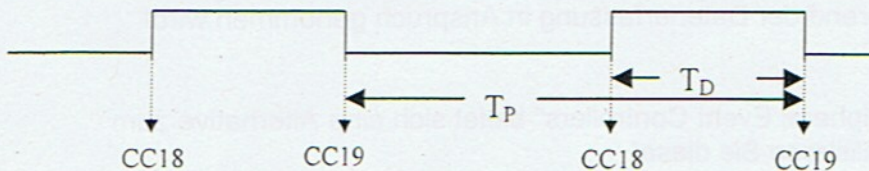


Versuch 3: PWM-Eingangssignal

Erfassen und Berechnen der Temperatur:

Das Ausgangssignal des Temperatursensors ist ein pulswidenmoduliertes Signal und liegt an den Eingängen P8.2 (CC18) und P8.3 (CC19) des C164 an:



T_D ist die Pulsdauer

T_P die Periodendauer (sie beträgt ungefähr 200 μsec)

Es gilt folgender Zusammenhang für die Temperatur t in $^{\circ}\text{C}$:

$$T_D / T_P = 0.32 + 0.0047 \cdot t$$

Die Ermittlung der interessierenden Zeiten kann wie folgt durchgeführt werden:

- Die Zeitbasis liefert T8 (Capcom2), der freilaufend (Reload = 0) ständig mit $f_{\text{CPU}}/8$ getaktet wird.
- Die ansteigende Flanke führt zur Übernahme des aktuellen Zählerstandes in das Compare-Register CC18.
- Die fallende Flanke übernimmt nach CC19.
- Die Differenz von CC19-CC18 ist die aktive Impulsdauer T_D .
- Die Differenz zwischen dem gespeicherten alten CC19-Wert (*static!!*) und dem aktuellen CC19-Inhalt ergibt die Periodendauer T_P .
- Der Zeitpunkt für die Bestimmung dieser Werte ist die abfallende Signalflanke, d.h. das Capture-Ereignis in CC19 löst einen Interrupt aus. In dessen ISR werden T_D und T_P ermittelt und in getrennte Puffer (Arrays!) übernommen.
- Wählen Sie 50 als Größe für die nötigen Puffer.
- Sind alle 50 Werte erfasst, so sollen weiter eintreffende Signalflanken ignoriert werden (keine weiteren Interrupts!).

Beachten Sie:

Da zum Zeitpunkt des Auftretens der ersten negativen Signalflanke – also beim ersten Interrupt – u.U. noch keine gültige positive Flanke und deshalb auch noch kein gültiger CC18-Wert vorhanden ist, muss der erste von den Folgeinterrupts unterschieden werden. Das gleiche gilt für den nötigen vorhergehenden CC19-Wert, auch dieser liegt noch nicht vor – Synchronisationsproblematik!!

Mit Drücken der Taste 3 ist die Temperaturmessung zu starten – "*StartTemp()*". Sind wie oben beschrieben alle nötigen 50 Messwerte erfasst, so liefert der Aufruf der Funktion "*bTempDa()*" den Wert "TRUE" zurück. Der dann folgende Aufruf der Funktion "*fGetTemp()*" realisiert die Berechnung der Temperatur aus dem Mittelwert der 50 in Folge durchgeführten Einzelmessungen. Das Resultat ist in der 3. Zeile des LC-Displays anzuzeigen.

Die Funktionalität aus dem zweiten Versuch sollte unverändert bleiben!

- a) Realisieren Sie die Temperaturmessung und testen Sie diese mit geeigneten Simulationsfunktionen, die aus der Toolbox zu starten sind, für -10°C , 20°C und 60°C . Sorgen Sie dafür, dass diese Simulationsfunktionen direkt beim Eintritt in den Debug-Modus zur Verfügung stehen (in Target Options/Debug DBGSTART.INI als "Initialization File" angeben!). Setzen Sie geeignete Unterbrechungen, z.B. in der ISR des CC19 und überprüfen Sie die

Ergebnisse auf Plausibilität!

- b) Testen Sie nun Ihr erzeugtes Projekt im Target!

Die Frequenz des PWM-Signals beträgt etwa 5 kHz, dies entspricht der Interruptfrequenz während der Messdatenerfassung.

- c) Ermitteln Sie in etwa, welcher Teil der verfügbaren Prozessorzeit durch die obige Methode der Temperaturmessung während der Datenerfassung in Anspruch genommen wird!
- d) Durch Einbeziehung des "Peripheral Event Controllers" bietet sich eine Alternative zum obigen Messverfahren an, realisieren Sie diese!