

Interrupts sind Programmunterbrechungen, die durch die Hardware ausgelöst werden. Fast könnte man sagen, sie sind der Natur abgeschaut. Ein Vergleich verdeutlicht das: Ein Erwachsener liest einen Zeitungsartikel. Das 'Programm' besteht also darin, dass er die Sätze liest und sich deren Inhalt merkt, so lange bis der Artikel zu Ende ist. Es wäre schlecht um ihn bestellt, wenn er während des Vorgangs nicht die Möglichkeit hätte, auf Interrupts zu reagieren. Wird er währenddessen von einer Biene gestochen, so unterbricht er das 'Hauptprogramm' Zeitungslesen und führt eine 'Interruptroutine' aus, indem er das Insekt vertreibt oder sich um den Stich kümmert. Anschließend liest er an der Stelle weiter, wo er vor dem Interrupt aufgehört hat. Das 'Programm' läuft also an alter Stelle weiter.

Und was passiert wenn mehrere Vorfälle zur gleichen Zeit erledigt werden soll? Also der Erwachsene liest gerade eine Zeitung ('Programm') und spürt den Stich der Biene. Währenddessen läutet auch noch die Türglocke, so merkt er sich diese neue Interruptanforderung und öffnet die Tür, sobald er sich hinreichend um den Insektenstich gekümmert hat. Wird er in der Hauptbeschäftigung zuerst durch die Türglocke und dann beim öffnen durch einen Insektenstich gestört, so ist der Insektenstich in der Lage, die erste Unterbrechung wiederum zu unterbrechen, da ein Schmerz allgemein Vorrang hat. Interruptanforderungen können also verschiedene Prioritäten haben. Schwieriger für eine Entscheidung ist das gleichzeitige Eintreffen von Interruptanforderungen gleicher Priorität: Beim Zeitungslesen läuten Türglocke und Telefon gleichzeitig. Wenn er weder einen Anruf noch einen Gast erwartet, wird die Entscheidung zufällig ausfallen.

Der Zufall ist der einzige Unterschied zwischen Natur und Maschine, ansonsten stellt man überraschende Parallelen fest. Ein Prozessor kann demnach nie genügend Interruptquellen besitzen, denn mit ihrer Hilfe kann er spontan auf äußere und innere Ereignisse reagieren und hat damit mit einem Lebewesen, das überlegt handelt, etwas gemein.


#### Aufgabe:

Ordnen Sie den verschiedenen Interrupts (Insektenstich, Türglocke) Prioritäten zu. (Prioritätsstufen 0 – 3; Priorität 3 ist die höchste)

In dem Ordner SW Interrupt ist ein Bsp. abgelegt. Um Ihnen die Funktionsweise des Interrupts zu verdeutlichen, wird bei dem Projekt in der main-Schleife Primzahlen ermittelt. Wenn bei der Berechnung ein Interrupt durch Pin 3.2 ausgelöst wird, unterbricht der µC die Berechnung und springt in die entsprechende ISR.

#### Aufgaben:

1. Lesen Sie sich das Infoblatt Interrupt durch und überspringen Sie dabei das Kapitel 7.4.
2. In dem Bsp. Interrupt werden welche Interrupts frei geschaltet? Sie können den Quellcode durchgehen oder das Bsp. simulieren. Lassen Sie sich dabei die Interrupts anzeigen.
3. Warum wird zum Setzen der Bits des SFRs IEN0 der „bitweise ODER“ – Operator benutzt?

 <b>MAX-TAUT-SCHULE</b> <b>OBERSTUFENZENTRUM</b> <b>VERSORGUNGSTECHNIK</b> <small>Fachoberschule - Berufsfachschule - Berufsschule - Technisches Gymnasium</small>	<b>µC Programmierung</b> <b>Interrupts</b>	Datum ..... Name ..... Klasse .....
---	---	---

4. Aktivieren Sie den externen Interrupt 1. Er soll flankengesteuert sein. Wenn das entsprechende Pin auf Low gesetzt wird, soll der Text „Interrupt ausgelöst“ seriell ausgegeben werden. Tipps: Bedenken Sie die Interrupt-Prioritäten. Das entsprechende Pin können Sie mittels dem Infoblatt und Datenblatt ermitteln.