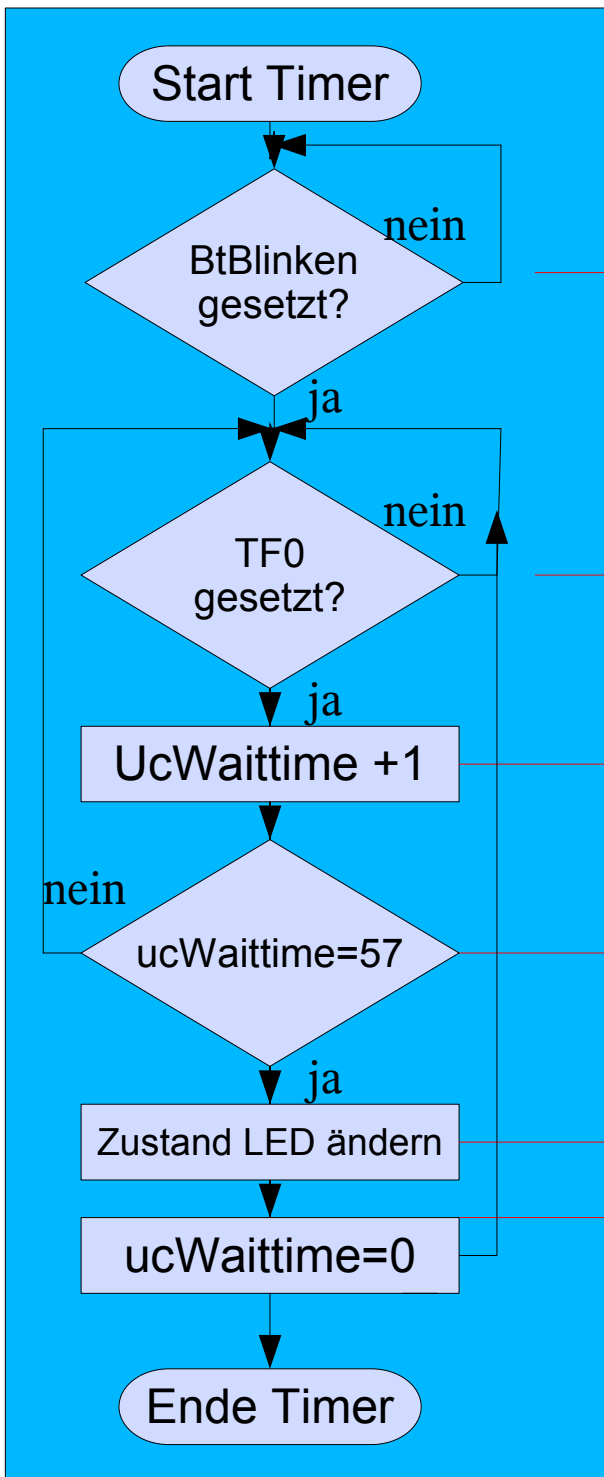


- > Die Verwendeten Ports $P1^3 = \text{Taste}$ und $P2^0 = \text{LED}$ werden Initialisiert
- > Zählvariable wird definiert
- > Zustände EIN und AUS werden definiert
- > Das Speichern des btBlinken Flags wird Initialisiert
- > Der Timer Null wird mit Reloadwert geladen
- > Timer 0 starten

- Initialisierung der ISR
- Freigabe des Interrupts 0
- Allgemeine Interruptsperre aufheben
- Taste löst an Port $P1^3$ den externen Interrupt 0 aus

Unterprogramm Timer



>Das btBlinken Flag wird abgefragt, ist dieses gesetzt, dann wird die Abfrageschleife TF0 ausgeführt.
>Flag nicht gesetzt = Es wird Keine Operation ausgeführt

> In dieser Abfrage wird das Überlaufflag des Timers 0 abgefragt. Da der Timer aus einem Hexadezimalzähler besteht und nur 8Bit für den Zählvorgang zur Verfügung stehen würde er bei der Zahl 0xFFFF überlaufen, dies bedeutet, das er wieder bei 0x0000 das Zählen beginnt. Bei dem Wechsel wird ein Flag gesetzt : das TF0.
> Ist dieses Flag gesetzt springt das Programm in die Zählschleife.

>Im ersten Teil der Zählschleife wird die Zählvariable ucWaitime bei jedem gesetzten TF0- Flag um eins erhöht.
>Der nun verarbeitete Zustand des Flags von Timer Null wird nun wieder gelöscht.

Da der Timer zwischen zwei Überläufen ca. 17Ms benötigt, wir aber eine Zeit von 1s zwischen den Zuständen AN und AUS der LED benötigen wird hochgezählt, bis das TF0- Flag 57x gesetzt wurde. $57 * 17ms = 1s$. Ist dies der Fall wird der Zustand der LED umgeschaltet.

Die Variable ucWaitime wird nun wieder für den nächsten Durchgang zurückgesetzt.

