Aufgabenblatt 05 Termine: KW 23

Gruppe	
Name(n)	Matrikelnummer(n)

5 Elektrische Ansteuerung eines Displays

Flüssigkristalldisplays (LCDs) sind ein häufig verwendetes Ausgabegerät für eingebettete Systeme. Entsprechend des Bedarfes werden einfache Zeichen-Displays oder auch grafische Displays eingesetzt. Für die späteren Aufgaben der Übungen zu ES werden Sie ein grafisches 1.8" TFT-LCD (Thin-film-transistor liquid-crystal display) verwenden, um darauf grundlegende Funktionalitäten zur visuellen Ausgabe zu implementieren.

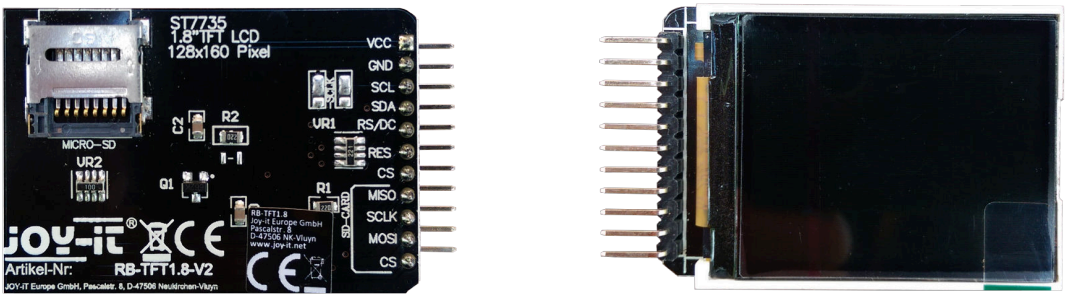

Abbildung 1: 1.8" TFT-Display mit zusätzlichem SD-Card-Slot.

Die Schnittstelle dieses Displays wird durch den Displaycontroller (Sitronix ST 7735R) bereitgestellt. Setzen Sie sich mit dem Datenblatt des Displaycontrollers auseinander. Der Displaycontroller verfügt über eine parallele, eine serielle 3-Draht und eine serielle 4-Draht Schnittstelle. Über die äußere Beschaltung des Displaycontrollers auf dem Breakout-Board des in den ES Übungen verwendeten Displays (siehe Abb. 1) ist die serielle 4-Draht Schnittstelle hardwareseitig fest konfiguriert. Abb. 2 zeigt das vereinfachte Timing-Diagramm dieser Schnitstelle. Weshalb erscheint in diesem Diagramm das Signal `SDA` zweimal; einmal als `SDA` und einmal als `SDA (DOUT)`?

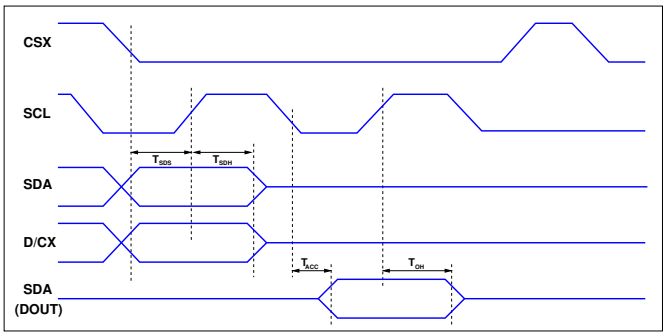

Abbildung 2: Vereinfachtes Timing-Diagramm der 4-Draht Schnittstelle des ST7735R;
Leitungen: `CSX` – Chip-Select, `SCL` – Serial Clock Line (Taktsignal),
`SDA` – Serial Data Line (Datenleitung), `D/CX` – Data/Command;
Timingangaben: T_{SDS} – Data setup time, T_{SDH} – Data hold time,
T_{ACC} – Access time, T_{OH} – Output disable time
(vgl. Datenblatt ST7735R, S.25)

Aufgabe 5.1 Diskussion eines Anschlussbeispiels

Diskutieren Sie das in Abb. 3 skizzierte Beispiel aus elektrischer Sicht zum Anschluss des Displays an einen Arduino. Für erste Überlegungen gehen Sie davon aus, dass für die Anschlüsse des Displays entsprechend Abb. 2 seitens des Arduinos normale digitale I/O-Pins verwendet werden.

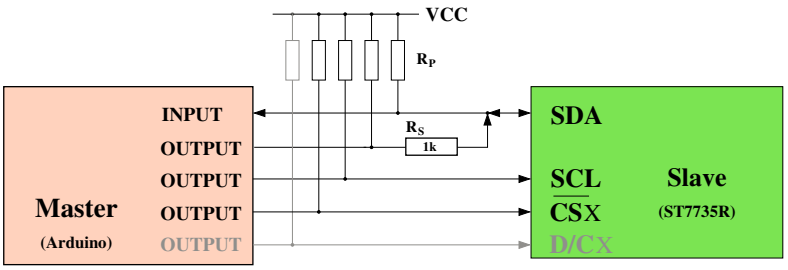

Abbildung 3: Anschlussvorschlag des Displays mit ST7735R-Controller

- Wäre diese Beschaltung grundsätzlich möglich und sinnvoll?
- Welche Bedeutung hätte dann der Widerstand R_S?
- Könnte er auch fortgelassen werden?
- Was können Sie über die Dimensionierung von R_S aussagen?
- Welche Bedeutung haben die Pullup-Widerstande R_P?
- Können/sollten Pullup-Widerstande fortgelassen werden?
  Und falls ja, welche und unter welchen Voraussetzungen?

Die in engebetteten Systemen eingesetzten Prozessoren stellen i. d. R. verschiedene serielle Schnittstellen, die für die Ansteuerung von Sensoren und Aktuatoren üblicherweise benötigt werden, bereits zur Verfügung. So stellen auch die Prozessoren der Arduino-Boards folgende **serielle Protokolle** bereit:

- **UART**: **U**niversal **A**synchronous **R**eceiver/**T**ransmitter
  Beispielsweise **RS-232**: Ein Kommunikationspartner pro Schnittstelle, asynchroner Datentransfer
- **I²C**: **I**nter-**I**ntegrated **C**ircuit
  Serieller Bus, (multi-)**Master/Slave** Kommunikation, synchroner Datentransfer
- **SPI**: **S**erial **P**eripheral **I**nterface
  Serieller Bus, **Master/Slave** Kommunikation, synchroner Datentransfer

Aufgabe 5.2 Serielle Busprotokolle

Machen Sie sich mit den o. g. seriellen Schnittstellen und ihren (Bus)Protokollen vertraut. Die referenzierten Wikepedia-Seiten bieten schon einen guten Überblick.
Lässt sich eine dieser Schnittstellen zur Ansteuerung des Displays über die serielle 4-Draht Schnittstelle des ST 7735R-Displaycontrollers (siehe Abb. 2) verwenden? Sie werden feststellen, dass keine dieser Schnittstellen auf den ersten Blick mit der seriellen 4-Draht Schnittstelle des ST 7735R-Displaycontrollers kompatibel ist. Berücksichtigen Sie bei Ihrer Einschätzung nicht nur die elektrischen Eigenschaften der Hardwareverbindungen, sondern hier auch insbesondere die Protokolle der Schnittstellen.
Welche dieser Schnittstellen halten sie ggf. für geeignet, das in den ES-Übungen verwendete Display anzusteuern. Begründen Sie Ihre Entscheidung.

Aufgabe 5.3 Befehlssatz des ST 7735-Controllers

Um mit dem TFT-Display arbeiten zu können, ist es nicht unbedingt erforderlich, den gesamten Befehlssatz im Detail zu kennen (siehe dazu Datenblatt ST 7735-Controller). Für den komfortablen Umgang mit dem Display werden Sie sich in den späteren Aufgaben eine eigene kleine Bibliothek an Funktionen schreiben, um von der Kommunikation mit dem Displaycontroller zu abstrahieren. Für diese Funktionen werden Sie nur eine begrenzte Anzahl von von Befehlen des Controllers benötigen.
Bevor Sie beispielsweise das Display tatsächlich benutzen können, sind die folgenden Befehle (*LCD-COMMAND*) als Initialisierungssequenz an das Display zu übertragen. Dabei sind abhängig vom jeweiligen Befehl noch die erforderlichen Parameter - dann als Daten (*LCD-DATA*) - zu übertragen.
Die angegebene Initialisierungssequenz initialisiert nur die Register des Displaycontrollers, die entweder durch einen HW- bzw. SW-RESET nicht initialisiert wurden, oder die abweichend von geladenen Default-Werten gesetzt werden sollten:

1. `0x01` : SWRESET (software reset)
2. `0x11` : SLPOUT (Sleep out & booster on)
3. `0x36` : MADCTL (Memory Data Access Control)
   `0xC8` (Row-, Column-, Exchange-, Refresh-, RGB-, Refresh-Order)
4. `0x3A` : COLMOD (RGB-format, 12/16/18bit)
   `0x55` (16-Bit/pixel)
5. `0x2A` : CASET (Column adress set)
   `0x00` (first column, high order byte)
   `0x02` (first column, low order byte)
   `0x00` (last column, high order byte)
   `0x81` (last column, low order byte)
6. `0x2B` : RASET (Row adress set
   `0x00` (first row, high order byte)
   `0x01` (first row, low order byte)
   `0x00` (last row, high order byte)
   `0xA0` (last row, low order byte)
7. `0x13` : NORON (Partial off (Normal))
8. `0x29` : DISPON(Display on)

Die `grau` hinterlegten Werte der Tabelle sind entweder die Hex-Codes des Befehls (LCD-COMMAND) oder eingerückt, die Hex-Werte der/des Parameters (LCD-DATA).

Machen Sie sich vorab mit diesen Befehlen vertraut und ermitteln Sie den Befehl, mit dem Pixelwerte nun tatsächlich in den Speicher des Displaycontrollers übertragen werden.