



h_da

HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

fbi
FACHBEREICH INFORMATIK

MIKROPROZESSORPRAKTIKUM

WS2023

Termin 5

Programmierung für eingebettete Systeme: Pointer, Peripherie, USART

Name, Vorname	Matrikelnummer	Anmerkungen
Datum	Raster (z.B. Mi3x)	Testat/Datum

Legende: V: Vorbereitung, D: Durchführung, P: Protokoll/Dokumentation, T: Testat

Arbeitsverzeichnis:

Kopieren Sie sich aus dem „sftp://stxyyyy@user.v-shell.fbi.h-da.de/share/LabDisk/MI/“ den Ordner mpsDUEWS2023. Dort finden Sie zu jedem Termin vorgegebene Dateien.

Lernziele:

Die Kenntnis der Basistechnologie zur Implementierung einer Schnittstelle zwischen Anwendungsprogrammen und Betriebssystemen. Möglichkeiten der Ein- und Ausgabe von Zeichen über eine serielle Schnittstelle. Vorbereiten von Testszenerarien bei Zugriff auf ein Zielsystem im Labor.

Nehmen Sie die zur Verfügung gestellten Dateien in ein neues Projekt auf, testen, diskutieren, verstehen und dokumentieren Sie diese.

```
+-----+
| A -   Serial Device       : /dev/ttyUSB0
| B - Lockfile Location    : /var/run
| C -   Callin Program     :
| D -   Callout Program    :
| E -   Bps/Par/Bits       : 115200 8E1
| F - Hardware Flow Control : No
| G - Software Flow Control : No
| H -   RS485 Enable       : No
| I -   RS485 Rts On Send  : No
| J -   RS485 Rts After Send : No
| K -   RS485 Rx During Tx : No
| L -   RS485 Terminate Bus : No
| M -   RS485 Delay Rts Before: 0
| N -   RS485 Delay Rts After : 0
|
|   Change which setting?
+-----+
```

Erklären Sie welche Parameter in der Terminalemulation (z.B. minicom) einzustellen sind, damit eine Kommunikation über die serielle Schnittstelle funktionieren wird.

Die erzeugten Ausgaben von CR (Wagenrücklauf) und LF (Zeilenvorschub) sollten auf einem Terminal an der seriellen Schnittstelle zu sehen sein. Verwenden Sie in einer separaten Shell das Programm „*minicom*“ als Terminalersatz. Hinweis: „*minicom*“ muss auf dem zum Entwicklungssystem zugehörigen PC gestartet werden.

Aufgabe 1:

Bitte führen Sie die Initialisierung der USART-Schnittstelle auf dem Arduino Due durch. Verwenden Sie die Pins 12 (PA12) und 13 (PA13) des Parallel Input/Output Controller A (PIOA) sowie den Universal Synchronous Asynchronous Receiver Transmitter 1 (USART1). Stellen Sie sicher, dass die Kommunikationsschnittstelle korrekt eingerichtet wird, um zukünftige Datenübertragungen zu ermöglichen.

Wie kann die USART-Schnittstelle auf dem Arduino Due initialisiert werden?

Welche Pins werden für die Kommunikation verwendet, und welcher Controller ist damit verbunden?

Warum ist es wichtig, die Kommunikationsschnittstelle korrekt einzurichten?

Aufgabe 2:

Ihre Aufgabe besteht darin, ein einzelnes Zeichen über die USART-Schnittstelle zu übertragen. Verwenden Sie dazu das Transmit Holding Register (US_THR). Überwachen Sie anschließend mithilfe des Channel Status Registers (US_CSR), ob die Übertragung erfolgreich war.

Welches Register wird verwendet, um ein Zeichen über die USART-Schnittstelle zu übertragen?

Wie lässt sich überprüfen, ob die Übertragung erfolgreich war?

Warum ist das Channel Status Register (US_CSR) relevant?

Aufgabe 3:

Empfangen Sie ein Zeichen über die USART-Schnittstelle und verwenden Sie das Receive Holding Register (US_RHR) für diesen Zweck. Nutzen Sie auch das Channel Status Register (US_CSR), um den erfolgreichen Empfang zu bestätigen.

Welche Register werden genutzt, um ein empfangenes Zeichen zu speichern und zu überprüfen?

Welche Rolle spielt das Channel Status Register (US_CSR) beim Empfangsprozess?

Warum ist es wichtig, den erfolgreichen Empfang eines Zeichens zu bestätigen?

Aufgabe 4 (Optional):

Nutzen Sie den Peripheral DMA Controller (PDC), um mehrere Zeichen aus dem Speicher des Prozessors zu übertragen. Verwenden Sie das Transmit Pointer Register (US_TPR) und das Transmit Counter Register (US_TCR) für die Kontrolle dieser Übertragung.

Wie kann der Peripheral DMA Controller (PDC) für die Übertragung von Zeichen aus dem Prozessorspeicher verwendet werden?

Welche Register sind relevant, um die Datenübertragung mithilfe des PDC zu steuern?

Warum könnte die Verwendung des PDC für die Zeichenübertragung vorteilhaft sein?

Aufgabe 5 (Optional):

Setzen Sie den Peripheral DMA Controller (PDC) ein, um mehrere Zeichen in den Speicher des Prozessors zu empfangen. Verwenden Sie das Receive Pointer Register (US_RPR) und das Receive Counter Register (US_RCR), um die empfangenen Zeichen effizient zu verarbeiten.

Welche Vorteile bietet der Peripheral DMA Controller (PDC) für den Empfang von Zeichen im Vergleich zur herkömmlichen Methode?

Welche Register sind notwendig, um den Empfang von Zeichen mithilfe des PDC zu kontrollieren?

Wie kann der korrekte Empfang von Zeichen mithilfe des PDC sichergestellt werden?

Aufgabe 6 (Optional):

Sie sollten bereits jetzt darüber nachdenken, wie Sie Ihre Messwerte (sowohl Integer als auch Gleitkommazahlen) über die USART-Schnittstelle übertragen können? Möglicherweise könnten Sie dies bereits im Vorfeld für den Termin 6 vorbereiten.

Aufgabe 7:

Erstellen Sie zu diesem Termin ein Protokoll mit den Lösungen zu den Aufgaben und Ihren Erkenntnissen. Das Protokoll sollen Sie zu den nächsten Terminen vorlegen können. Denken Sie daran, dass zum letzten (sechsten Termin) auch eine Dokumentation zu Ihrem Projekt (Funktions- und Programmbeschreibungen, Installationsanleitung, Inbetriebnahme, Benutzerhandbuch) vorgelegt werden soll.