

Übung 12

Tasterinterrupt

Ziele dieser Übung:

Sie sollen in der Lage sein,

- 1) Auf dem Leguan-Board einen Tasterinterrupt zu programmieren
- 2) Die Register von GPIO und EXTI-Controller sowie des NVIC für die Interruptverarbeitung in der Programmiersprache C korrekt zu initialisieren.

Verwenden Sie für den Test Ihres Codes den Debugger und das Leguan-Board.

Aufgabe 12.1 Taster-Interrupt Leguan-Board

Die Betätigung des Tasters ganz rechts auf dem Leguan-Board soll einen Interrupt auslösen.

Der Taster ist mit GPIO Port D pin 11 des Microcontrollers verknüpft. Damit der Taster einen Interrupt auslösen kann müssen diverse Register initialisiert werden:

- Clock für folgende Peripherie freischalten: GPIOD und SYSCFG. Dies wird schon in der Funktion LEGUAN_Init() gemacht.
- GPIOD pin 11 als Input definieren (Register GPIOD_MODER), ohne Pull-up und Pull-down (Register GPIOD_PUPDR).
- Line Mapping für EXTI definieren (Register SYSCFG_EXTICR3), im Code SYSCFG->EXTICR[2].
- Interrupt für Line_11 im EXTI aktivieren (Register EXTI_IMR1)
- Flanken im EXTI-Controller für Line_11 definieren (Register EXTI_RTSTR1 und EXTI_FTSR1)
- Interrupt im NVIC für EXTI15_10 freischalten, HAL-Funktion HAL_NVIC_EnableIRQ();
- Globale Interrupts freischalten (__enable_irq())

Übersicht der Register:

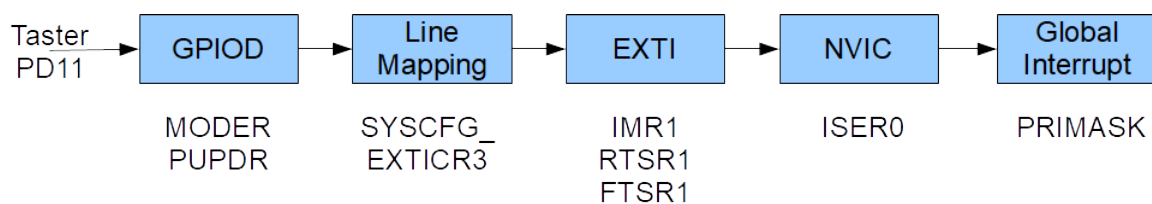


Abbildung 1: Verwendete Register für den Tasterinterrupt

Eine detaillierte Beschreibung der einzelnen Register finden Sie in folgenden Dokumenten:

- RM0433 Reference Manual STM32H7xx von ST Microelectronics
- ARM[™] v7-M Architecture Reference Manual von ARM

Gehen Sie wie folgt vor:

- 1) Die Musterlösung zu dieser Aufgabe finden Sie auf Moodle. Importieren Sie dieses Projekt in die Entwicklungsumgebung STM32CubeIDE.
- 2) Prüfen Sie den Code für die Initialisierung der Hardware-Register (GPIO, EXTI, NVIC).
- 3) Prüfen Sie den Code des Handlers EXTI15_10_IRQHandler in derselben Datei. Dieser prüft, ob der IRQ durch den Taster (EXTI Line_11) ausgelöst wurde und inkrementiert eine counter-Variable. Der Wert der Variablen wird in der Endlosschleife in main() auf dem 7-Segment Display ausgegeben.
- 4) Testen Sie das Programm mit dem Debugger, indem Sie die Initialisierung der HW-Register verfolgen. Prüfen Sie dazu die Werte im Tab "SFR" im Debugger. Setzen Sie anschliessend einen Breakpoint in Handler EXTI15_10_IRQHandler. Wenn Sie nun den Taster drücken, wird das Programm beim Breakpoint angehalten.

Tipps für die Programmierung von anderen Interrupt-Quellen:

Falls Ihr Programm nicht auf ein Ereignis reagiert, prüfen Sie bitte folgende Punkte:

- Ist der Name des modifizierten IRQHandlers korrekt? Dies ist der Name, welcher den Default-Eintrag in der Vektortabelle überschreibt.
- Setzen Sie einen Breakpoint nach der Initialisierungsroutine und prüfen Sie, ob Sie dort korrekt halten.
- Prüfen Sie, ob Sie alle Register gemäss Aufgabenstellung initialisiert haben. Die Werte können Sie im Debugger im Fenster "SFRs" (Tab neben dem Fenster "Registers") lesen.
- Setzen Sie einen Breakpoint in der ersten Instruktion des IRQHandlers. Prüfen Sie, ob beim Eintreffen des Ereignisses (z.B. Tastendruck) der Breakpoint auslöst.

Angezeigter Wert auf dem 7-Segment Display:

Wenn Sie das Programm ohne Breakpoint laufen lassen und den Taster betätigen, kann es sein, dass der angezeigte Werte nicht um 1 inkrementiert wird, sondern um eine beliebige Zahl. Was könnte die Ursache sein, wie könnten Sie dieses Problem beheben?