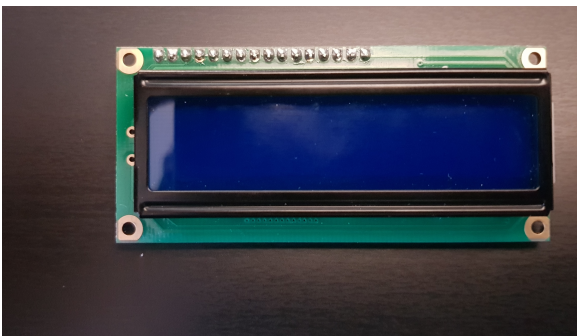


Praktikum 4: Digitale Anzeige der Ampelzustände

4 Praktikum Display

Ziel des Praktikums ist es, ein Display in den bisherigen Aufbau zu integrieren. Die folgenden Abbildungen (a) und (b) zeigen das Display einmal von vorn und einmal von hinten.



(a) Vorderseite des Displays



(b) Rückseite des Displays

Das Display wird in diesem Praktikum per GPIO angesteuert. Eine weitere Möglichkeit der Ansteuerung besteht darin, dass das Display per I2C angeschlossen wird. Den dazu nötigen I2C-Adapter haben Sie in der Kiste erhalten. Im Rahmen dieses Praktikums müssen Sie den Adapter aber nicht verwenden.

Als erstes muss das Display an den STM-Microcontroller angeschlossen werden. Anschließend wird ein Header geschrieben, der verschiedene Funktionen zur Bedienung des Displays beinhaltet. Wenn die Steuerung des Displays fertig gestellt ist sollen die Zustände der Ampelschaltung auf dem Display ausgegeben werden.

1.1 Aufbau der Schaltung

Am Display befinden sich 16 Pins. Diese werden zum Einen für die Stromversorgung des Displays und zum Anderen zur Steuerung des angezeigten Inhalts gebraucht. In der folgenden Tabelle sind die Anschlüsse 1-16 ausgeführt:

Pin	Symbol	Art	Funktion
1	VSS	Stromversorgung	GND
2	VCC	Stromversorgung	+ 5V
3	VEE	Stromversorgung	Kontrastspannung
4	RS	Datenleitung	0 = Befehl, 1 = Daten
5	R/W	Datenleitung	0 = schreiben, 1 = lesen
6	E	Datenleitung	Enable
7	D0	Datenleitung	Datenleitung 0
8	D1	Datenleitung	Datenleitung 1
9	D2	Datenleitung	Datenleitung 2
10	D3	Datenleitung	Datenleitung 3
11	D4	Datenleitung	Datenleitung 4
12	D5	Datenleitung	Datenleitung 5
13	D6	Datenleitung	Datenleitung 6
14	D7	Datenleitung	Datenleitung 7
15	A	Stromversorgung	Hintergrundbeleuchtung Anode
16	K	Stromversorgung	Hintergrundbeleuchtung Kathode

Füllen Sie die oben stehende Tabelle aus und erstellen Sie eine Schaltskizze des Aufbaus. Benutzen Sie zur Stromversorgung des Displays und der Hintergrundbeleuchtung die 5V Ausgänge des Mikrocontrollers. Bitte benutzen Sie für die Datenübertragung die Pins PD0 - PD7. Für RS verwenden Sie bitte PD12 und für E Pin PD14. Dies dient dazu, dass wir Ihre Lösung besser kontrollieren können. Den Pin R/W verbinden Sie bitte mit GND. Damit wird sichergestellt, dass das Display keine Signale mit 5V Pegeln an den Controller sendet. Dies schränkt den Funktionsumfang zwar ein, dient aber dem Schutz des Mikrocontrollers. Für die Hintergrundbeleuchtung und die Kontrasteinstellung benutzen Sie bitte die Drehpotentiometer. Informieren Sie sich, wie diese richtig eingesetzt werden! Bringen Sie dazu in Erfahrung, welche Funktion die Pins der Drehpotentiometer haben und welcher Widerstand sich eignet. Finden Sie im Betrieb heraus, welche Einstellung der beiden Drehpotentiometer einen guten Kontrast und eine ausreichende Hintergrundbeleuchtung liefert. Stellen Sie die Hintergrundbeleuchtung nicht zu stark ein, da dies die Komponenten belastet. Beachten Sie, dass das Display viel Platz braucht und auch die Drehpotentiometer auch Platz auf dem Breadboard brauchen.

1.2 Programmierung

Für die Ansteuerung verwenden Sie bitte Ihren GPIO-Treiber. Ihr Display-Treiber soll die Abstraktionsebene des GPIO-Treibers nutzen, um die GPIO-Ansteuerung zu realisieren. Sie sollen also in Ihrem Display-Treiber möglichst die Hardware nicht direkt, sondern über Ihren bereits beschriebenen Treiber ansteuern. Implementieren Sie dann die zur Ansteuerung nötigen Funktionen, indem Sie die Code-Vorlage in Ilias vervollständigen.

Die Vorlage enthält ebenfalls eine `main.c`, die Sie zum Testen Ihres Treibers einsetzen können. Führen Sie ein Code Review mit Ihrem Praktikumpartner durch und erarbeiten Sie eine gemeinsame Lösung.

Hinweis: Sollte das Display nicht richtig reagieren, kann es sein, dass das Display noch mit der Abarbeitung vorheriger Befehle beschäftigt ist. Testen Sie in diesem Falle bitte, ob sich das Problem durch Einfügen von Delays beheben lässt.

Das Display verfügt auch über ein Busy-Bit, mit dem überprüft werden kann, ob weitere Befehle an das Display gesendet werden können. Die Implementierung der Abfrage ist allerdings nicht trivial und erfordert den Anschluss des RW-Pins. Da der Controller hier bei falscher Implementierung beschädigt werden kann, ist dies nicht Teil des Praktikums. Wenn Sie dennoch als optionale Aufgabe versuchen möchten, das zu implementieren, sprechen Sie sich bitte unbedingt vorher mit Herrn Dey ab.

1.3 Ampelschaltung mit Anbindung an das Display

Binden Sie das Display in Ihre Ampelschaltung mit Interrupt-Funktion aus dem vorherhigen Praktikum ein. Erweitern Sie ihr Programm, so dass die aktuelle Ampelphase auf dem Display ausgegeben wird (z.B. ROT:1 GELB:1 GRUEN:0). Laden Sie das vollständige Programm inklusive der Treiberdateien als Gruppe in dem Abgabe-Ordner hoch.

Als optionale Aufgabe können Sie die Delays durch Timer, die Sie in der Vorlesung kennengelernt haben ersetzen. Ebenso können Sie als Zusatzaufgabe die Anzeige auf dem Display um einen Countdown für die jeweilige Ampelphase erweitern.

ABGABE: 21.05.2023