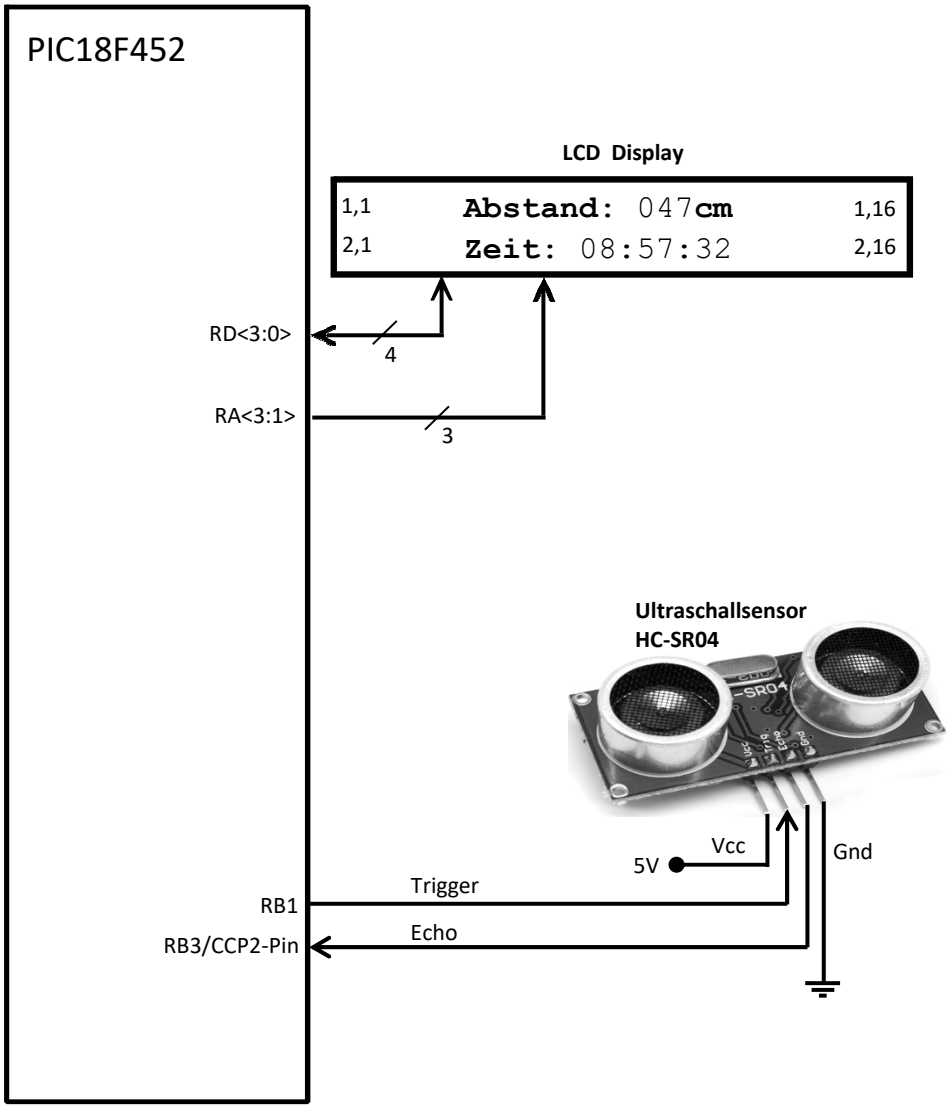
Hochschule Mannheim Laborversuch PIC 3 Interrupt, CCP / Ultraschall Version 06.2018		Institut für Embedded Systems Prof. Dr. Kurt Ackermann
Name, Vorname: WS/SS___Gruppennummer:_____	Matr. Nr.:	Testat:

Die folgende Anordnung des Mikrocontrollers PIC 18F452 ist gegeben:



Aufgabe:

Entwickeln und testen Sie ein aus zwei Teilaufgaben bestehendes Programm in C:

Task 1: Entfernungsmessung mittels eines Ultraschallsensors (hohe Priorität)

Der Ultraschallsensor HC-SR04 wird über zwei Pins angesteuert. Über RB1 soll zyklisch alle 100ms ein Triggerimpuls an den Sensor übertragen werden. Die fallende Flanke dieses Pulses, der mindestens 10µs lang sein muss, veranlasst den Sensor ein 40kHz Burst-Signal auszugeben. Der Sensor misst nun über seinen Empfänger die Laufzeit des an einem Objekt reflektierten Signals. Als Messergebnis wird am Echo Pin des Sensors ein Puls ausgegeben, dessen Breite proportional zur Entfernung des Objekts ist. Der vom PIC an RB3 empfangene Puls soll ausgewertet werden, indem die Dauer von der steigenden bis zur fallenden Flanke gemessen wird. Der Zusammenhang zwischen der Pulsbreite und der Entfernung ist:

Abstand [cm] = Pulsbreite [µs] / 58

Gemäß dem Weg-Zeit-Gesetz gilt: $s = v * t$

Die Schallgeschwindigkeit beträgt v_{Schall} = 343 m/s

Weiterhin gilt: s = 2*Abstand, t = Pulsbreite = Laufzeit des Ultraschallsignals

Die gemessene Entfernung soll auf dem LCD Display ausgegeben werden.

Task 2: Zeituhr (niedrige Priorität)

Unter Verwendung von Timer3 soll eine Zeituhr programmiert werden. Timer3 zählt hierzu mit der Befehlsfrequenz ($F_{OSC}/4$ = 1 MHz) aufwärts und erzeugt beim Übergang von 0xFFFF → 0000 einen Interrupt. Der Timer ist so einzustellen, dass zyklisch alle 100ms ein Interrupt auftritt und somit auch die Zeitbasis für Task 1 gebildet wird. Zur Umsetzung der Uhr ist eine Variable „Vorzähler“ zu verwenden, die in den 100ms Intervallen inkrementiert wird. Sobald diese den Wert 10 erreicht, ist folglich eine Sekunde vergangen. Die Variablen „Sekunde“, „Minute“ und „Stunde“ sollen entsprechend inkrementiert und an der LCD-Anzeige visualisiert werden. Siehe obiges Schaltbild.

Anleitung zur Entwicklung:

Verwenden Sie das CCP2 Modul im Capture-Modus, um mit Hilfe von Timer1 die Dauer der an RB3 empfangenen Echo-Pulse zu berechnen. Hierzu soll das CCP2 Modul hochpriorisierte Interrupts bei allen Signalflanken an RB3 erzeugen.

Setzen Sie vor dem Aussenden eines Trigger-Pulses den Zählwert von Timer1 zurück. Bei einer steigenden Flanke des Echo-Signals ist der Capture-Wert des CCP2 Moduls als Subtrahend für die spätere Berechnung zwischenzuspeichern. Bei der fallenden Flanke (die Reflexion des Echosignals wurde empfangen) ist der gespeicherte Wert von dem aktuellen Capture-Wert zu subtrahieren. Teilt man die Differenz nun durch 58, so erhält man den gesuchten Abstand in Zentimeter. Sollte Timer1 überlaufen (TMR1IF=1), so wurde kein Objekt detektiert und der Abstand ist auf den Maximalwert zu setzen.

Niedrigpriorisierte Interrupts sollen bei Überläufen von Timer3 alle 100ms entstehen. Dabei soll der ermittelte Objektabstand auf dem LCD Display ausgegeben werden. Falls kein Objekt detektiert wurde, soll eine entsprechende Anzeige erfolgen. In der niedrigpriorisierten ISR soll zudem der Vorzähler der Uhr inkrementiert werden. Nach jeweils 1 Sekunde sind die drei Zeitvariablen zu aktualisieren und die Uhrzeit ist auf dem LCD Display auszugeben. Weiterhin wird in dieser ISR ein 10µs langer Triggerpuls an RB1 ausgegeben, um die nächste Messung auszulösen (Hinweis: das in dem inkludierten Header-File enthaltene Makro „Nop()“ kann verwendet werden um ein Delay von 1 µs zu erzeugen). Vor der Ausgabe des Pulses ist es sinnvoll TIMER1 zurückzusetzen, um zu vermeiden, dass zwischen den beiden Flanken eines Echo-Pulses ein Zählerüberlauf stattfindet.

Anleitung zur Simulation:

Sowohl die Zeituhr als auch das Trigger- und Echosignal des Sensors können mitsamt Ausgabe der LCD-Anzeige simuliert werden.

Mit aktiver Zeile „**#define Simulator**“ sind die LCD-Befehle durch bedingte Kompilierung inaktiv (so wie auskommentiert). Wenn man die beiden Strings LCDtext1 + LCDtext2 in ein Watschfenster nimmt, kann man ersatzweise dort die per sprintf()-Befehl generierten Ausgabetexte für das LC-Display anschauen.

Überprüfung der Uhr und des Triggerimpulses (niederpriore ISR):

Zunächst kann die Zeituhr durch Ausgabe von LCDtext2 kontrolliert werden. Um den an **RB1** erzeugten Triggerimpuls sichtbar zu machen, kann der Logic Analyzer zu Hilfe genommen werden:

> View > **Simulator Logic Analyzer**

Darin mit dem Button <Channels> das Menü „Configure Channels“ öffnen und dort in der linken Spalte aus „Available Signals“ RB1 auswählen und mit „Add =>“ auf der rechten Seite unter „Selected Signals“ hinzufügen. Zur Beobachtung einen Haltepunkt dort setzen, wo RB1 wieder auf 0 zurückgesetzt wird. Im Logic Analyzer den Impuls anschauen:

Lupe (-) = Zoom Out All Axes → Impuls wird ganz rechts sichtbar.

[ ] = Zoom Box und Maus ziehen um den Impuls
→ Mehrmals anwenden, um hinein zu zoomen.

Für die Zeitmessung kann in Zusammenwirken mit Haltepunkten auch eine Stoppuhr zu Hilfe genommen werden mit > Debugger > **StopWatch**

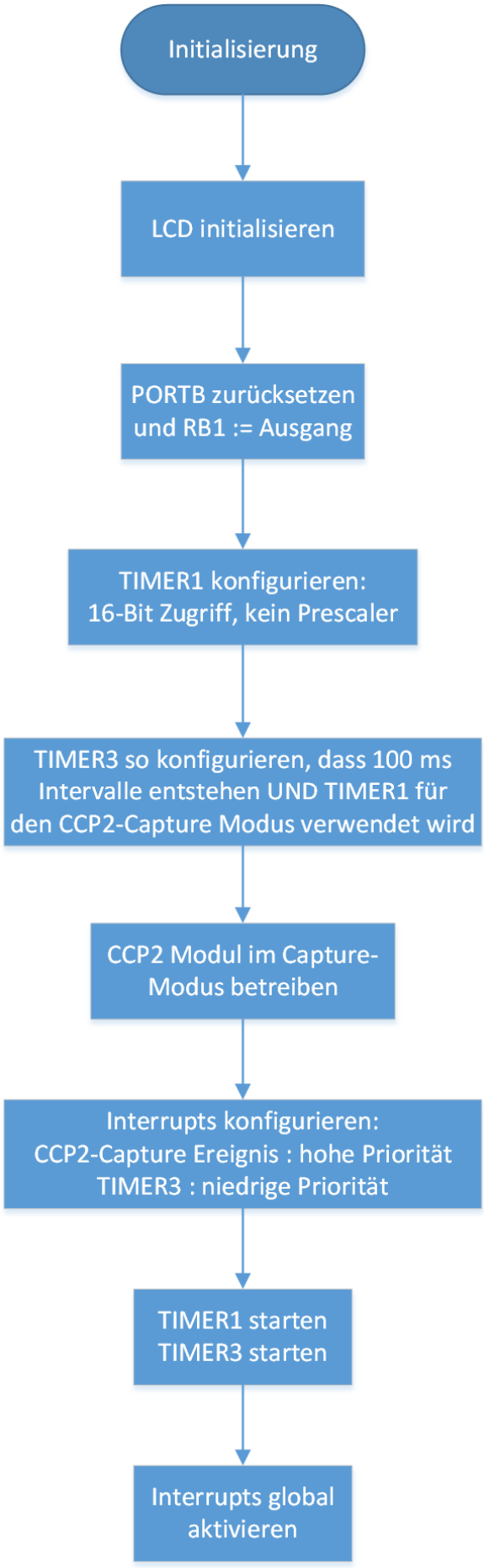
Überprüfung der Ultraschall-Abstandsmessung (höherpriore ISR):

Anders als bei der Hardware geschieht bei der Simulation die Generierung des Echoimpulses vollkommen losgelöst vom Triggerimpuls, was zur Überprüfung aber völlig ausreicht. Hierfür wird mit

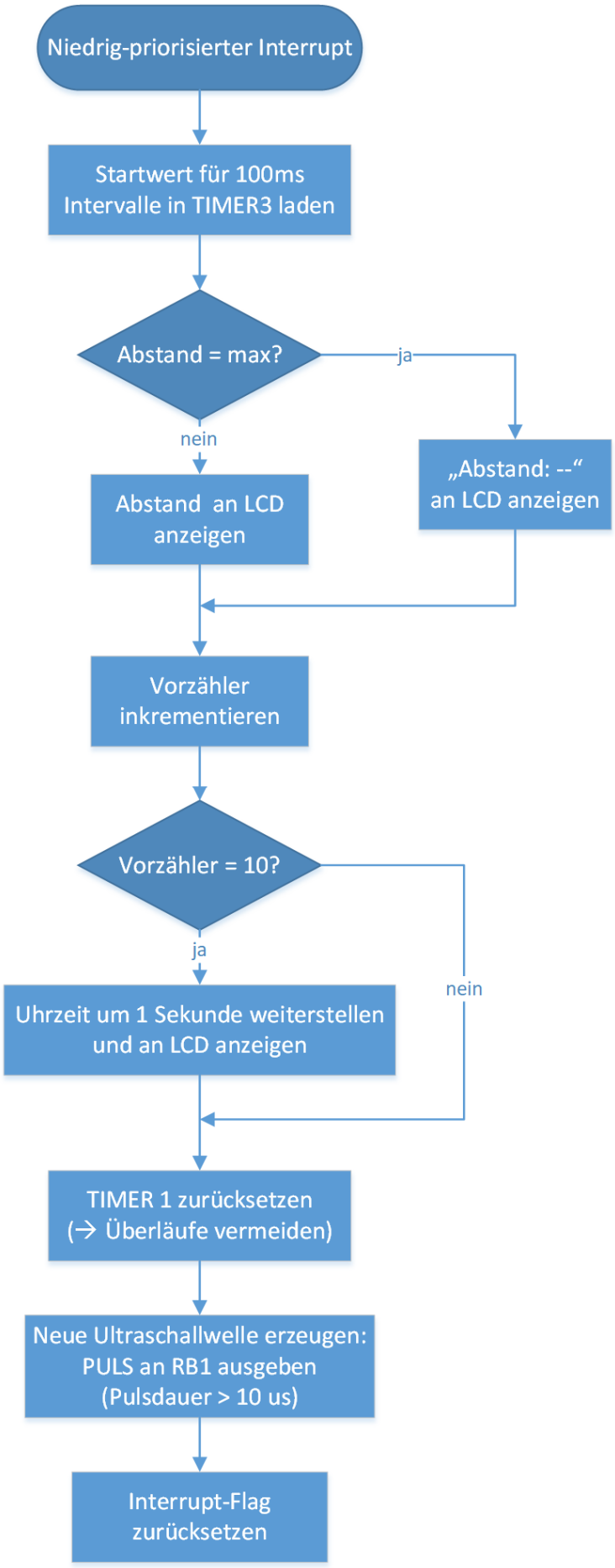
> Debugger > Stimulus > Open Workbook: „**Stimulus RB3 Ultraschall.sbs**“

ein Stimulus geöffnet, der Impulse vorgegebener Länge am Echoeingang RB3 erzeugt. Lassen Sie mit <Run> das Programm laufen und drücken Sie einen gewünschten Fire-Button, um jeweils einen Echoimpuls zu generieren, danach halten Sie das Programm an und bekommen in LCDtext1 den entsprechenden Abstand gezeigt. Dies funktioniert auch ohne Haltepunkt.

Flussdiagramm der Initialisierung:



Flussdiagramm der niedrig-priorisierten ISR:



Flussdiagramm der hoch-priorisierten ISR: