



MIKROCONTROLLERTECHNIK

## Labor 8

Vorbereitungsaufgaben

*Autor:* Richard GRÜNERT

28.4.2020

# 1 Prinzip der Frequenzmessung

Durch Festlegen einer *Torzeit* (Gate Time)  $T_{\text{Gate}}$  und Zählen der Flanken des Eingangssignals ( $Z$ ) während dieser Zeit, kann die Frequenz des Signals berechnet werden.

$$f = \frac{Z}{T_{\text{Gate}}}$$

## 2 Umsetzung

Das zu messende Eingangssignal liegt an P2.2 an, welcher zur Capture-Einheit 0 des Timers A gehört. Dort muss dann im TACCTL-Register über das CCIS-Flag der Eingang CCI0B (01b) ausgewählt werden. Die Torzeit wird dann mit dem Timer B generiert. Die Berechnung wird im Interrupt (TAIFG) des Timer B durchgeführt.

## 3 Erzeugung der Torzeit

Ist der Timer im continuous-Mode, lässt sich bei einer Taktfrequenz von 16 MHz und dem maximalen Teilerwert von 8 eine maximale Periodendauer von

$$T_{\text{max}} = \frac{2^{16} - 1}{16/8 \text{ MHz}} = 32.7675 \text{ ms}$$

einstellen. Um eine Torzeit von 1 s zu erreichen, muss man nun die Anzahl der Timerüberläufe (Interrupt TAIFG) zählen, bis diese den nötigen Wert  $k$  erreicht haben. Dieser ist für  $T_{\text{max}}$

$$k = \frac{1 \text{ s}}{T_{\text{max}}} = 30.518$$

Da  $k$  in diesem Fall kein ganzzahliger Wert ist, man aber nur ganze Überläufe zählen kann, entstehen Ungenauigkeiten. Um das zu verhindern, kann man den Timer auch in den up-Mode setzen und zu einem

Wert zählen lassen, der bei der gewünschten Zeit eine ganzzahlige Anzahl an Überläufen/Interrupts erzeugt, beispielsweise 40 Überläufe, woraus sich eine Periodendauer von

$$T_{up} = \frac{1 \text{ s}}{40} = 25 \text{ ms}$$

ergibt.

Um diese Zeit zu erzeugen, braucht der Timer im up-Mode den Comparewert

$$\text{CCR0} = 25 \text{ ms} \cdot 2 \text{ MHz} = 50000$$

## 4 Genauigkeit

Die sich ergebende Messabweichung ist

$$F = \pm \frac{1}{2} \cdot \frac{1}{T_{\text{Gate}}}$$

## 5 Portinitialisierung

P2.2 muss für die Input-Richtung konfiguriert werden (0).

```
P2DIR  &= ~(1<<2);
```

Außerdem muss die Zweitfunktion ausgewählt werden (1).

```
P2SEL  |= (1<<2);
```