



# Praktikum Mikrocomputertechnik V4 (LCD)

2020

## Versuch 4: I<sup>2</sup>C

In diesem Versuch ergänzen wir unser bisheriges Programm und verwenden die I<sup>2</sup>C-Schnittstelle. Über die I<sup>2</sup>C-Schnittstelle werden die Messdaten eines Temperatursensors ausgelesen. Außerdem wird über die I<sup>2</sup>C-Schnittstelle eine LCD-Anzeige zur Visualisierung des Temperatur-Messwertes angesteuert.

### I<sup>2</sup>C:

I<sup>2</sup>C (gesprochen "I-Quadrat-C") ist eine Zwei-Draht-Kommunikationsschnittstelle. Zahlreiche Bausteine (Sensoren, Aktoren, Speicher, ...) verwenden diese Schnittstelle und können damit an den STM32-Mikrocontroller angeschlossen werden. Die Schnittstelle ist ein serieller Bus und besteht aus einer Takt- und einer Datenleitung, sie erlaubt ca. 100 Teilnehmer. Jeder Teilnehmer hat eine 7-Bit Adresse. Nach erfolgreicher Adressierung können Daten bidirektional im synchronen Halbduplex-Verfahren mit 100 kBit/s ausgetauscht werden.

Informieren Sie sich über I<sup>2</sup>C und über das Übertragungsprinzip. Eine kurze Beschreibung zum Thema I<sup>2</sup>C finden Sie in den Unterlagen zum Versuch sowie im Internet unter:

<https://www.youtube.com/watch?v=6IAkYpmA1DQ>

<http://www.i2c-bus.org/>

<https://www.mikrocontroller.net/articles/I%C2%B2C>

In diesem Versuch werden wir mit einer Bibliothek auf die I<sup>2</sup>C-Schnittstelle zugreifen, Sie brauchen dafür keine Kenntnisse über die I/O-Register des I<sup>2</sup>C-Teils des Prozessors (falls es Sie doch interessiert: siehe Seite 542 des Hardwaremanuals). Die I<sup>2</sup>C-Bibliothek besteht aus den Dateien `I2C.c` und `I2C.h`, es wird der I<sup>2</sup>C-Kanal 1 verwendet mit SCL an Port B6 und SDA an Port B7.

**Am Ende sollten Sie ein funktionierendes Projekt haben, im dem alle wesentlichen Punkte der Versuche 2 bis 4 integriert sind!**



# Praktikum Mikrocomputertechnik V4 (LCD)

2020

## Aufgabe 4.1: I<sup>2</sup>C- Temperatur

Als Temperatursensor wird der Baustein MCP9808 mit der Adresse 0x18 verwendet. Das Datenblatt finden Sie in den Unterlagen, für das Verständnis der Kommunikation mit dem Baustein ist das Kapitel 5 (Seite 16) wichtig.

Kurze Erklärung des Prinzips:

Der MCP9808 besitzt 9 Funktions-Register und ein Auswahl-Register (Registerpointer), das die Adresse (0...8) des ausgewählten Funktionsregisters enthält. Wenn man also auf eines der Funktionsregister zugreifen will, muss vorher der Registerpointer beschrieben werden.

Für uns interessant ist das Ambient Temperature Register (Register 5), ein 16-Bit Register, das neben einigen Statusbits die Temperatur in Grad Celsius mit 4 binären Nachkommastellen enthält (siehe Seite 24 im Datenblatt).

Zusätzlich wird noch das Config-Register verwendet (Register 8, Seite 29), hier werden in Bit 0 und 1 verschiedene Temperaturlösungen eingestellt. Es kann zunächst die Voreinstellungen des Bausteins mit der höchsten Auflösung von 0,0625 °C verwendet werden.

### Aufgabe:

Verwenden Sie Ihr Programm aus dem letzten Versuch. Fügen Sie in das Projekt die C-Module `I2C.c` und `Temperatur.c` ein. Ergänzen Sie die Funktionen im Modul `Temperatur.c` so, dass die Temperatur ausgelesen werden kann und mit der Funktion `set_TempRes` die Auflösung geändert werden kann. Die von `I2C.c` zur Verfügung gestellten Funktionen sind weiter unten aufgelistet.

Die Execute-Funktion im Hauptprogramm soll um die Kommandos `tr` und `tsx` ergänzt werden. Das Kommando `tr` liest die Temperatur vom Temperatursensor aus und zeigt die Temperatur mittels `putty` an, das Kommando `tsx` setzt die Auflösung der Temperaturmessung (`x=0 => 0,5°C`, `x=1 => 0,25°C`, etc.).

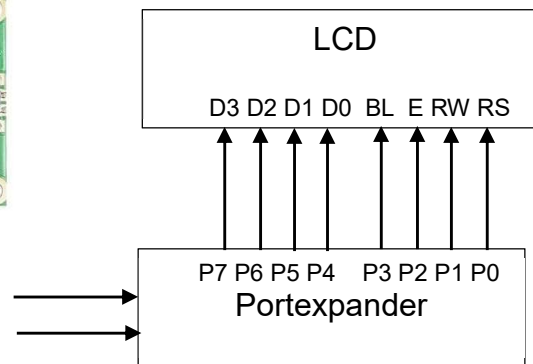
Analysieren Sie das I<sup>2</sup>C-Protokoll mit Hilfe des Oszilloskops.

# Praktikum Mikrocomputertechnik V4 (LCD)

2020

## Aufgabe 4.2: I<sup>2</sup>C LCD-Anzeige

Die LCD-Anzeige besteht aus zwei Modulen, einem sog. Portexpander und der eigentlichen 2-zeiligen LCD-Anzeige.



Der Portexpander basiert auf dem Baustein PCF8574 und ist ein digitaler 8-Bit-Port, der über die I<sup>2</sup>C-Schnittstelle angesprochen wird. Die LCD-Anzeige verwendet den Baustein HD44780 und ist an den digitalen Port angeschlossen.

Die Leitungen P4-P7 werden zu Datenübertragung verwendet, P0 ist mit dem Pin RS und P2 mit dem Enable-Eingang des LCD-Bausteins verbunden. Über P3 wird die Hintergrundbeleuchtung angesteuert (1 = an).

### Übertragungsschema:

RS = 0 => Es wird ein Kommando übertragen, RS = 1 => Es werden Daten übertragen.

RW = 0 => Write

Der LCD-Baustein übernimmt die Daten mit der fallenden Flanke des Signals E, d.h. die Daten müssen in 2 Schritten übertragen werden. Im 1. Schritt ist P2 gesetzt, im zweiten Schritt mit identischen Daten liegt P2 auf 0. Damit wird die fallende Flanke an E realisiert. Nachdem nur 4 Datenleitungen verwendet werden, muss ein Byte in zwei Teilen übertragen werden. Zuerst werden die oberen 4 Bit (D4-D7), dann die unteren 4 Bit (D0-D3) übertragen.

### Aufgaben:

**4.2.1** Binden Sie die Datei `LCD.c` und `LCDInit.o` in Ihr Projekt ein. Verwenden Sie die Funktion `lcd_init()` zum einmaligen Initialisieren des Displays. Das Modul `LCD.c` enthält die Funktionen `lcd_send_cmd` und `lcd_send_data` zum Ansteuern des Displays. Ergänzen Sie diese Funktion `lcd_send_data` entsprechend dem obigen Übertragungsschema.

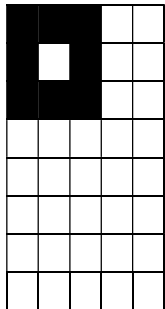
Fügen Sie zum Testen der Übertragung in die Funktion `ExecuteCmd` die Kommandos `dc[cmd]` und `dd[data]` ein, wobei `cmd` und `data` für eine Hexzahl stehen.

Testen Sie die LCD-Kommandos laut Datenblatt „LCD1602“ und geben Sie einen Text aus.

# Praktikum Mikrocomputertechnik V4 (LCD)

2020

**4.2.2** Schreiben Sie nun eine Funktion, um das Symbol „Grad“ im Zeichensatz des Displays an ASCII-Position 0 abzuspeichern. Rufen Sie diese Funktion einmalig in `main()` auf und testen Sie die Ausgabe des selbstdefinierten Symbols mit den Kommandos aus 4.2.1. Siehe Dokument „LCD1602“.



## Aufgabe 4.3: Temperatur anzeigen

Nun soll zyklisch die Temperatur abgefragt und im Display angezeigt werden. Verwenden Sie dazu die Funktion `sprintf` und geben Sie dann die Zeichen des erzeugten Strings mit `lcd_send_data()` auf dem Display aus. Nutzen Sie dazu das selbstdefinierte Symbol „°“ und ein dahinter gestelltes „C“

Zusätzlich soll der Schrittmotor als analoge Temperaturanzeige verwendet werden. Dazu müssen Sie aus dem Temperaturwert die Sollposition des Schrittmotors berechnen und die Skalierung so bemessen, dass der Temperaturbereich von 20 Grad Celsius bis ca. 30 Grad Celsius auf eine 180 Grad-Drehung des Schrittmotors abgebildet wird.

Der zyklische Aufruf erfolgt am einfachsten, wenn Sie im Handler des SysTick-Timers, der bereits für den Schrittmotor benötigt wird, eine globale Variable Ticks hochzählen und im Hauptprogramm diese Variable abfragen. Die Temperaturmessung mit Anzeige soll im Hauptprogramm einmal pro Sekunde aufgerufen werden.



# Praktikum Mikrocomputertechnik V4 (LCD)

2020

## I2C- Bibliothek:

Folgende Funktionen werden für die *I2C-Bus Kommunikation* zur Verfügung gestellt:

- *i2c.c*:

```
/* =====  
I2C Init Routine  
===== */  
void I2C_init(  
    unsigned char address          // 7-Bit I2C Adresse des Masters (µC)  
)  
  
/* =====  
I2C Write n Byte From Buffer Routine  
===== */  
void I2C_write(  
    const unsigned char * buf      // Puffer der zu sendenden Byte  
    ,unsigned int n                // Anzahl zu sender Byte  
    ,unsigned char dest           // 7-Bit I2C Adresse des Busteilnehmers  
)  
  
/* =====  
I2C Read n Byte To Buffer Routine  
===== */  
void I2C_read(  
    unsigned char * buf            // Puffer für eingelesene Byte  
    ,unsigned int n                // Anzahl zu lesender Byte  
    ,unsigned char dest           // 7-Bit I2C Adresse des Busteilnehmers  
)
```

### Typische Werte:

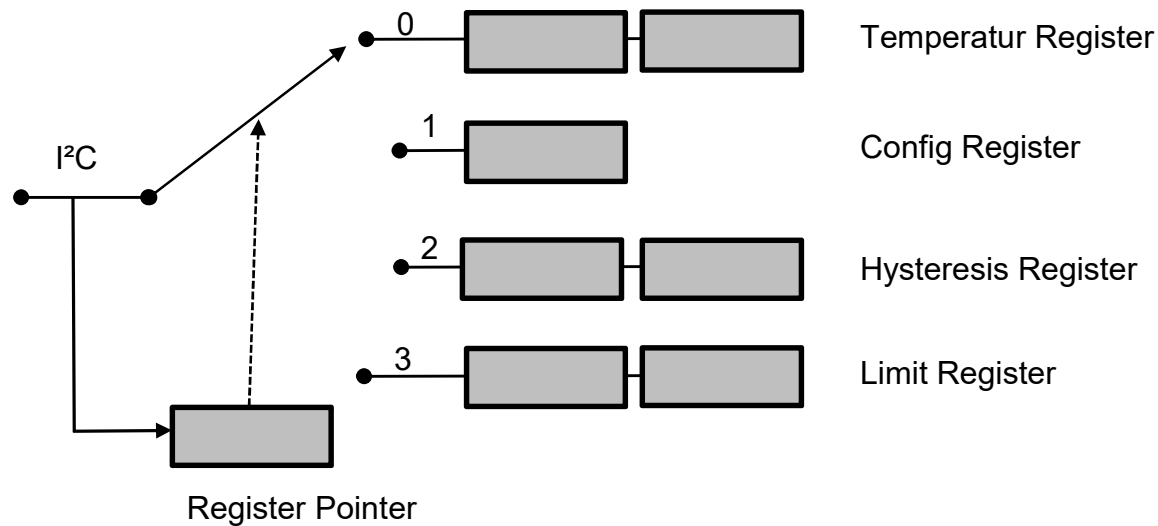
I<sup>2</sup>C-Adresse des Busmasters (µC): 0x08 (d.h., niedriger als jeder Slave)

I<sup>2</sup>C-Adresse der Busteilnehmer: aus den Datenblättern entnehmen, muss die *7-Bit Adresse* sein!

Puffer buf: ausreichend lang für Daten und Befehle machen, mindestens n Byte!

## Hinweise

Bei der Verwendung des Register-Pointers im Temperatursensor hilft die Vorstellung, den Register-Pointer wie einen Umschalter zu verwenden.



Um also die Temperatur auszulesen muss vorher der Register Pointer mit 0 beschrieben werden. Anschließend können die 2 Bytes des Temperaturregisters gelesen werden