

### Aufgabe 3

**Lernziele:** Machen Sie sich mit dem Prinzip der seriellen, asynchronen Datenübertragung mittels der UART-Schnittstelle vertraut. In dieser Aufgabe sollen Sie lernen

- a) wie man die UART-Schnittstelle des MSP430 konfiguriert,
- b) wie man über die UART-Schnittstelle Daten empfängt und versendet,
- c) wie man komplexe Abläufe in einer ereignisgesteuerten Applikation in C programmiert.

Die in den Aufgaben 1 und 2 implementierten Funktionalitäten (d.h. die Ausgabe eines Blinkmusters, das Entprellen von Tasten sowie die Ansteuerung der vierstelligen Siebensegmentanzeige) sind auch in der 3. Aufgabe zu behalten und ggf. zu erweitern. Für die 3. Aufgabe stehen keine C- und H-Dateien zur Verfügung. Basierend auf dem Wissen aus der Vorlesung und den bisherigen Übungsstunden kann man selbständig und frei entscheiden, wie die 3. Aufgabe am besten zu lösen ist, insbesondere wie die zu implementierenden Funktionalitäten zwischen einer Interrupt-Service-Routine und der main-Funktion verteilt werden können.

### Aufgabenstellung

Im Rahmen der 3. Aufgabe sollen Daten zwischen dem Eval-Board mit dem MSP430 und einem Terminalprogramm (z.B. putty) auf einem herkömmlichen PC über einen bidirektionalen Kommunikationskanal ausgetauscht werden. Der bidirektionale Datenaustausch besteht aus zwei weitgehend unabhängigen Abläufen. Der eine Ablauf wird durch das Terminalprogramm auf einem PC initiiert und führt auf dem Eval-Board zum Empfang einer vierstelligen Dezimalzahl über die UART-Schnittstelle, zum Weiterleiten dieser Zahl an den Handler zur Ansteuerung der vierstelligen Siebensegmentanzeige und anschließend zur Rücksendung dieser Zahl über die UART-Schnittstelle an das Terminalprogramm. Der zweite Ablauf wird durch das Eval-Board initiiert. Und zwar konnte man in der 2. Aufgabe mit Hilfe der vier Buttons auf der Erweiterungsplatine die einzelnen Positionen der vierstelligen Siebensegmentanzeige inkrementieren oder dekrementieren. Diese Funktionalität soll in der 3. Aufgabe derart erweitert werden, dass solche Änderungen nicht nur auf der vierstelligen Siebensegmentanzeige angezeigt werden, sondern auch über die UART-Schnittstelle an das Terminalprogramm auf einem PC weitergeleitet werden.

Das Format eines Datenpakets, das vom Eval-Board empfangen wird, besteht aus insgesamt fünf Bytes, die ersten vier Bytes sind ASCII-Zeichen aus dem Bereich '0' bis '9', das fünfte Byte ist das Abschlusszeichen '\r'. Nur Datenpakete, die in solchem Format empfangen worden ist, werden als korrekte Dezimalzahlen interpretiert und weiter verarbeitet. Jede Abweichung von diesem Format (z.B. in der Länge, oder Zusammensetzung der ASCII-Zeichen) ist als Fehler zu betrachten. Fehlerhafte Datenpakete werden immer verworfen.

Das Format eines Datenpakets, das vom Eval-Board versendet wird, besteht aus insgesamt sechs Bytes, die ersten vier Bytes sind ASCII-Zeichen '0' bis '9', das fünfte und sechste Byte sind zwei Abschlusszeichen '\r' und '\n'. Das Versenden eines Datenpakets über die UART-Schnittstelle darf andere Abläufe im System nicht blockieren und muss deshalb mit Hilfe einer Zustandsmaschine implementiert werden.

Die UART-Schnittstelle des MSP430 ist so zu initialisieren, dass Daten byteweise gesendet und empfangen werden können, und zwar mit einem Stoppbit, mit gerader Parität und bei einer Baudrate von 14400 bit/s. Während der Übertragung wird das LSB zuerst gesendet bzw. empfangen.

Derjenige Taster auf der Hauptplatine, mit dem man in den Aufgaben 1 und 2 zwischen verschiedenen Blinkmustern umschalten konnte, hat in der 3. Aufgabe keine Funktion mehr.

Da während der Datenübertragung verschiedene Fehler auftreten können, muss die Applikation in der Lage sein, auf diese Fehler korrekt zu reagieren und solche Fehler auch optisch anzuzeigen. Die folgende Tabelle gibt einen Überblick über mögliche Fehler und mit welchen Blinksequenzen sie anzuzeigen sind.

Priorität	Fehler/Zustand	Blinkmuster
1 (am höchsten)	Break detect	schnelles Gleichtaktlicht
2	Framing, Overrun, Parity error	3 x Blinken
3	Character error	2 x Blinken
4	Buffer error	1 x Blinken
5	Byte empfangen	unterbrochenes Licht
6	kein Fehler	unterbrochenes Licht

Der Zustand *Break detect* liegt vor, wenn die Datenverbindung zwischen dem Eval-Board und dem PC getrennt ist und das UART-Modul des MSP430 diesen Zustand erkannt hat. Der *Character error* liegt vor, wenn ein Byte empfangen ist, das keine Dezimalziffer im ASCII-Code ist. Ein *Buffer error* liegt vor, wenn die vierstellige Dezimalzahl zu wenig (<4) oder zu viel (>4) Ziffer enthält. Treten mehrere Fehler gleichzeitig auf, so wird nur derjenige Fehler mit der höchsten Priorität angezeigt. Ein Fehler wird so lange angezeigt, bis entweder ein höherpriorisierter Fehler auftritt, oder eine vierstellige Dezimalzahl fehlerfrei empfangen worden ist. Die Implementierung einer sinnvollen Fehlerbehandlung wird den Kursteilnehmern/innen überlassen.

Nach der Reset/Power-On-Phase ist die Funktion Inkrementieren aktiv, die vierstellige Dezimalzahl wird mit 0000 initialisiert und die vierstellige Siebensegmentanzeige zeigt den Wert 0000 an.

Beachten Sie bei der Implementierung folgende Konstellation: während die ISR mit dem Versenden einer Zeichenkette über die UART-Schnittstelle beschäftigt ist, können in dieser Zeit neue Aufforderungen (Events) zum Senden kommen. Dies ist beispielsweise dann der Fall, wenn die Tasten schnell getippt werden. Bei dieser Konstellation darf es zu keinem Zeitpunkt dazu kommen, dass die Zeichenkette, die über die UART-Schnittstelle ausgegeben wird, durch eine neue Zeichenkette überschrieben wird. Es darf auch nicht passieren, dass dabei zu einem Event-Fehler kommt.