

Microcontroller

Selbststudium Semesterwoche 10

Kontrollfragen

1. Wozu werden Interrupts benötigt? Beispiele? (Überall wo sofort und in genauer Echtzeit reagiert werden muss)
2. Was ist eine ISR?
3. Welche Arten von Interrupts gibt es beim MC9S08GB60?
4. Erklären Sie den Unterschied zwischen Polling und Interrupt
5. Was versteht man unter einem Interruptvektor?
6. Was ist eine Daisy-Chain?
7. Worin besteht der Unterschied zwischen einer Subroutine und einer ISR?
8. Wieso braucht man ein Timersystem? Würde es nicht genügen, Zeitverzögerungen mit NOP-Befehlen zu programmieren?
9. Lässt sich der Timer beim MC9S08GB60 anhalten? Lässt sich der Timer zurücksetzen? TPMxSC
10. Studieren Sie im Skript die Übersichten über das Timersystem (6-40 und 41). Schliessen Sie das Skript und notieren Sie alle möglichen Einstellungen in den zum Timersystem gehörenden Steuerregister.
11. Können abhängig von Timer-Ereignissen direkt Pins verändert werden? Wenn ja, nur wenn Interrupts auftreten?
12. Erklären Sie was beim Timersystem das Modulo Register und die Input Capture Funktion!

Antworten

1. Ein Interrupt verursacht eine kurzfristige Unterbrechung eines Programms, um eine andere, meist kurze, aber zeitkritische Verarbeitung durchzuführen.
 - Tastatureingabe
 - Daten über Netzwerk empfangen
 - Timer Event
2. ISR = Interrupt Service Routine. Ein Interrupthandler ist eine Funktion, welche ausgeführt wird, wenn das Hauptprogramm durch einen Interrupt unterbrochen wird.
3. Beim MC9S08GB60 gibt es folgende Arten von Interrupts:
 - 1 Real-Time Interrupt
 - 1 IIC-Interrupt
 - 1 A/D Interrupt
 - 1 Keyboard Interrupt
 - 7 Interrupts für serielle Kommunikation
 - 10 Interrupts für Timer System
 - 1 Internal Clock
 - 1 Low Voltage Detection
 - 1 Interrupt für ext. IRQ Pin
 - 1 Software Interrupt Pin
 - 1 Reset Interrupt
4. Polling bezeichnet in der Informatik die Methode, den Status eines Geräts aus Hard- oder Software mittels zyklischem Abfragen zu ermitteln. Im Unterschied zum Interrupt braucht der Polling Approach

kontinuierlich Systemressourcen. Auf ein Interrupt wird genau dann reagiert, wenn dieser ausgelöst wird.

5. Der Interruptvektor ist ein Speicherplatz, der die Startadresse der zugehörigen ISR enthält (Pointer auf Startadresse).
6. Eine Daisy Chain (englisch, wörtlich „Margeritenkette“) kaskadiert externe Geräte mittels einer Interrupt-Enable Leitung (IE_I, IE_O) so, dass man hardwaremässig gerade eine Prioritätsregelung erhält. (Siehe Script „Microcontroller“, 6-31). Ein höher priorisiertes Busgerät kann beim Auslösen eines Interrupts verhindern, dass während dem Ausführen der ISR die niedriger priorisierten Busgeräte Interrupts auslösen können.
7. Folgende Unterschiede existieren zwischen ISR und Sub Routine:
 - Eine Subroutine läuft, wenn man sie aufruft – eine ISR, wenn ein Interrupt ausgelöst wird.
 - Das Retten des CPU Status' entfällt beim ISR.
 - Beim ISR wird mit „RTI“ zurückgesprungen während bei einer Subroutine mit „RTS“ zurückgesprungen wird.
8. Timer Systeme werden benötigt, jegliche Arten von Zeitmessaufgaben zu lösen (Zeitmessung, Wecker).

Der Nachteil an anderen „Verzögerungsmechanismen“ wie NOP oder FOR-Schleifen besteht darin, dass man nicht genau weiss, wie lange die CPU für die Verarbeitung dieses Leerbefehls benötigt. Eine genaue Messung ist nur mit dem Timer System möglich.
9. Ein Timer System (= das ganze System mit allen Channels) kann angehalten werden, in dem die Clock Source CLKSA : CLKSB auf 0 : 0 gesetzt wird. Ein Reset des Timers kann mit TPM1CNT = 0 ausgelöst werden. Der Timer beginnt nun wieder bei 0.