Interrupts

Bei den Arduino Mikrocontrollern sind Interrupts im Grunde ein Signal, das es ermöglicht, eine Funktion in einem Sketch **jederzeit** direkt aufzurufen, gleichgültig, womit der Arduino sonst gerade „beschäftigt“ ist.

Wird ein Interrupt ausgelöst, wird der Ablauf des Programms angehalten, die Interrupt-Funktion ausgeführt [1)], wenn die ISR abgearbeitet ist, wird das das Programm an der Stelle der Unterbrechung fortgeführt.

Für ISR gelten einige besondere Regeln:

```
* ISR sollten möglichst kurz gehalten werden.
* ISR können keine Argumente bekommen oder Rückgabewerte zurückgeben.
Stattdessen werden - ausnahmsweise! - globale Variablen benutzt, um Daten
zwischen Interrupt Service Routinen und dem Hauptprogramm zu tauschen. Damit
die Variablen dabei korrekt geändert werden, sollten sie als ''volatile''
deklariert werden.
* Einige Funktionen verhalten sich in ISRs anders als gewohnt oder
funktionieren gar nicht:
  * ''millis()'' verlässt sich zum Zählen auf Interrupts, wird also in einer
Interrupt Service Routine niemals hochzählen.
  * ''delay()'' benutzt ebenfalls Interrupts und wird deshalb gar nicht in
einer Interrupt Service Routine funktionieren - sollte dort aber aucch n
iemals benutzt werden, weil, die ISR ja schnell abgearbeitet werden sollte!
  * ''micros()'' wird anfangs gut funktionieren, sich aber nach etwa 1 bis 2
ms unvorhersehbar verhalten - möglichst nicht benutzen.
  * ''delayMicroseconds()'' benutzt keine Zähler und wird deshalb normal
funktionieren.
```

1)
„ISR“ - Interrupt Service Routine

From:
https://wiki.qg-moessingen.de/ - **QG Wiki**

Permanent link:
https://wiki.qg-moessingen.de/faecher:nwt:arduino:lernbaustein2:interrupt:start?rev=1684774415

Last update: **22.05.2023 18:53**