

Abgabeübung ■ 1

Abzugeben sind nach Bearbeitung der Aufgaben:

- Dokumentation: Details zum Softwareentwurf (Ablaufpläne, Pseudo-Code, etc.),
Generelle Beschreibung der Programmteile, Projektplanung
- Programmdateien (c.-files)

Hierbei soll die Dokumentation nur eine Datei (z.B. *.pdf) umfassen, die auch den Code der entwickelten C-Funktionen und des Hauptprogramms enthält. Die Datei muss als Dateinamen den Namen des jeweiligen Studenten enthalten.

Die komplette Lösung der Abgabeübung (Dokumentation und Programme) sollen in Moodle als ZIP-file hochgeladen werden. Jeder Student erhält eine Eingangsbestätigung im System.

Die Abgabeübung wird benotet, wobei sowohl die Funktionalität der Programme als auch die Qualität der Dokumentation (Ausarbeitung und Kommentare im Text) in die Notenfindung einbezogen wird.

Hierbei gilt die folgende Verteilung:

Funktionalität (60%)	Dokumentation (30%)
→ Aufgabe 1 (5%)	→ Beschreibungen
→ Aufgabe 2 (5%)	→ Analyse
→ Aufgabe 3 (15%)	→ Form und Stil
→ Aufgabe 4 (15%)	Programmdesign (10%)
→ Aufgabe 5 (20%)	→ Modularität
	→ Effektivität
	→ Form, Stil, Struktur

Werden weniger als 40% der erreichbaren Punkte erzielt, wird die Abgabeübung nicht anerkannt und es ist eine Ersatzaufgabe zu lösen.

Programm und Programmdokumentation sind selbstständig zu erstellen. Sind Lösungen oder Teile von Lösungen kopiert worden, wird die Übung nicht anerkannt.

Es sind ca. zwei Wochen Bearbeitungszeit für die Abgabeübung angesetzt.

Der Bearbeitungszeitraum beginnt am **21. Dezember 2023** und endet am **14. Januar 2024** um 23:59 Uhr.

Bei verspäteter Abgabe werden Punktabzüge eingeführt, die durch einen Multiplikator angegeben werden:

Bis zu zwei Tage zu spät	Erzielte Punkte werden multipliziert mit 0,9
Bis zu vier Tage zu spät	Erzielte Punkte werden multipliziert mit 0,8
Bis zu sieben Tage zu spät	Erzielte Punkte werden multipliziert mit 0,7
Mehr als sieben Tage zu spät	Erzielte Punkte werden mit 0 multipliziert

1. Aufgabenstellung ■ Entwicklung eines Datenloggers mit Trendanzeige

Das AVR-Microcontrollersystem soll als einfacher Datalogger eingesetzt werden. Die Spannung des ersten Potentiometers (Poti_1) wird erfasst und als Balkendiagramm in der oberen Zeile des Character-LCD angezeigt. Der aktuell abgetastete Messwert wird jeweils rechts in die Darstellung eingefügt. Gleichzeitig wird der „älteste“ Messwert links aus dem Display herausgeschoben. Ein Beispiel für die entsprechende Anzeige ist in der folgenden Abbildung dargestellt.



Abbildung 1 – Darstellung des Datenloggers nach sieben Abtastungen

Nach dem Start der Aufzeichnung erscheinen zunächst Striche in der oberen Zeile, da noch keine Messwerte eingelesen wurden. Zusätzlich zu dem Balkendiagramm werden die aktuelle Spannung (obere Zeile) sowie die Abtastgeschwindigkeit und die Anzahl der bereits eingelesenen Werte angezeigt. Die Messwerte werden in einem internen Buffer, der eine Größe von 100 Werten hat, abgespeichert. Werden mehr als 100 Werte aufgezeichnet, werden die ältesten Werte im Buffer überschrieben. Der Start der Aufzeichnung soll mit der Taste ‚A‘, das Beenden der Aufzeichnung mit der Taste ‚B‘ möglich sein. Die Wahl der Abtastrate erfolgt durch Eingabe einer Ziffer, ein Ändern der Abtastrate bei laufender Aufzeichnung ist nicht möglich. Nach dem Beenden der Aufzeichnung können die aufgezeichneten Messwerte dargestellt werden. Der jeweils dargestellte Ausschnitt wird mit dem Potentiometer Poti_2 gewählt.

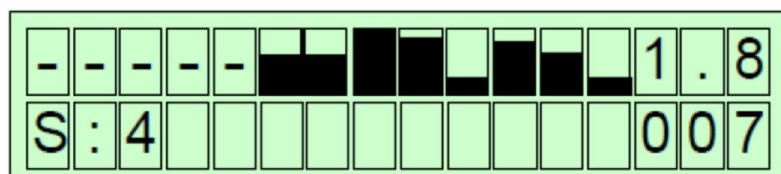


Abbildung 2 – Darstellung des Datenloggers nach acht Abtastungen

Die hier genannte Funktionalität stellt die Basis Funktion des Systems dar. Es ist Studierenden hierbei möglich das Projekt durch sinnvolle Ergänzungen dergestalt zu erweitern, dass entweder zusätzliche Features implementiert werden und/oder entsprechende Erweiterungen oder Verbesserungen bereits beschriebener Funktionen realisiert werden. Die Implementierung von Zusatzfunktion wird hierbei durch Extrapunkte bei der Bewertung belohnt, welche dementsprechend nicht erreichte Punkte in anderen Bewertungskategorien (z.B. im Bereich der Dokumentation und/oder der Programmqualität) ausgleichen.

Aufgaben ■ beschreibung

Gegenstand der Abgabeübung ist die Realisierung eines einfachen Datenloggers mit Trendanzeige. Hierzu bedarf es zunächst der Implementierung von einzelnen Programmmodulen zur Realisierung von Teilaufgaben des Gesamtprojektes. Um hierbei eine möglichst systematische Vorgehensweise zu verfolgen seien im Folgenden einzelne Aufgaben definiert, welche jeweils die Implementierung von entsprechenden Teilmodulen beinhalten. Diese seien dann abschließen zu einem Gesamtprojekt zu vereinen.

Aufgabe 1 ■ Analog-Digital-Wandlung

Die wesentlichste Funktion des oben beschriebenen Datenloggers besteht neben der eigentlichen Aufzeichnung von Datenwerten in einem Speicher in erster Linie in der Realisierung einer Funktion zur Analog-Digital-Wandlung der Spannungen an den beiden Potentiometern auf dem AVR-Microcontrollersystem. Hierzu soll eine Funktion entwickelt werden, welche den internen Analog-Digital-Wandler (ADC) des verbauten Microcontrollers nutzt um die anliegenden Spannungen an den entsprechenden Eingangspins einzulesen. Die Funktion soll hierbei möglichst modular und zur generellen Verwendung formuliert werden, so dass als Übergabeparameter (i). der einzulesende Kanal (0-7) und (ii). die entsprechenden Minimal- und Maximalwerte des Eingangssignals übergeben werden sollen. Letztere Information wird hierbei zur Skalierung des vom ADC definierten 10 Bit Wandlungsergebnisses benötigt. Die Funktion liefert hierbei das Ergebnis als Gleitkommazahl entsprechend der vorgegebenen Skalierung zurück. Die Funktion soll dergestalt implementiert werden, dass bei jedem Aufruf der Funktion eine einzelne Wandlung des vorgegebenen Kanals angestoßen wird.

Aufgabe 2 ■ Ausgabe des Analogwertes auf dem C-LCD

Um den von Kanal 0 (Poti_1) eingelesenen Wert anzuzeigen, bedarf es der Implementierung einer Funktion zur Darstellung des Wertes auf dem Character-LCD (C-LCD). Hierbei ist die entsprechende Position laut Abbildung 1 und die zugehörige Skalierung zu beachten. Die Funktion ist hierbei dergestalt auszuführen, dass dieser der anzuzeigende Wert, sowie die Position auf dem C-LCD, in Form der entsprechenden Adresse, übergeben werden kann. Es ist hierbei von einer Gleitkommazahl kleiner Zehn mit einer Stelle nach dem Komma auszugehen.

Aufgabe 3 ■ Realisierung des Speichers

Wie bereits in der Aufgabenstellung beschrieben, soll das System in der Lage sein einhundert einzelne Werte in einem Buffer zu speichern. Hierzu ist eine entsprechende Funktion zu implementieren, welche bei jedem Aufruf einen neuen Wert in das Buffer schreibt und auch dafür sorgt, dass bei Erreichen der maximalen Anzahl von Werten entsprechende Bufferelemente überschrieben werden. Diese Funktion ist dabei dergestalt auszuführen, dass Sie für beliebige Buffergrößen funktioniert.

Aufgabe 4 ■ Graphische Ausgabe des Trends

Wie in Abbildung 1 dargestellt sind jeweils eine Reihe von Messwerten in graphischer Form auf dem C-LCD darzustellen. Hierzu bedarf es zum einen der Definition der entsprechenden Zeichen zur Realisierung der Bargraphen und zum anderen der Implementierung der eigentlichen Ausgabe auf dem C-LCD. Für letzteres ist hierbei stets zu unterscheiden in welchem Modus sich das System derzeit befindet (Aufnahmemodus oder gestoppter Modus) und dann sind entweder stets die zuletzt aufgenommenen Spannungen graphisch darzustellen oder die entsprechende Position des Buffers auf Basis der eingelesenen Spannung von Poti_2.

Aufgabe 4.1 ■ Implementierung einer Funktion zur Erzeugung der Zeichen

Es ist zunächst eine Funktion zu implementieren, welche die acht notwendigen Zeichen zur Darstellung eines Bargraphen im CGRAM des C-LCDs erzeugt. Diese Initialisierungsfunktion kann dann stets zu Beginn des Programmes aufgerufen werden.

Aufgabe 4.2 ■ Ausgabe des Bargraphen

Zur Darstellung des Bargraphen auf dem C-LCD ist eine Funktion zu implementieren, welche die graphische Ausgabe der eingelesenen Spannungswerte auf dem Display je nach Modus richtig darstellt. Es ist hierbei außerdem zu beachten, dass eventuell noch nicht beschriebene Bufferzellen als □ dargestellt werden sollen.

Aufgabe 5 ■ Implementierung des Gesamtprogramms

Es ist abschließend auf Basis der zuvor implementierten Funktion nun das Gesamtprojekt auszuarbeiten, welches die vollständige Funktionalität abbildet. Hierbei sind die noch fehlenden Elemente (z.B. die Nutzung des Keypads, die variable Abtastgeschwindigkeit, etc.) noch zusätzlich zu implementieren. Zur Kontrolle der Implementierung steht eine HEX-Datei zur Verfügung, welche die Grundfunktionalität komplett abbildet und als Muster Verwendung finden kann.