffähigen Programme müssen präsentiert werden. Die einzelnen Studierenden müssen die Lösungen und den Quellcode verstanden haben und erklären können.

Bewertungskriterien	Gewichtung
Die in Aufgabe 3.1 geforderte Funktionalität wurde umgesetzt.	2/4
Die Daten werden gemäss Aufgabe 3.2 auf dem Display ausgegeben	1/4
Die Fragen in Aufgabe 3.3 wurden korrekt und vollständig beantwortet	1/4