

# Rechnerarchitekturen Labor

Matrikelnummern:

- 4962704
- 3277496

## Aufbau

Der Aufbau besteht aus:

- PIC32MM0256GPM064 uC
- LCD - Display
- HC-SR04 Ultraschall Sensor

Das LCD ist folgendermaßen angeschlossen:

Pin #	Funktion	PIC32 Belegung
1	VDD	3.3V
2	VSS	GND
3	SDA	RB7 (SDA3)
4	SCL	RB13 (SCL3)
5	RST	3.3V über R=3.3kOhm
6	A (Backlight +)	3.3V
7	K (Backlight -)	GND

Der Ultraschall Sensor ist wie folgt angeschlossen:

Funktion	PIC32 Belegung
VCC	5V
Trig	RA12 (OCM1A)
Echo	RA9 (5V tolerant)
GND	GND

## Programmierung

### Implementierte Funktionalitäten

- Auslesen des Ultraschall Sensors
  - Trigger Signal durch **Output Compare Unit**
    - \* Mode: Dual Edge Compare
    - \* Trigger Signal: 10us

- \* Zeit, die Ultraschall Sensor wartet:  $8 * 1/40\text{kHz}$
- \* Antwortzeit von Ultraschall Sensor: 38ms
- \* Zeit für eine Periode:  $t = 10\mu\text{s} + 8 * 1/40\text{kHz} + 38\text{ms} = 38.21\text{ms}$
- \* Periode:  $PR = 0xFFFF$  weil  $0xFFFF \times 16/24\text{MHz} = 43.69\text{ms}$ ,  
so hat man  $43.69\text{ms} - 38.21\text{ms} = 5.48\text{ms}$  Puffer für andere Operationen
- \* Steigende Flanke:  $RA = 0$
- \* Fallende Flanke:  $RB = 0xF$  weil  $0xF \times 16/24\text{MHz} = 10\mu\text{s}$
- Echo Signal wird über die **Input Capture Unit** aufgenommen
  - \* Die Input Capture Unit nutzt als Eingabe den Pin RA9 (5V tolerant)
  - \* Sie wird im *Every Rise/Fall (16-bit capture)* Modus betrieben
  - \* Nach 2 Capture Events wird ein Interrupt ausgelöst. So enthält der Buffer die Zeistempel der steigenden und fallenden Flanke
  - \* Die Input Capture Unit wird mit einem Prescaler von 1:64 betrieben
  - \* Die Impulsweite wird in einer **Interrupt Routine** berechnet
  - \* Die Umrechnung der Impulsweite in eine Distanz erfolgt in Assembly. Hier aber die Implementierung in C:

```

u32 risingEdge = CCP2BUF;
u32 fallingEdge = CCP2BUF;
u32 diff = fallingEdge - risingEdge;
diff = diff >> 1; //halbe strecke
// 125 * (24M / 64) / 2 * (64/24MHz) * 343.2m/s / 1000000 = 2.145 cm (min dist)
// 25000 * (24M / 64) / 2 * (64/24MHz) * 343.2m/s / 1000000 = 429 cm (max dist)
// (64/24M) * 343.2 = 0.0009152
u32 distance = diff * 9152 / 100000;

```

- Ausgabe auf dem LCD
  - Speichern der Distanz in einem String und ausgeben
- Ausschluss fehlerhafter Messungen und Mittelwert
  - Für die maximale Distanz beträgt die Pulsweite 25ms
  - Die daraus resultierende Distanz ist ungefähr 430 cm
  - Alle Werte über 420 cm können also höchstwahrscheinlich ignoriert werden
  - Alle Werte über 420 cm werden abgeschnitten.

- Es wird ein Ringpuffer mit 4 Werte verwendet, um einen Mittelwert zu bilden
- Anzeigen eines Balkens zur Visualisierung der Distanz
  - Als maximale Distanz wurde 64 cm gewählt
  - Bei 16 Spalten des LCDs ergibt das eine Spalte pro 4 cm
  - Es wird die Differenz zwischen der Distanz und dem Maximalwert 64 gebildet
  - $barLength = (64 - distance)/4$
- Anzeigen der Uhrzeit in der zweiten Zeile des LCDs
  - Die Register der RTCC werden mit dem Wert des Präprozessor Makros `__TIME__` initialisiert.
  - Die aktuelle Zeit wird bei jedem Durchlauf der `while(1)` Schleife abgefragt, in einen String umgewandelt und auf dem LCD ausgegeben.
- Konfiguration der Uhrzeit über ein extra Menü
  - Über den Taster S1 kann das Menü aufgerufen werden
  - Mit dem Taster S2 kann man zwischen Stunden, Minuten und Sekunden wechseln
  - Mit dem Potentiometer kann der jeweilige Wert eingestellt werden

### **Zusatzfunktionen**

- Input Capture Unit für Rückgabewert des Ultraschall Sensor
- Output Compare für Impulse des Trigger Eingangs am Ultraschall Sensor
- Werte über 420 werden ausgeschlossen, Mittelwert wird immer über die letzten 4 Werte gebildet (Ring Buffer)
- Anzeigen der Zeit (Initialisiert mit Compile-Zeit)
- Anzeigen der Entfernung mit einem Balken (nah -> Balken “voll”)
- Individuelles Setzen der Zeit über Taster
- Verwendung von Interrupts für Input Capture Unit

### **Bedienung**

Nach dem Runterladen des Programs sollte auf dem Display die Distanz in cm und der Balken zu sehen sein:

Drückt man jetzt auf den Button S3, wird im Display statt dem Balken die aktuelle Zeit angezeigt:

## **Änderung der Zeit**

Drückt man auf den Button S1 kann eine eigene Zeit konfiguriert werden:

Mithilfe des Potentiometers kann eine Zahl zwischen 0 und 23 eingestellt werden:

Drückt man auf S2 wechselt man zur Konfiguration der Minuten, mit dem Potentiometer kann dann wieder die Minutenzahl zwischen 0 und 59 eingestellt werden:

Drückt man erneut auf S2 wechselt man zur Konfiguration der Sekunden, mit dem Potentiometer kann die Sekundenzahl eingestellt werden:

Die Zeit wird automatisch geändert sobald die Zahlen geändert wurden.

Drückt man auf S1 wird das Konfigurationsmenu verlassen und die neue Zeit übernommen:

Die aktuelle Zeit ist dann überschrieben, man kann nur per "Reset" wieder zurück zur Compile Zeit, allerdings ist diese ja nicht mehr korrekt.