

Medizinisch-Technische Informatik an der Hochschule Reutlingen:

„... das Bachelor Studium mit der Eintrittskarte in die Erfolgswelt der Medizintechnik“

<http://www.inf.reutlingen-university.de/studium/bachelor/medizinisch-technische-informatik/>

Vorlesungsbegleitblatt – Eingebettete Systeme und Robotik: SS2020


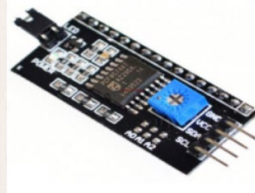
Workshop LCD1602

Vorwort

Im Rahmen des diesjährigen Projektes kann das LCD1602 zur Anzeige von Textausgaben zum Einsatz kommen. Für die Anzeige von Informationen stehen Bei diesem LCD 2 Zeilen mit jeweils 16 Zeichen zur Verfügung.



Die Beschaffung erfolgt über die üblichen Internetquellen, wie z.B. ...

 <p>NEU</p> <p>Salesman-Kuri</p>	<p>HD44780 LCD 1602 16x02 Modul blau</p> <p>Lieferumfang: 1x LcdModul / LCD 1602 / 16x02 Modul mit blauem Display, 16Pin Anschluss und Hintergrundbeleuchtung</p> <p>Auf Lager</p>	<p>2,50 € inkl. MwSt.</p> <p>zzgl. Versandkosten</p> <p>In den Warenkorb</p> <p>Mehr</p>
 <p>Salesman-Kuri</p>	<p>I2C Interface Modul PCF8574AT</p> <p>Lieferumfang: 1x I2C Interface Modul mit PCF8574AT Chip Specification# I2C Address: 0x27 oder # Backlight# Supply voltage: 5V# Size: 82x35x18 mm # Come with IIC interface, which can be connected by DuPont Line</p> <p>Auf Lager</p>	<p>1,89 € inkl. MwSt.</p> <p>zzgl. Versandkosten</p> <p>In den Warenkorb</p> <p>Mehr</p>

Es fällt auf, dass das Display meistens zusammen mit dem I²C Interface Modul PCF8574AT angeboten wird. Dieses Modul dient als Schnittstelle zwischen einem Mikrocontroller und dem LCD1602 und vereinfacht die Ansteuerung des LCD über eine Mikrocontroller. Da die Module, wie meistens, von chinesischen Lieferanten stammen, ist

es schwierig den Hersteller ausfindig zu machen und entsprechend Datenblätter zu finden. Solange jedoch keine verbindlichen Datenblätter verfügbar sind, muss man einiges an Detektivarbeit leisten, um die Funktionsweise der Bausteine zu erkunden. Dies ist unvermeidbar, sofern man nicht Gefahr laufen will, das Launchpad u.U. zu zerstören.

Auch wenn es sich bei den Modulen offenbar um Standardbausteine handelt, die von vielen unterschiedlichen Lieferanten angeboten werden, ist es grundsätzlich wichtig die Produkte selbst zu prüfen.

Wie den oben gezeigten Abbildungen zu entnehmen ist, stellt der Verkäufer nur minimale Informationen zur Verfügung und es muss daher über andere Quellen bzw. durch Experimente überprüft werden:

- a) Wie die Programmierung des Displays erfolgt
- b) Wie die Spannungsversorgung des Displays erfolgen kann (reichen hierfür auch die 3,3V des Launchpads?)
- c) Über welche I²C Befehle der Portexpander angesprochen werden kann
- d) Wie die Hardwarekonfiguration des I²C Anschlusses aussieht
- e) Welche weiteren Funktionen des I²C Modul bietet und wie diese genutzt werden können (z.B. Poti zur Kontrasteinstellung oder Steuerpin für die Hintergrundbeleuchtung des Displays)

Ziel soll am Ende sein, für das Display ein API zu erstellen, welches die Einbindung des Displays in andere MSP430FR2355 Projekte ermöglicht.

Hintergrundinformationen

Um die Module zu erkunden muss man sich zuerst auf die Suche nach Datenblättern und ggf. nach Schaltplänen machen.

Anzeigemodul LCD1602

Das LCD1602 ist schon seit vielen Jahren auf dem Markt und wurde ursprünglich über eine 8-Bit breite Datenverbindung angesteuert. D.h. die anzuzeigenden Buchstaben werden byteweise über einen 8-Bit Datenbus an das Display übertragen. Zusätzlich sind für die Übertragung der Daten noch Steuerleitungen erforderlich. Als Schnittstelle zu der LC-Display Hardware wurde ursprünglich der Controller HD44780 von Hitachi verwendet. Der Befehlssatz dieses Controllers definierte einen inoffiziellen Standard, der von allen späteren Herstellern ebenfalls übernommen wurde.

Will man das Display also selbst programmieren, so muss man sich mit diesem Befehlssatz auseinandersetzen. Informationen hierzu findet man im zugehörigen Datenblatt:

<https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>

Allgemeine Informationen zur Nutzung des Display Controllers findet man z.B. bei Wikipedia:

<https://de.wikipedia.org/wiki/HD44780>

I²C Schnittstelle zum Display - PCF8574A

Die Nutzung eines 8-Bit breiten Daten/Steuerbusses belegt bei einem Mikrocontroller 8 DIO-Leitungen. Zusätzlich werden noch 3 Steuerleitungen benötigt, so dass für den Anschluss des Displays 11 DIO-Pins eines Mikrocontrollers erforderliche wären.

Da bei den Mikrocontrollern die Zahl der verfügbaren DIO-Pins begrenzt ist, werden für die Ansteuerung zwei Tricks verwendet:

1. Das Display kann mit einem 8-Bit breiten Datenbus + 3 Steuerleitungen angesteuert werden oder mit einem 4-Bit breiten Datenbus + 3 Steuerleitungen. Der 4-Bit Modus hat somit den Vorteil, dass für die Datenübertragung und die Steuerung insgesamt nur 7 Bit benötigt werden, so dass die komplette Steuerung des Displays über einen 8-Bit Bus erfolgen kann. Der Nachteil besteht darin, dass ein 8-Bit Textzeichen nun in zwei Schritten à 4 Bit übertragen werden muss. Die Übertragungszeit verdoppelt sich also.
2. Wenn für die Ansteuerung des Displays nur 7 Bit benötigt werden, so kann für die Übertragung dieser 7 Bit ein 8-Bit I²C Portexpander eingesetzt werden. Dieser liest vom I²C Bus seriell 8 Bit ein und gibt diese anschließend auf einem 8-Bit Datenbus parallel aus. D.h. der Portexpander übernimmt die Funktion eines seriell / parallel Übertragers (Schieberegister mit Latch).

Anstelle der ursprünglich benötigten 11 Leitungen für das Display, muss der Mikrocontroller nun also nur den I²C Bus, also 2 Leitungen, ansteuern.

Theoretisch kann mit den Portexpandern vom Typ PCF8574 über den I²C Bus die Zahl der nach extern bereitgestellten DIO beinahe beliebig erweitert werden, wenn der MC selbst zu wenige DIOs besitzt. Dies wird nur dadurch begrenzt, dass jeder der Bausteine eine eigene I²C Adresse besitzen muss, um von MC eindeutig adressiert werden zu können.

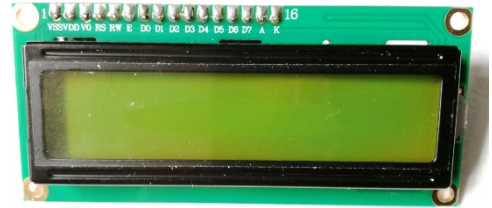
Das PCF8574 Modul stellt jedoch nicht nur einen Portexpander zur Verfügung, sondern, wie eingangs bereits erwähnt, auch

- die Möglichkeit der Einstellung des Displaykontrastes über ein Poti
- die Spannungsversorgung der Hintergrundbeleuchtung,
- eine Möglichkeit die Hintergrundbeleuchtung über den MC an- und ausschalten zu können. Da für die Steuerung des Displays nur 7 Bit erforderlich sind, kann das noch freie Bit für die Steuerung der Hintergrundbeleuchtung eingesetzt werden.



Verbindung des Portexpandermoduls mit dem Display

Das Portexpandermodul ist so aufgebaut, dass es direkt mit dem Display zusammen- gesteckt werden kann. Die Pins müssen dann nur noch verlötet werden.



Dummerweise findet sich kaum Dokumentation darüber, welche Leitungen des Portexpanderbausteins PCF8574 nun mit den jeweiligen Pins des Displays verbunden sind. Der Verkäufer stellt nur folgende Information zur Verfügung:

Bei der Kombination Display und Interface handelt es sich um 2 Einzelmodule, die bei Kauf an Sie versandt werden.

Sollten Sie einen speziellen Farbwunsch haben bitte angeben- Wir führen blau oder gelb-grün im Sortiment.

Standard 16X2 character LCD module (backlight)
mit HD44780 Controller Version.

1602 with standard 16 pin connector including:

Pin 1: VSS is in power

Pin 2: VDD 5V positive power supply connection

Pin 3: VO LCD contrast adjustment for the client then the contrast is the weakest power ground power the highest contrast the contrast is too high will have a "ghosting" when used with a 10K potentiometer to adjust contrast

Pin 4: RS for the register, selection the selection of data register high low - select the instruction register.

Pin 5: R / W to read_and write signal lines when high read low write. When the RS_and RW can be written together for the low level command or display the address when RS is high low RW can be read busy signal when RS is low for the high RW can write data.

Pin 6: E side to side to make when E Duanyou high jump into a low the LCD module execute the command.

Pin 7 to 14: D0 ~ D7 for the 8-bit bidirectional data line.

Pin 15 feet: Backlight positive supply

Pin 16 feet: Backlight negative supply

IIC/I2C/TWI 1602 Serial LCD Module

As the pin resources of Arduino controller is limited, your project may be not able to use normal LCD shield after connected with a certain quantity of sensors or SD card. However, with this I2C interface LCD module, you will be able to realize data display via only 2 wires. If you already has I2C devices in your project, this LCD module actually cost no more resources at all. It is fantastic for Arduino based project.

Specification

I2C Address: 0x27 oder 0x3F

Backlight

Supply voltage: 5V

Size: 82x35x18 mm

Come with IIC interface, which can be connected by DuPont Line

Unter Berücksichtigung des Pinlayouts des PCF Bausteins und mit Hilfe eines Multimeters („Durchklingeln“ der Verbindungen) kommt man schließlich zu folgender Zuordnung:

PCF	A0	A1	A2	P0	P1	P2	P3	Vss	P4	P5	P6	P7	!INT	SCL	SDA	VDD
Pin	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LCD	-	-	-	RS	RW	E	-	GND	D4	D5	D6	D7	-	-	-	VDD

Pin: Pinnummer am Gehäuse des PCF8574

PCF: Bedeutung / Funktion des Pins am Gehäuse des PCF8574

LCD: Signalleitung des LCD, mit dem das jeweilige Pin des PCF8574 verbunden wird

Somit gilt für den 4-Bit Steuermode des Displays:

P0 – P3: Steuerung der Kommunikation mit dem Display

P4 – P7: Daten Kommando oder Text: Pro Byte müssen 2 Nibble übertragen werden

Bei der Verbindung werden also 3 Bit als Steuerleitungen (RS, R/W, E) verwendet, 4 Bit als Datenleitungen (D4-D7). Außerdem wird die Kontraststeuerleitung des LCD mit einem Poti gegen VDD (positive Versorgungsspannung) verbunden, so dass mit dem Poti der Kontrast eingestellt werden kann. Ebenso wird die Hintergrundbeleuchtung mit VDD verbunden, sofern der Jumper auf dem Port-Expander gesetzt ist.

Wie man sieht, sind nur die Datenleitungen D4 -D7 mit dem Port-Expander verbunden, d.h. das LCD muss also im 4-Bit Mode angesteuert werden. Wie dies funktioniert, kann aus dem Datenblatt des LCD1602 entnommen werden, bzw. aus einer allgemeinen Beschreibung des HD44780 Standards:

- For 4-bit interface data, only four bus lines (DB4 to DB7) are used for transfer. Bus lines DB0 to DB3 are disabled. The data transfer between the HD44780U and the MPU is completed after the 4-bit data has been transferred twice. As for the order of data transfer, the four high order bits (for 8-bit operation, DB4 to DB7) are transferred before the four low order bits (for 8-bit operation, DB0 to DB3).

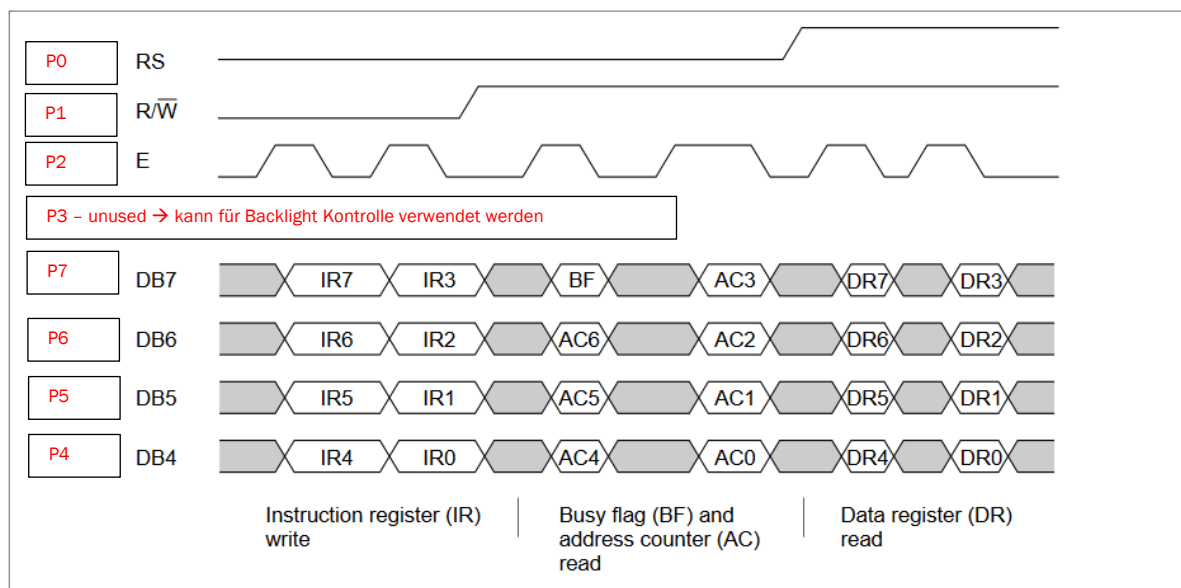
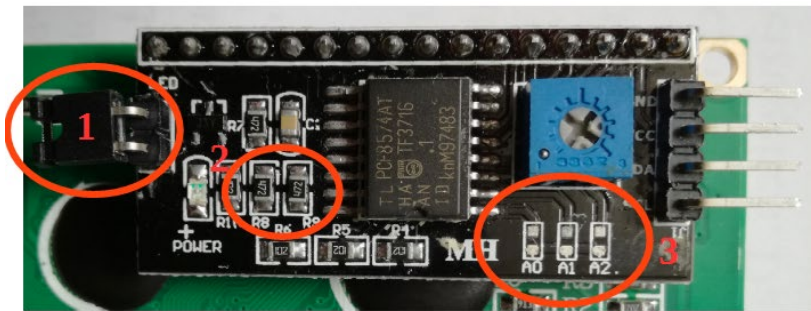


Figure 9 4-Bit Transfer Example

Bit3 des Port-Expanders (eines der 8-Datenbits) ist nicht mit dem LCD verbunden (wird nicht benötigt). Anhand dieses Bits kann die Hintergrundbeleuchtung des Moduls an- oder ausgeschaltet werden. Nochmals: Dieses Bit hat keine Wirkung auf das LCD, sondern richtet sich nur an das Portexpander Modul.

Verbindung des Portexpandermoduls mit dem Mikrocontroller

Bei näherer Betrachtung des Aufbaus des Portexpandermoduls lassen sich noch folgende Eigenschaften erkennen:



DW OR N PACKAGE
(TOP VIEW)

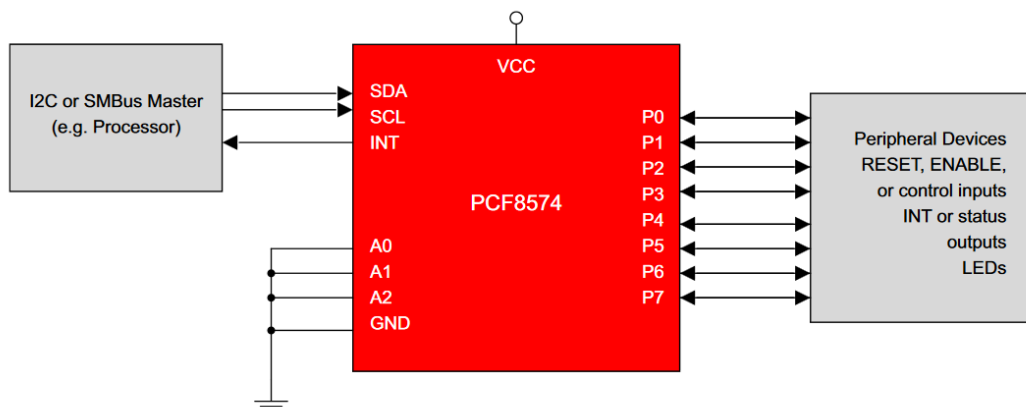
A0	1	16	V _{CC}
A1	2	15	SDA
A2	3	14	SCL
P0	4	13	INT
P1	5	12	P7
P2	6	11	P6
P3	7	10	P5
GND	8	9	P4

1: Jumper für Hintergrundbeleuchtung: wird dieser Jumper abgezogen, so ist die Beleuchtung deaktiviert

2: 4,7k Pullup Widerstände für I²C (SCL und SDA)

3: Lötbrücken für I²C Adresse

Dem Datenblatt des PCF8574 von TI kann folgendes Anschlussschema entnommen werden:



<http://www.ti.com/lit/ds/symlink/pcf8574.pdf>

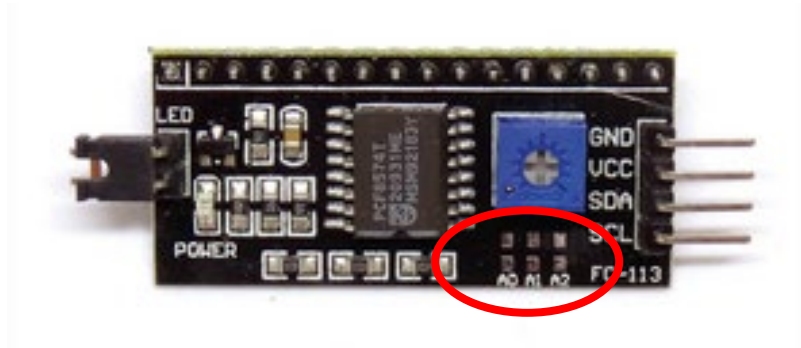
Der Baustein PCF8574A setzt, laut Datenblatt von TI oder NXP, die I²C Spezifikation vollständig um, so dass eine Ansteuerung über das I²C Modul des MSP430 möglich ist. D.h. die Steuerung der Kommunikation wird über das I²C Modul erledigt, so dass die CPU von der Steuerung des I²C Busses befreit ist und die Sleep Modes des MSP430 nutzen kann.

SCL und SDA werden auf dem Port-Expander bereits auf dem Expander Board mit jeweils einem 4,7k Widerstand (R8 und R9) gegen VDD des Port-Expanders verbunden, so dass die Verwendung von externen Pullup Widerständen für den I²C Bus nicht erforderlich ist. Hierbei müssen aber unbedingt die Hinweise zur Wahl der Versorgungsspannung (siehe

unten) beachtet werden. Andernfalls droht die Zerstörung des Launchpads bzw. des MSP430.

I²C-Adresse des Bausteins

Auf dem Board sind die Lötbrücken für A0, A1 und A2 nicht gesetzt.



Die Brücken stellen eine Verbindung zu GND her → ohne Brücke liegt der Pegel auf H (interner Pullup Widerstand, vermutlich). Die aus der Beschaltung resultierende Adresse kann dem NXP-Datenblatt zum PCF8474A entnommen werden.

NXP Semiconductors

PCF8574; PCF8574A

Remote 8-bit I/O expander for I²C-bus with interrupt

Table 5. PCF8574A address map

Pin connectivity			Address of PCF8574A								Address byte value		7-bit hexadecimal address without R/W
A2	A1	A0	A6	A5	A4	A3	A2	A1	A0	R/W	Write	Read	
V _{SS}	V _{SS}	V _{SS}	0	1	1	1	0	0	0	-	70h	71h	38h
V _{SS}	V _{SS}	V _{DD}	0	1	1	1	0	0	1	-	72h	73h	39h
V _{SS}	V _{DD}	V _{SS}	0	1	1	1	0	1	0	-	74h	75h	3Ah
V _{SS}	V _{DD}	V _{DD}	0	1	1	1	0	1	1	-	76h	77h	3Bh
V _{DD}	V _{SS}	V _{SS}	0	1	1	1	1	0	0	-	78h	79h	3Ch
V _{DD}	V _{SS}	V _{DD}	0	1	1	1	1	0	1	-	7Ah	7Bh	3Dh
V _{DD}	V _{DD}	V _{SS}	0	1	1	1	1	1	0	-	7Ch	7Dh	3Eh
V _{DD}	V _{DD}	V _{DD}	0	1	1	1	1	1	1	-	7Eh	7Fh	3Fh



Adresse ohne Lötbrücken: 0x3F (dez 63)

Machbarkeitsbetrachtung

Versorgungsspannung

Es gibt die LDC1602 Anzeigen sowohl in einer 3,3V Version (etwas schwieriger zu finden) als auch in einer 5V Version (typische Versorgungsspannung beim Anschluss an ein Arduino Board). Während der Portexpanderbaustein lt. Datenblatt mit 3,3V

Versorgungsspannung klar kommt, kann das Display ggf. die erwünschte Funktion nicht erbringen.

Welche Version des Displays vorliegt, ist den Displays von außen meist nicht anzusehen. Man muss sich also auf den Angaben des Verkäufers verlassen (und hierbei feststellen, dass die Verkäufer z.T. selbst nicht wissen, was sie da verkaufen).

Also – es bleibt uns auch hier nichts Anderes übrig, als eine Textstrategie auszudenken.

Folgende Fälle werden nachfolgend betrachtet:

- a) Das Display wird mit 3,3V betrieben – wenn es sich nicht ansprechen lässt → offenbar nicht 3,3V kompatibel.
- b) Das Display wird an 5V angeschlossen, der Portexpander an 3,3V ... geht das?
- c) Das Display und der Portexpander werden an 5V angeschlossen. Dann muss das Launchpad aber vor zu hohen Signalspannungen geschützt werden. Dies erfordert die Modifikation des Portexpandermoduls oder den Einsatz eines Levelkonvertierers.
- d) I²C Bus aber vom Launchpad aus mit 3,3V Pullup Widerständen versehen werden

Fall A: Port Expander Vcc = 3,3V; Display Vcc = 3,3V

Der I²C-Port Extender kann über Vcc des Launchpads versorgt werden. Die Ruhestromaufnahme beträgt 12mA @3,3V mit Hintergrundbeleuchtung, bzw. 6mA @3,3V ohne Hintergrundbeleuchtung (Jumper abgezogen). Lt. Spezifikation des PCF8574 arbeitet dieser zwischen 2,6V – 6V; also kein Problem für unsere Anwendung.

Zum Display finden sich keine näheren Angaben, ob es bereits ab 3,3V ausreichend Kontrast aufweist, damit die Anzeige abgelesen werden kann (hierzu Poti auf der Rückseite im Uhrzeigersinn an den Anschlag drehen). Dies muss in einem Test herausgefunden werden.

Fall B: Port Expander Vcc = 3,3V; Display Vcc = 5V

In diesem Fall muss das Display so eingestellt werden, dass seine Datenleitungen nur als Input dienen, also selbst keine Daten versenden; dies geschieht am besten, indem die R/W-Steuerleitung des LCD auf GND gelegt wird. Somit kommt das Display nie in den Read Mode (vom MC aus gesehen) und sollte also keine 5V Signale an die IO-Pins des Port Expanders anlegen. Dies wäre der Tod der IO-Pins des Expanders, da der Expander in diesem Fall ja nur mit 3,3V betrieben wird.

Ist das (R/W) Steuerbit hingegen mit einem IO-Pin des Port Expanders verbunden, so muss dafür Sorge getragen werden, dass das Signal am Pin immer auf 0V liegt. Auch nur das kurzzeitige Hochschalten des Pins kann dazu führen, dass dann die IO-Pins der Datenleitungen zerstört werden. Diese Lösung ist riskant, da nicht für jeden Moment auszuschließen ist, dass das R/W Pin nicht doch einmal kurz auf H-Pegel liegt.

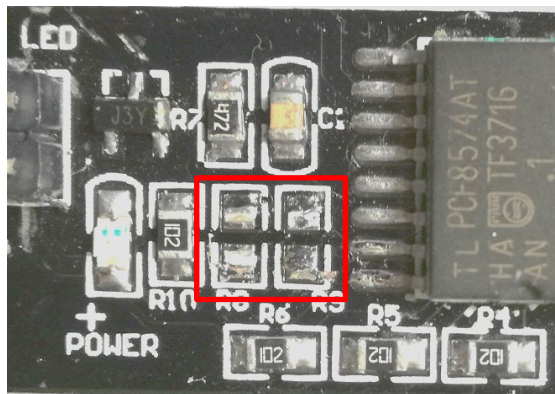
Bei dieser Lösung ist es nicht möglich, den Status des Displays abzufragen: Nach jeder Übertragung eines Kommandos oder Zeichens kann D7 (busy flag) eingelesen werden, wenn das R/W auf H-Pegel gelegt wird. Solange das D7 gesetzt ist, können keine neuen Daten vom LCD entgegengenommen werden.

Wird der LCD Status (busy flag) vom MC nicht ausgewertet, so muss nach jedem übertragenen Wert eine Mindestzeit gewartet werden, bis das nächste Datum übertragen werden kann. Die Wartezeiten sind im HD44780 Datenblatt hinterlegt.

Fall C: Port Expander Vcc = 5V; Display Vcc = 5V

Für diesen Fall muss der Port Expander baulich verändert werden: Auf dem Board befinden sich die Widerstände R8 und R9 (4,7kOhm). Dies sind Pullup Widerstände, die die SCL und SDA-Leitung auf V_{DD} des Port Expanders legen, also in diesem Fall auf 5V.

Dies ist jedoch nicht mit dem MSP430 verträglich, da dieser mit 3,3V versorgt wird. Hier darf die Spannung für die I²C-Leitungen maximal auf dem Pegel der Versorgungsspannung des Launchpads liegen (also z.B. auf 3,3V), d.h. die Pullup Widerstände dürfen nicht mit 5V (V_{DD} des Port Expanders) sondern nur mit 3,3V (V_{CC} des Launchpads) verbunden werden.



Hierzu müssen die Widerstände R8 und R9 mit einem Lötkolben entfernt werden (siehe Abbildung) und es müssen externe Pullup Widerstände (4,7kOhm) gegen 3,3V angeschlossen werden.

Laut Spezifikation des PCF8574 genügt für das Erkennen eines H-Pegels auf den I²C-Leitungen ein Pegel von $0,7 \cdot V_{DD}$, also 3,5V @V_{CC}=5V. Diese Vorgabe wird mit den 3,3V des Launchpads knapp nicht erfüllt, der I²C Bus

könnte also ggf. trotz des Umbaus nicht funktionieren. In Fällen, bei denen 3,3V als I²C-Pegel des Slaves nicht genügen, ist es erforderlich einen sogenannten bidirektionalen Pegelkonverter zwischen den MSP430 und den Slave zu schalten, so dass Master und Slave mit dem erforderlichen Pegel arbeiten können.

Fazit

Alle drei Vorschläge kommen in Betracht und sollten in folgender Reihenfolge umgesetzt werden:

- 1) Lösung A ist der Vorrang zu geben, in der Hoffnung, dass das LCD mit 3,3V klar kommt. Hinweis vorweg: Der Kontrast ist ziemlich flau, aber man kann das Display noch ablesen
- 2) Sollten 3,3V für das Display nicht genügen, so kann Lösung C umgesetzt werden: Hierzu die Pullup Widerstände R8 und R9 entfernen, was jedoch ggf. ebenfalls nicht

zum Ziel führt. Die einzige verbleibende Variante ist dann noch der Einsatz eines Pegelkonvertierers.

- 3) Lösung B macht Sinn, wenn das R/W Pin auf Masse gelegt wird. Hierzu muss die Kombination aus Port-Expander und LCD jedoch auch baulich verändert werden: die Verbindung der R/W-Leitung auf dem Pfostenstecker zwischen LCD und Port-Expander muss durchtrennt werden und die R/W-Leitung des Displays mit GND verbunden werden.

Weitere Informationen aus dem NXP-Datenblatt PCF8574A

NXP PCF8574A Product specification (2002)

Overview

The PCF8574/74A provides general-purpose remote I/O expansion via the two-wire bidirectional I²C-bus (serial clock (SCL), serial data (SDA)).

The devices consist of eight quasi-bidirectional ports, 100 kHz I²C-bus interface, three hardware address inputs and interrupt output operating between 2.5 V and 6 V. The quasi-bidirectional port can be independently assigned as an input to monitor interrupt status or keypads, or as an output to activate indicator devices such as LEDs. System master can read from the input port or write to the output port through a single register.

The low current consumption of 2.5 µA (typical, static) is great for mobile applications and the latched output ports directly drive LEDs.

The PCF8574 and PCF8574A are identical, except for the different fixed portion of the slave address. The three hardware address pins allow eight of each device to be on the same I²C-bus, so there can be up to 16 of these I/O expanders PCF8574/74A together on the same I²C-bus, supporting up to 128 I/Os (for example, 128 LEDs).

The active LOW open-drain interrupt output ($\overline{\text{INT}}$) can be connected to the interrupt logic of the microcontroller and is activated when any input state differs from its corresponding input port register state. It is used to indicate to the microcontroller that an input state has changed and the device needs to be interrogated without the microcontroller continuously polling the input register via the I²C-bus.

The internal Power-On Reset (POR) initializes the I/Os as inputs with a weak internal pull-up 100 µA current source.

Design

Software Design

Sollen Textzeichen auf dem Display ausgegeben werden, so muss man sich außer mit der Frage des Hardwareanschlusses des Moduls natürlich auch mit dem Softwaredesign befassen. Dies beinhaltet:

- 1) I²C Programmierung des Portexpanders
- 2) Steuerung des HD44750 Display Controllers

Das Datenblatt des Display Controllers verrät, dass die Initialisierung und die Steuerung des Displays tiefe Kenntnisse der Funktionsschnittstelle voraussetzt. Diese komplexe Funktionalität muss über ein API vom Anwender ferngehalten werden (Kapselung).

I²C Programmierung

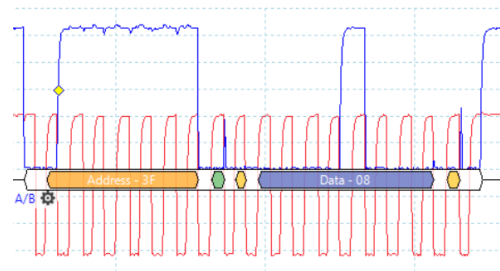
Hierbei geht es darum, eines der beiden I²C- Peripherals des MSP430FR2355 so zu konfigurieren, dass die Übertragung der 8-Bit Steuerwörter vom MC in den Portexpander per I²C Protokoll erfolgen kann.

Gewählt wird hier eUSCI_B0; die der SDA und SCL Leitung zugeordneten DIO-Pins können dem datasheet des MSP430FR2355 entnommen werden.

Als Vorlage für die Programmierung dient das TI Driverlib Sample Programm **eusci_b_i2c_ex4_masterTxSingle**.

Jedes der 8-Bit breiten Kommandos wird als einzelnes Byte versendet, wie nachfolgend dargestellt:

- Startkondition
- Übertragung der Slave Adresse
- Übertragung des Datenbytes
- Stoppkondition



Um die geplanten API Befehle zur Ansteuerung des Displays auszuführen, ist meistens eine Sequenz von Bytes vom MC an das Display zu übertragen. Die API Befehle werden als synchrone Befehle ausgeführt, d.h. ein API-Befehl kehrt erst dann zurück, wenn alle im Rahmen des Befehls an das Display zu übertragende Bytes tatsächlich auf das Display übertragen wurden. Hierzu muss der MC die Steuerbytes in den I²C Sendepuffer schreiben und warten, bis die Übertragung zum Display erfolgt ist, d.h. vom Portexpander über den I²C-Bus mit einem ACKN quittiert wird. Das Warten kann interruptgesteuert erfolgen oder indem die API Funktion den Zustand der I²C state machine des MSP430 I²C Peripherals abfragt (polling).

Nach der Übertragung eines Bytes per I²C, muss dem Display Zeit gegeben werden, um den Befehl auszuführen, bevor der nächste Befehl gesendet werden kann. Es besteht die Möglichkeit hierzu den BUSY-state des Displays abzufragen. Alternativ kann auch einfach für eine bestimmte Zeit gewartet werden, bis mit der Übertragung des nächsten Befehls begonnen wird. Angaben, wie lange bei welchem Befehl gewartet werden muss, finden sich im HITACHI Datenblatt.

Diese Wartezeit soll der Prozessor im Sleep Mode verbringen. Hierzu wird ein Sleep-Timer aufgesetzt, für den eine Wartezeit in ms vorgegeben werden kann. Eine Warteperiode kann über den Befehl `sleep(ms)`; gestartet werden. Der Befehl kehrt erst dann zurück, wenn die vorgegebene Warteperiode abgelaufen ist.

API-Design

Für die API werden folgende Befehle geplant:

- `lcd1602_init()` – Initialisierung des Displays

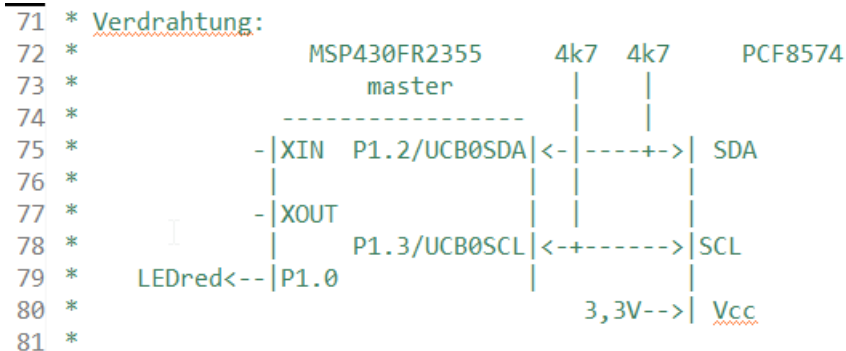


- lcd1602_write() – Übertragung einer anzuzeigenden Textzeile
- lcd1602_clear() – Löschen der Anzeige des Displays
- lcd1602_backlight() – Aktivierung des Backlights

Das API lässt sich selbstverständlich je nach Bedarf erweitern.

Hardware Design

Das Hardware Design ist sehr einfach: Es genügt, die SCL und SDA Leitungen mit dem Display zu verbinden und die Versorgungsspannung (3,3V) zu übertragen.



Für den I²C-Bus werden die bereits auf dem Portexpander Modul vorhandenen Pullup Widerstände verwendet.

Software Entwicklungsplan und Umsetzung

Wichtig: Immer nach geeigneten Sample Programmen suchen, bei denen die Grundeinstellungen schon getroffen wurden.

Alle hier beschriebenen Iterationen sind auf dem Gitlab Server der Hochschule abgelegt:

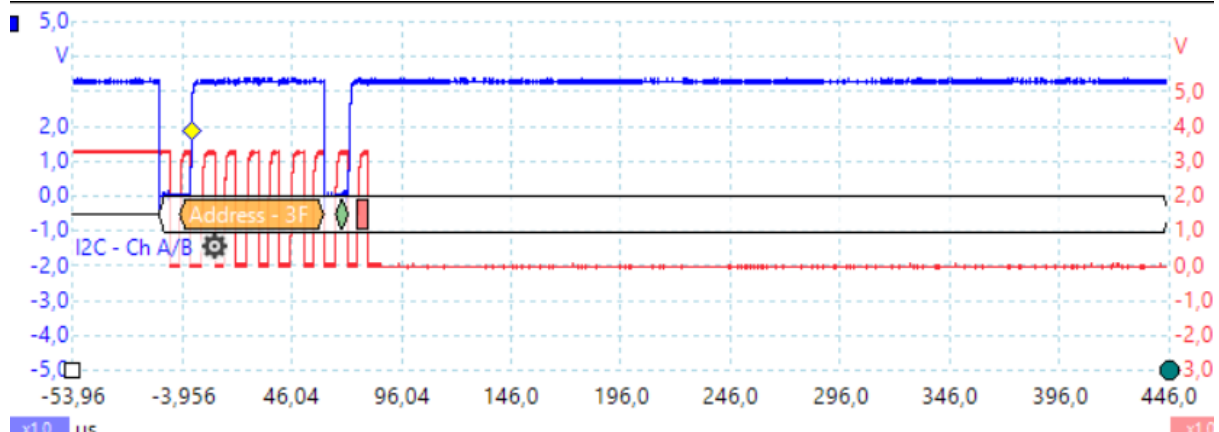
Iteration 1 → Version 0.1

Programm, welches über den REFO einen 1 MHz SMCLK erzeugt

Iteration 2 → Version 0.2

Programm, welches den I²C Bus mit eUSCI_B0 in Betrieb nimmt. Es soll die Slave-Adresse des PCF8574 ausgegeben werden, das Display wird jedoch noch nicht angeschlossen. Da in diesem Fall auf die Ausgabe der Adresse des Slaves keine Antwort kommt, kann davon ausgegangen werden, dass die interne protocol state machine des MSP430FR2355 zum Stehen kommt. Im Schirmbild ist zu sehen, dass auf das 9.Bit kein Ackn kommt.

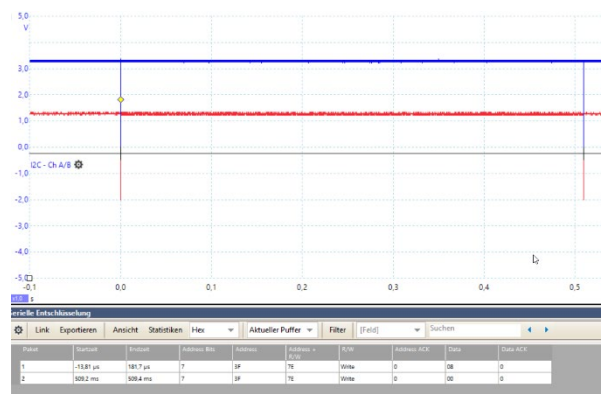
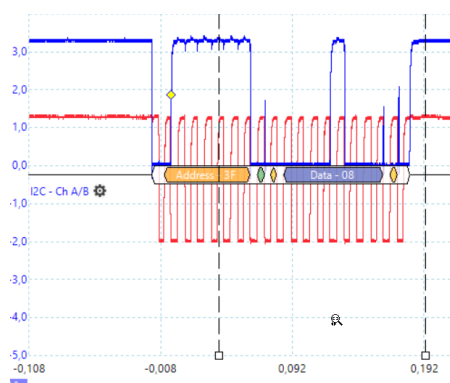
Ergebnis:



Iteration 3 → Version 0.3

Programm, welches über den Portexpander BIT3 setzt oder löscht und damit die Hintergrundbeleuchtung im ca. 500ms Takt an- oder ausschaltet. Dieses Programm beschränkt sich also darauf zu zeigen, dass die Ansteuerung des Portexpandermoduls funktioniert. Das Programm nimmt noch keinen Einfluss auf die Steuerung des LCD.

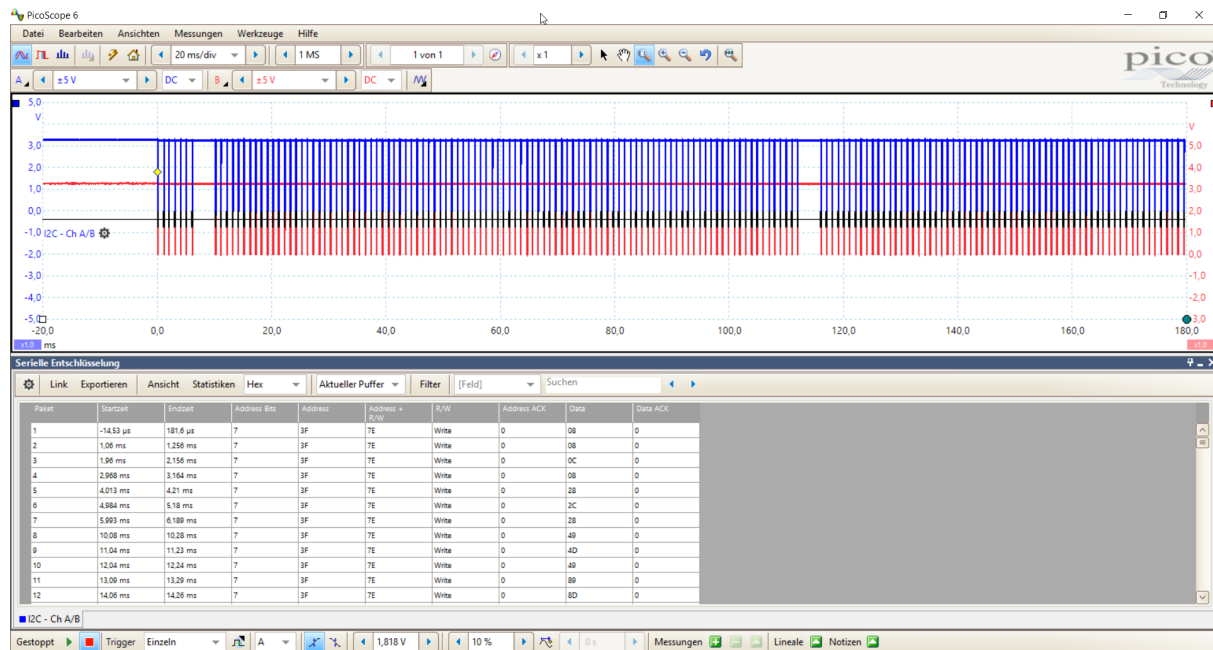
Ergebnis:



Iteration 4 → Version 0.4

Programm, welches auf den beiden Zeilen des Displays vordefinierte Strings ausgibt.

Ergebnis:



Sequentielle Übertragung des High-Nibble gefolgt vom Low Nibble → Übertragung eines Bytes. Die Übertragung der Nibble erfolgt durch das setzen des Enable Bits (Paket 2 und Paket 5).

Iteration 5 → Version 0.5

Hier nicht mehr umgesetzt ...

Umstellung des Busy Waiting bei der Übertragung der I²C Daten auf eine interruptgestützte Methode. Auch hierfür stellt die Driverlib geeignete Funktionen zur Verfügung.

Integration und Test

- Integration und Test wurden nach jedem Iterationsschritt durchgeführt.

Serienreifmachung

- Hier nicht belegt.
- Noch zu erledigen: doxygen-geeignete Kommentierung