

# Übungen zu LLWin

**START**

Die Aufgaben werden zum Großteil gemeinsam erarbeitet, da sie zum Kennenlernen der LLWin-Umgebung dienen. Die Ausführung der Programme setzt nur teilweise ein Interface und ein ft-Modell voraus, da oftmals Ein- und Ausgaben mit Hilfe des Terminal-Bausein gelöst werden. Die Hinweise zum Vorgehen sind beim Bearbeiten zu beachten.

## (1) Schalter 1

Wenn ein Taster (E17) gedrückt wird, soll auf der Anzeige des Terminals das Wort „gedrückt“ erscheinen, andernfalls „ungedrückt“.

*Vorgehensweise:* Zuerst werden wichtige Informationen im Text unterstrichen, dann ein Zustandsdiagramm gezeichnet. Dann entwickelt man ein informelles<sup>1</sup> Steuerflussdiagramm, das direkt in LLWin-Programm übersetzt werden kann. Wichtig ist es, einen Kommentartext mit Namen einzufügen. Speichern des Programms unter der Bezeichnung „UEB1\_1.MDL“ auf Diskette.

## (2) Schalter 2

Das Programm soll dem aus (1) entsprechen, aber eine Lampe an M1 ansteuern, wenn Taster E1 gedrückt wird.

*Vorgehensweise:* Zuerst sind die wichtigen Informationen im Text hervorzuheben. Da das Programm weitgehend (1) entspricht, ist ein neues Zustands- und Flussdiagramm nicht notwendig, sondern es genügt, den LLWin-Ablauf<sup>2</sup> anzupassen. Dateiname: UEB1\_2.MDL

## (3) Motor

Durch Schließen des Tasters an E1 soll ein Motor M1 nach rechts drehen, wird E2 gedrückt, so soll sich der Motor linksherum drehen, ist kein Taster gedrückt, so steht der Motor.

*Vorgehensweise:* Eine neue Aufgabenstellung verlangt wieder ein erneutes Analysieren des Textes und die Entwicklung eines neuen Zustandsdiagramms. Auch das Flussdiagramm muss neu gezeichnet werden. Anhand dieses Diagramms ist es sehr einfach, das LLWin-Programm zu schreiben (UEB1\_3.MDL)

## (4) Einfaches Codeschloss

Ein Codeschloss soll simuliert werden, indem auf dem Terminaldisplay „offen“ oder „geschlossen“ angezeigt wird. Um das Schloss zu öffnen, müssen die Schalter E17-E26 richtig ein- oder ausgeschaltet sein.

*Vorgehensweise:* Nachdem die wichtigen Informationen im Text farblich markiert sind, muss man die richtige Schalterstellung zur Öffnung des Schlosses festlegen. Ein Zustandsdiagramm erübrigt sich, da die Zustände nicht so kompliziert sind, aber ein schematisches Flussdiagramm ist notwendig. Dies wird dann in ein LLWin-Programm umgesetzt. (Speichern unter UEB1\_4.MDL)

<sup>1</sup> informell = ohne vorher vereinbarte Befehlsbezeichnungen, aber eindeutig beschrieben.

<sup>2</sup> oftmals auch als (Programm-)Faden (engl. Thread) bezeichnet

## (5) Komplexes Codeschloss

Das in (4) beschriebene Schloss ist natürlich nicht besonders sicher. Möglich wäre es, z.B. 10x die richtige Kombination von vier Zahlen (EA-ED) zu fordern, um das Schloss zu öffnen. Hier kann man seinen Gedanken freien Lauf lassen!

## (6) Zufallszahlen-Generator

Das Programm soll auf dem Display 1 des Terminals ständig wechselnde Zufallszahlen ausgeben.

*Hinweise zur Programmierung:* Eine Möglichkeit zur Problemlösung benutzt zwei parallel ausgeführte Programmfäden und einen WARTE-Baustein mit negativer Konstante. Dateiname: UEB1\_6.MDL

## (7) Reaktionstester

Sobald der Taster E17 geschlossen wurde, löscht das Programm den Meldungstext im Terminalbaustein, wartet eine zufällige Zeit (max. 10 sec), um dann im Terminal den Text „Jetzt!“ auszugeben. Dann wird die Zeit gemessen (in 1/10 sec), die es dauert, bis E17 wieder losgelassen wurde. Diese Zeit wird dann im Display 1 des Terminals ausgegeben und das Programm wartet auf ein erneutes Drücken von E17. Wird der Taster vorher losgelassen, so erscheint ein Meldungstext „Geschummelt!“ und das Programm wird beendet.

*Hinweise zur Programmierung:* Markiere alle wichtige Informationen im Text und versuche, die verschiedenen Zustände des Programms in einem Diagramm festzuhalten. Bei der Programmierung wirst du wahrscheinlich mehrere parallel arbeitende Prozesse<sup>3</sup> brauchen. Welcher Prozess tut was? Wie werden sie synchronisiert<sup>4</sup>? Zeichne ein Flussdiagramm des Hauptprozesses und setze das Programm in LLWin um. Beachte bei der Programmierung, dass die größte angezeigte Zahl im Display „9999“ ist! (Überlauf<sup>5</sup>) (UEB1\_7.MDL)

## (8) Addierer (Taschenrechner)

Bei Programmstart zeigt Display 1 des Terminals die Zahl „0“. Bei einem Impuls auf E17 (Drücken und Loslassen) wird zu der Zahl in Display 1 der Wert des (positiven) Terminalparameters EA addiert, und wieder in Display 1 ausgegeben. Dies soll beliebig oft wiederholbar sein.

*Hinweise zur Programmierung:* Ein Zustandsdiagramm bringt uns bei diesem Programm nicht weiter, da der Zustand immer nur aus dem Merken einer Zahl besteht. Dafür stehen wir vor dem Problem, dass LLWin keine Rechenarten zu Verfügung stellt, und wir so die Addition mit anderen Mitteln lösen müssen. Welche Möglichkeiten haben wir? Beschreibe informell<sup>6</sup>, wie du vorgehen willst, und zeichne ein Steuerflussdiagramm. Übertrage dein Programm nach LLWin. (Speichern unter (UEB1\_8.MDL)

*Erweiterung:* Bei Impuls auf E18 soll zusätzlich eine Subtraktion durchgeführt werden (UEB1\_8A.MDL). Welche Funktionen kannst du noch einbauen?

<sup>3</sup> Prozesse = (Programm-)Fäden, Abläufe

<sup>4</sup> Synchronisiert = aufeinander abgeglichen, entspricht hier dem Austausch von Daten (Informationen) zwischen den Prozessen

<sup>5</sup> engl. Overflow = Der Wert einer Variable wird größer als erlaubt, oft wird dann auf 0 zurückgesetzt

<sup>6</sup> informell hier: indem du einen beschreibenden Text verfasst