



Aufgabenblatt 6 Termine: 23.06./26.06.

Gruppe	
Name(n)	Matrikelnummer(n)

Ein häufiger Anwendungsfall eingebetteter Systeme ist die Datenaufzeichnung. Die Langzeittemperaturaufzeichnung ist ein gutes Beispiel eines solchen Anwendungsfalls, der stark vom Einsatz kostengünstiger und wartungsarmer (bzw. leicht austauschbarer) eingebetteter Systeme profitiert. In Abhängigkeit von den Performanzanforderungen und den Anforderungen an die Modularität, erfolgt häufig die Anbindung von externem Speicher (im Gegensatz zum "on-board" Speicher). Bei der Bearbeitung der nachfolgenden Aufgaben werden Sie ein System realisieren, das eine **microSD Speicherkarte** als externes Speichermedium anbindet und die auf der Speicherkarte abgelegten Daten liest, um diese auf dem LC-Display (Aufgabenblatt 5) zu präsentieren.

Aufgabe 6.1

Für die Lösung dieser Aufgabe benötigen Sie ein Wireless SD Shield (arduino.cc/en/pmwiki.php?n=Main/ArduinoWirelessShield) zuzüglich einer vorbereiteten microSD Speicherkarte.

Achtung: Da die microSD Speicherkarte (genau wie das LC-Display) über SPI kommuniziert und das Wireless SD Shield bei direkter Verbindung mit dem Arduino Due Board den SPI-Block unzugänglich macht, wird empfohlen die notwendigen Signale über das Steckbrett durchzuschleifen.

Beachten Sie bitte folgende Hinweise zur Verdrahtung:

- Verbinden Sie VCC 3.3V, VCC 5V und GND mit den entsprechenden Anschlüssen des Wireless SD Shields.
- Verbinden Sie die MISO, MOSI und SCK Anschlüsse über das Steckbrett mit den entsprechenden Anschlüssen des SPI Blocks.
- Das Wireless SD Shield sieht **Pin 4** exklusiv als Slave-Select Anschluss vor (Aufdruck SD_CS). Verbinden Sie diesen Anschluss mit einem noch freien Slave-Select Anschluss des Arduino Due (4, 10 oder 52). **Wichtig:** Übergeben Sie die Nummer des von Ihnen gewählte Anschlusses am Arduino Due als Argument an die Initialisierungsfunktion `SD.begin(pin)`.

Machen Sie sich zunächst mit dem Funktionsumfang der Arduino SD Bibliothek (`#include <SD.h>`) vertraut arduino.cc/en/Reference/SD. Schauen Sie sich insbesondere die folgenden Funktionen an:

* <code>SD.begin()</code>	arduino.cc/en/Reference/SDbegin
* <code>SD.exists()</code>	arduino.cc/en/Reference/SDexists
* <code>SD.open()</code>	arduino.cc/en/Reference/SDopen
* <code>File.available()</code>	arduino.cc/en/Reference/FileAvailable
* <code>File.read()</code>	arduino.cc/en/Reference/FileRead
* <code>File.close()</code>	arduino.cc/en/Reference/FileClose

Auf der Ihnen ausgehändigten microSD Speicherkarte befinden sich auf der obersten Ebene des Dateisystems Textdateien mit der Endung `.txt`:

- `text1.txt`
- `text2.txt`

sowie Bilddateien mit der Endung `.img`:

- `tams.img`
- `smile1.img`
- `smile2.img`
- `smile3.img`

Entwerfen Sie einen kleinen Befehlssatz für die Eingabezeile des seriellen Monitors, der Ihnen ermöglicht 1. die Existenz einer Datei anzufragen und 2. die existierende Datei auf dem LC-Display anzeigen zu lassen. Beachten Sie bitte folgende Hinweise bezüglich der Dateikodierung:

- Die Textdateien enthalten nur eine Textzeile, abgeschlossen durch ein `newline`-Zeichen (`\n`). Beispiel: `This is some text.\n`
- Die Bilddateien enthalten zwei Zeilen ASCII-kodierter Daten. Auch hier sind beide durch ein `newline`-Zeichen (`\n`) abgeschlossen.
- Die erste Zeile enthält die Dimensionen des Bildes:
 - Beispiel: `10,10\n`
- Die zweite Zeile enthält die Definition des jeweiligen Pixels (**zeilenweise angeordnet**) durch eine `0` oder eine `1`:
 - Beispiel: `0,0,0,0,0,1,1,1,1,1, ... ,1,1,1,1,1\n`

Weitere Hinweise:

- Bitte positionieren Sie die Bilder bei der Ausgabe **horizontal und vertikal zentriert** in der Anzeige des LC-Displays.
- Verwenden Sie den aus Blatt 6 bekannten ASCII-Datensatz zur Darstellung des Inhalts der Textdateien. Sorgen Sie für den entsprechenden Umbruch des Textes.

- Sollte eine Textdatei mehr Zeichen haben als auf dem LC-Display im 6x8-Format dargestellt werden können, so sollten Sie dieses entsprechend auf dem LC-Display mitteilen (Beispiel: Too many characters).
- *Optional* können Sie auch ein fortwährendes Scrollen implementieren, um den Text vollständig anzeigen zu können.