



PRAKTIKUM EINGEBETTETE SYSTEME

SS2026

Termin 3

C-Programmierung für eingebettete Systeme, Timer/Counter-
Baustein, Wave/Capture-Modus

Name, Vorname	Matrikelnummer	Anmerkungen
Datum	Raster (z.B. Mi3x)	Testat/Datum

Legende: V:Vorbereitung, D: Durchführung, P: Protokoll/Dokumentation, T: Testat

Lernziele:

Im diesem Termin beschäftigen wir uns mit der Distanzmessung mittels eines Ultraschallsensors HC-SR04. Das Trigger-Signal ist an TIOB7/PIOC29 und das ECHO-Signal ist an TIOA8/PIOD7 angeschlossen. Erfahrungen mit Timer-/Zählerperipherie zum Erzeugen von Signalen und zum Messen von Zeiten sollen gemacht werden.

Aufgabe 1

Initialisieren Sie in `sonic_init()` die verwendeten Timer-/Zählerbausteine so, dass an TIOB7 das benötigte Trigger-Signal generiert werden kann und über TIOA8 die Länge des ECHO-Signal erfasst werden kann.

Aufgabe 2

Implementieren Sie das Auslesen der zeitlichen Länge des ECHO-Signal innerhalb der Funktion `sonic_loop`. Verwenden Sie dafür die Register RA und RB, sowie das Statusregister (SR). Eine Messung kann über das CCR gestartet werden. (Verwenden Sie Busy-Waiting oder Polling.)

Aufgabe 3

Implementieren Sie nun die gleiche Messung mithilfe von Interrupts. Implementieren Sie dazu die Prozedur `TC8_Handler` zur Behandlung des Interrupts. Den Interrupt können Sie mit „Interrupt Set-enable Registers“ (ISER) aktivieren.

Aufgabe 4

Zeigen sie die ermittelte Distanz zum Hindernis auf der LED Matrix an, wenn eine Abstandsmessung vorliegt und der Status `state.sonic` (SONIC_ON) gewählt ist.

Leitfragen

- Welche Distanzen können gemessen werden?
- Welche Zeiten benötigen Abstandsmessungen?
- Welches Verfahren (Busy-Waiting, Polling oder Interrupt) werden Sie für die Projektlösung verwenden?
- ..