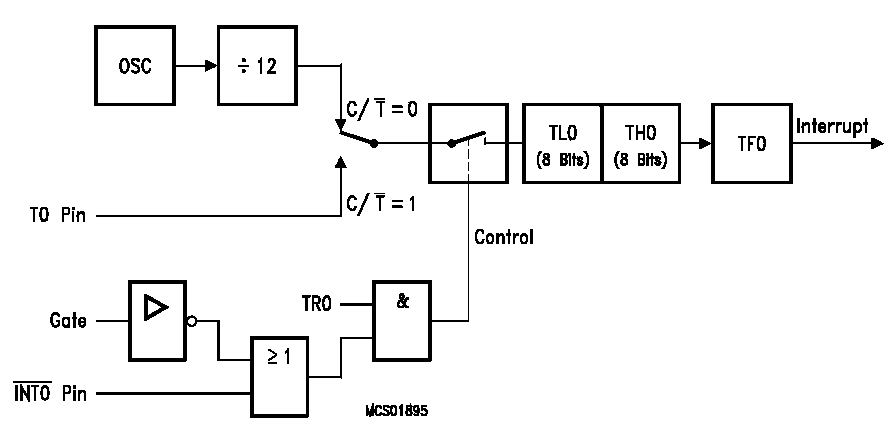
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Gate | C/T | M1 | M0 | Gate | C/T | M1 | M0 |
| Kontrolle für Timer 1 | | | | | Kontrolle für Timer 0 | | | |

# Zähler mit Counter

# Funktion:

****Auf einem Fließband werden mittels einer Lichtschranke Teile gezählt. Die Lichtschranke ist mit dem T0-Pin verbunden Das Band wird mit dem Eintaster eingeschaltet und nach 10 Teilen automatisch gestoppt. Danach werden keine Impulse von der Lichtschranke mehr gezählt.   
Wird der Eintaster erneut betätigt läuft das Band wieder los.

Zuordnungen:

Lichtschranke T0-Pin = P3.4

Starttaster = P1.0

Bandmotor = P1.2

# Aufgaben

1. Entwerfen Sie den PAP und
2. Das Assemblerprogramm.

TMOD-Register

# Stoppuhr mit Timer

# Funktion:

Eine Stoppuhr mit 1ms zeitlicher Auflösung soll programmiert werden. Zur Bedienung sind drei Taster für Start, Stopp und Rücksetzten vorgesehen.

Als Zeitbasis dient der Timer0 im Pollingbetrieb, d.h. das Timerflag wird ständig abgefragt. Die Anzeige, die alle 10ms aktualisiert werden soll, ist lediglich als Unterprogramm zu berücksichtigen. Die Grundstruktur des Programms entspricht einer vielfach verschachtelten Zählschleife, wobei jede Schleife 10mal durchlaufen wird. Die gemessene Zeit lässt sich dann folgendermaßen darstellen:  
  
R3\*10s+R4\*1s+R5\*100ms+R6\*10ms+R7\*1ms  
  
Jeder Zähler hat deshalb maximal den Wert 10.

Der Timer0 soll im Autoreload-Betrieb arbeiten.

# Aufgaben

1. Entwerfen Sie den PAP und
2. Das Assemblerprogramm.

Lösung zu Zähler mit Counter

Ein equ P1.0

Band equ P1.2

Init:

clr Band

mov tmod,#00000110b ;counter0 autoreload,

mov th0,#246 ;Zähler bis 10

mov tl0,#246

Start:

jb Ein, start ;Starttaste betätigt?

setb tr0 ;Zähler start

setb Band ;Band ein

jnb tf0,$ ;warte bis 10. Impuls

clr tf0 ;Flag löschen

clr Band ;Band stopp

clr tr0 ;Zähler inaktiv

jmp start

end

Lösung zu Stoppuhr mit Timer

Hinweis: Das UP Anzeige war nicht verlangt

offset equ 000h

ICM equ p1

Digit0 equ ICM.4

Digit1 equ ICM.5

Digit2 equ ICM.6

Digit3 equ ICM.7

Start equ P2.0

Stopp equ P2.1

Reset equ P2.2

DSEG at 30h ;Byte-Variable definieren mit Adresse

hunderdstel: DS 1 ;Byte Variable mit Angabe der Größe z.B. 2 Byte

Sekunde: DS 1

cseg

ORG offset;+100h;Startadresse für das Programm

jmp init

Init:

mov tmod,#00000010b ;timer0 autoreload,

mov th0,#156 ;100us Zeitbasis

mov tl0,#156

mov r2,#0

mov r3,#0

mov r4,#0

mov r5,#0

mov r6,#0

mov r7,#0



Ziel0:

call anzeige

jb start, Ziel1 ;Starttaste betätigt?

setb tr0

Ziel1:

jb Reset,Ziel2 ;Resettaste betätigt?

mov r2,#0

mov r3,#0

mov r4,#0

mov r5,#0

mov r6,#0

mov r7,#0

Ziel2:

jb stopp,Ziel3 ;Stopptaste betätigt?

clr tr0

jmp Ziel0

Ziel3:

jnb tf0,ziel0

clr tf0

inc r7 ;alle 100us

cjne r7,#10, ziel0

mov r7,#0

inc r6 ;alle ms

cjne r6,#10, ziel0

mov r6,#0

inc r5 ;alle 10ms

cjne r5,#10, ziel0

mov r5,#0

call anzeige

inc r4 ;alle 100ms

cjne r4,#10, ziel0

mov r4,#0

inc r3 ;jede s

cjne r3,#10, ziel0

mov r3,#0

inc r2 ;alle 10s

cjne r2,#10, ziel0

mov r2,#0

jmp ziel0

Anzeige:

mov ICM,#0f0h ;erste beide Stellen auf 0 setzen

mov a,r4 ;Einer ausgeben

orl a,#11110000b ;Zahlenwert bei ausgeschalteter Anzeige ausgeben

mov ICM,a

clr Digit1

setb Digit1

mov a,r5

orl a,#11110000b ;Zahlenwert bei ausgeschalteter Anzeige ausgeben

mov ICM,a

clr Digit0

setb Digit0

mov a,r2 ;Einer ausgeben

orl a,#11110000b ;Zahlenwert bei ausgeschalteter Anzeige ausgeben

mov ICM,a

clr Digit3

setb Digit3

mov a,r3

orl a,#11110000b ;Zahlenwert bei ausgeschalteter Anzeige ausgeben

mov ICM,a

clr Digit2

setb Digit2

ret

end

zu 1

t=s/v: 0,1m/2,4m/s=41,7ms 0,1m/1,6m/s=62,5ms

41700=0xA2E4 62500=0xF424