# Konsolenprozess (Process C)

Funktion „***process c“***

Zunächst wird der Stackpointer von „processC“ auf den, für Prozess C definierten, Stackbereich gesetzt.

Es startet eine Endlosschleife. In dieser wartet eine weitere Schleife auf Eingaben auf PORT1. Ist eine Eingabe erfolgt setzt sich das RI0-Empfangsflag und die Schleife endet. Die Eingabe wird anschließend in Register 7 (R7) kopiert. Danach wird die Funktion zur weiteren Eingabebehandlung („handleSerial0Input“) aufgerufen. Das RI0-Empfangsbit wird zurückgesetzt und die Endlosschleife von „process c“ beginnt von vorne.

Zusätzlich wird in der Endlosschleife zyklisch der Watchdogtimer zurückgesetzt. Andernfalls würde der Watchdog nach einer bestimmten Anzahl von Durchläufen annehmen, das Programm befände sich in einer Verklemmung und alle Register zurücksetzen.

Funktion „***handleSerial0Input***“

Gemäß der Aufgabenstellung ist der weitere Programmverlauf abhängig von der Tastatureingabe

|  |  |
| --- | --- |
| Taste | Aktion |
| A | Prozess A starten |
| B | Prozess A beenden |
| C | Prozess B starten |
| Sonst | Keine Aktion |

Hierzu wird der bedingte Aufruf CJNE verwendet. CJNE springt an das angegebene Label wenn die Bedingung nicht erfüllt wird. War die Eingabe weder a,b oder c, so wird nur der Inhalt von Register 7 (R7) gelöscht und zurück in die Endlosschleife von „***process c***“ gesprungen. Erfolgt eine gültige Eingabe wird der jeweilige Datenzeiger kopiert. Durch das Bit „isNew“ und „isDel“ wird angegeben ob der Prozess gestartet oder beendet werden soll.

# Prozess A

Funktion von Prozess A ist es im 1-Sekunden Takt den Buchstaben ‚a‘ auf PORT1 zu schreiben. Prozess A beendet sich nicht selbstständig sondern wird durch Benutzereingaben beendet.

Funktion „processA“

Zunächst wird der Stackpointer von „processA“ auf den, für Prozess A definierten, Stackbereich gesetzt.

Anschließend wird das Register 5 (R5) auf 246 gesetzt. Dieser Wert ergab sich durch experimentieren, läuft der Timer 246-mal nacheinander vergehen exakt 1,007625 Sekunden. Eserfolgt der Aufruf der „mainLoop“ diese ruft abwechselnd die Funktion „printAToUART“ und „waitRoutine“ auf.

Funktionsabschnitt „printAToUART“

„a“ wird auf PORT1 geschrieben. Danach wird durch zyklisches Abfragen von S0CON auf Beendigung des Sendevorgangs geprüft. Ist dieser erfolgt wird das TI1-Empfangsbit zurückgesetzt und der Funktionsabschnitt endet.

Funktionsabschnitt „waitRoutine“

Der Timer 0 wird gestartet. Nun wird solange gewartet bis der Timer 0 Overflowed. Nach zurücksetzen des Watchdogs und des Timer 0 Overflow Bits wird der Wert von R5 dekrementiert. Erreicht dieser 0 ist ca. eine Sekunde vergangen und es endet die „waitRoutine“.

# Prozess B

Aufgabe von Prozess B ist die Zeichenfolge „54321“ auf PORT1 zu schreiben und sich anschließend selbstständig zu beenden.

Zunächst wird der Stackpointer von „processB“ auf den, für Prozess B definierten, Stackbereich gesetzt.

Es wird der Dezimal 53 (ASCII-Wert für 5) in Register 1 (R1) kopiert. In der folgenden Schleife wird nun der Wert von Register 1 auf PORT1 geschrieben und auf Beendigung des Sendevorgangs gewartet. Der Wert von Register 1 wird dekrementiert, das „Senden Bit“ wird zurückgesetzt und die Schleife wiederholt solange bis Register 1 den Wert 48 erreicht, was dem ASCII Wert von 0 entspricht. Dadurch lässt sich die Ausgabe „54321“ generieren (ASCII-Werte : 53 = „5“, 52 = „4“, 51 = „3“, 50 = „2“, 49 = „1“, 48 = „0“). Anschließend beendet sich der Prozess selbst, indem er dem Scheduler seine Startadresse übergibt und das „isNew“ Bit auf „isDel“ setzt.