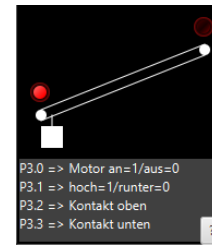


Ein paar Aufgaben zu Timer / Counter und externen Interrupts.

Aufgabe 1

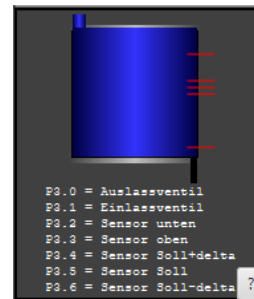
Legt an Port 3 das Seilbahnmodell an und stellt die Zeit pro Maschinen-Zyklus auf 10ms/MZ.



- Lasst die Seilbahn nach oben und unten fahren.
Verwendet zum Richtungswechsel die externen Interrupts 0 und 1.
D.h. im Hauptprogramm aktiviert ihr beide Interrupts flankengesteuert und lasst die Bahn nach oben fahren. In den beiden ISR dreht ihr P3.1 um dass die Richtung sich ändert.
ACHTUNG: P3.1 ist hoch=0 und runter=1 (im Simulator steht es falschrum)
- Nehmt Timer 0 als 8 Bit Timer hinzu, so dass die Seilbahn an den Haltestellen 1 Sekunde wartet. Dazu konfiguriert ihr den Timer in der Main mit passendem Reload-Wert, aktiviert den Interrupt für den Timer - startet ihn aber nicht.
In den ISR der externen Interrupts startet ihr jetzt den Timer und macht sonst nicht. Wenn der Timer überläuft, dann ändert ihr die Richtung der Bahn und stoppt den Timer.

Aufgabe 2

Legt jetzt das Modell „Behälter“ an Port 3 an.
Der Behälter soll vor Auslieferung gespült werden.
Hierzu wird der Zufluss geöffnet und der Abfluss gesperrt.
Löst der obere Sensor einen externen Interrupt 1 aus (P3.3),
so wird der Zufluss geschlossen und Timer 0 gestartet,
so dass er 4 Sekunden wartet.
Sind die 4 Sekunden um so wird der Behälter entleert.
Dieser Vorgang soll 3 Mal wiederholt werden, danach ist der Zulauf zu und der Ablauf bleibt auf.



- Main: Aktivieren Sie den externen Interrupt 1 (flankengesteuert)
Schließen Sie den Ablauf (P3.0=1), öffnen sie den Zulauf (P3.1=0) und lassen sie das Bit 3.3 auf 1 (für den Input der Flankenerkennung).
In der ISR von X1 schließen Sie den Zulauf.
- Ergänzen Sie in der Main die Konfiguration für einen Timer 0, so dass er alle 200MZ überläuft (das sind 2 s bei 10ms/MZ). Aktivieren Sie den Timer 0 Interrupt.
Gestartet wird der Timer in der ISR des externen Interrupts 1 nach Ventilschließung.
Die ISR des Timers zählt die Timer-Überläufe mit (z.B. in R0) und nach 2 Überläufen wird der Ablauf geöffnet, der Timer gestoppt und R0 auf 0 gesetzt.
- Um das jetzt 3 Mal in Folge durchzuführen gibt es unterschiedliche Möglichkeiten. Der externe Interrupt 0 kann nicht genutzt werden, da beim Leeren hier eine steigende Flanke entsteht (d.h. pegelgesteuert würde gehen).
Sie können aber in der ISR des Timers warten bis der Behälter leer ist (mit einer Schleife).
Wenn der Behälter leer ist: Schließen Sie den Ablauf, öffnen Sie den Zulauf.
Zählen Sie dort in R1 mit wie oft das schon passiert ist, beim 3. Mal öffnen Sie den Zulauf nicht und alles ist gut.

Lösungen:
Aufgabe 1

```
a)
include reg8051.inc
    jmp main
org 03h      ; ISR für Anschlag oben
    setb P3.1 ; Bahn fährt runter
    reti
org 13h      ; ISR für Anschlag unten
    clr P3.1  ; Bahn fährt hoch
    reti

main:
    setb IT0   ; Externer Interrupt 0
    setb EX0   ; flankengesteuert und Interrupt
    setb IT1   ; dito für externen Interrupt 1
    setb EX1
    setb EA    ; Interrupts an

    ; Bahn fährt anfangs hoch
    setb P3.0 ; Motor an
    clr P3.1  ; Richtung oben
    jmp $

end

b) (Änderungen zu a sind fett)
include reg8051.inc
    jmp main
org 03h
    setb TR0    ; Timer 0 starten
    reti
org 0Bh
    cpl P3.1
    clr TR0    ; Timer 0 anhalten
    reti
org 13h
    setb TR0    ; Timer 0 starten
    reti

main:
    mov TMod, #2      ; 2 = 0000 0010b
    ; 256-100=156 = Reloadwert für 100MZ
    mov TH0, #156    ; Reloadwert
    mov TL0, #156    ; Startwert für 1. Durchlauf

    setb ET0        ; Timer 0 Int zulassen

    setb IT0
    setb EX0
    setb IT1
    setb EX1
    setb EA

    setb P3.0
    clr P3.1
    jmp $

end
```

Aufgabe 2 Lösung

```
a) include reg8051.inc
   ablauf BIT P3.0
   zulauf Bit P3.1
       jmp main

   org 13h
       setb zulauf
       reti

main:
   ; externen Interrupt 1 aktivieren:
   setb IT1
   setb EX1
   setb EA
   ; Zulauf aufmachen
   clr zulauf
   setb ablauf ; ist eigentlich schon 1
   ; mit mov P3, #1111101 geht es auch

       jmp $

end

b) include reg8051.inc
   ablauf BIT P3.0
   zulauf Bit P3.1
       jmp main
org 0Bh
       call ISR_Timer ; sonst Code-Overlap
       reti

   org 13h
       setb zulauf
       setb TR0
       reti

ISR_Timer:
       inc R0
       cjne R0, #2, ende
       clr ablauf
       mov R0, #0
       clr TR0
ende: ret

main:
   ; Falls der Behälter
   ; zu Beginn voll ist
   clr ablauf
   jnb P3.2, $ ; warten bis er leer ist
   mov R0, #0
   ; Timer Konfiguration
   mov TMod, #2
   mov TH0, #56 ; 256-200
   mov TL0, #56 ; Startwert
   setb ET0
   ; externen Interrupt 1 aktivieren:
   setb IT1
   setb EX1
   setb EA
   ; Zulauf aufmachen
```

Informationstechnik

Coronatäre Woche 2

```
        clr zulauf
        setb ablauf ; ist eigentlich schon 1
        ; mit mov P3, #1111101 geht es auch
        jmp $

end

c) include reg8051.inc
   ablauf BIT P3.0
   zulauf Bit P3.1
       jmp main

org 0Bh
   call ISR_Timer ; sonst Code-Overlap
   reti

org 13h
   setb zulauf
   setb TR0
   reti

ISR_Timer:
   inc R0
   cjne R0, #2, ende
   clr ablauf
   mov R0, #0
   clr TR0

   inc R1 ; Anzahl der Reinigungen
   cjne R1, #3, nochmal
   ret

nochmal:
   jnb P3.2, $ ; warten bis Behälter leer
   setb ablauf
   clr zulauf
ende: ret

main:

   ; Falls der Behälter
   ; zu Beginn voll ist
   clr ablauf
   jnb P3.2, $ ; warten bis er leer ist
   mov R0, #0
   ; Timer Konfiguration
   mov TMod, #2
   mov TH0, #56 ; 256-200
   mov TL0, #56 ; Startwert
   setb ET0
   ; externen Interrupt 1 aktivieren:
   setb IT1
   setb EX1
   setb EA

   ; Zulauf aufmachen
   clr zulauf
   setb ablauf ; ist eigentlich schon 1
   ; mit mov P3, #1111101 geht es auch

   jmp $

end
```