

Timer KA

Timer

Timerbit

Man kann den Timer zum eine Bitweise mit **TF0** & **TF1** konfigurieren.

Timerinterrupt

Man kann den Timer auch per Interrupt Steuern(Wenn Timer abgelaufen ist wird ISR ausgeführt)

- ISR Aktivieren

Timermodi

16 bit (natürlich ohne autoreload)

- Zählerstand ist auf 2 Register aufgeteilt.

8-Bit autoreload

gezählt wird im Register TL0 bzw. TL1(der Reload erfolgt bei Überlauf aus TH0 bzw TH1.)

Weitere

- 2*8bit (Wird nicht angefragt)
- antiquierter 13 bit-timer

2 Betriebsarten

Counter: Spezielles bit wird gezählt(Port 3.4 oder 3.5)

- Man kann den Timer als Counter verwendet in dem man ein Timer mit reload verwendet & beim Durchzählen immer ein register erhöht.

Timer: Takt wird gezählt

- Man kann den den Timer als Timer nehmen, so dass dieser nach einer bestimmten Zeit ausgelöst wird.

-> Unterschied ist Taktmodus

Unterschied Timer <-> Counter

- Counter zählt Ereignisse -> Timer ist ein Counter der auf ein zeitliches Ereignis reagiert.

<https://www.mikrocontroller.net/topic/264150>

Interrupt

Einsprungadressen

Die Einsprungsadressen für den Interrupt sind:

- ISR 0: 0x03
- ISR 1: 0x13

Platzbedarf von ISR

- Ein ISR hat nicht so viel Platz, deshalb ist es ratsam UP bzw weniger code in der ISR haben.

IP-Register(Reihenfolge Interrupts -> Formelsammlung)

IP Interrupt-Prioritäten-Register, bitadressierbar, 0B8h								
Bit	7	6	5	4	3	2	1	0
	-	-	PT2	PS	PT1	PX1	PT0	PX0
Bit-Adr.	0BFh	0BEh	0BDh	0BCCh	0BBh	0BAh	0B9h	0B8h
Priorität von			Timer2	Serial Port	Timer1	Ext.Interrupt1	Timer0	Ext.Interrupt0

0: niedrige Priorität 1: höhere Priorität

Interrupts können nur von anderen Interrupts mit höherer Ebene unterbrochen werden. Treten 2 Interrupts gleicher Priorität gleichzeitig auf, so werden sie in folgender Reihenfolge bearbeitet:

ExtInt0 → Timer0 → ExtInt1 → Timer1

Interrupt	Einsprungsadresse	Interrupt-Anforderungs-Bit
Externer Interrupt 0	0003h	IE0
Timer0 - Überlauf	000Bh	TF0
Externer Interrupt 1	0013h	IE1
Timer1 – Überlauf	001Bh	TF1
Serieller Schnittstellen-Interrupt	0023h	RI oder TI

Polling oder Interrupt

- Man unterscheidet zwischen 2 Betriebsarten bei Timer/Interrupts

1. Polling -> Man fragt die ganze Zeit ab
 - Bsp USB = Nicht echtzeitfähig
2. Interrupt -> Man hört/abwartet bis irgendwas passiert

Unterschiedliche Zustände -> Unterschiedlichen Zeiten

- Beispiel Ampel (Verschiedene Ampelphasen)
 - Register
 - Tabelle
 - Register/Tabelle im ROM -> Unterschied Tabelle im ROM & Register benutzen

Aufgabenstellungen

- PAP
- Code
- Initialisierung
- Timer berechnen
- erklären

- Timerschachtelung -> Bsp Pause Timer schachteln(Was tun wenn Timerdurchlauf zu kurz ist)
- Timerstartwertberechnung