Automatisierungssysteme

Thierry Prud'homme
thierry.prudhomme@hslu.ch

Aufgabenserie: **#3** Themen: **Endliche Automaten**

[Aufgabe 1] (*Import/Export*) TwinCAT PLC Control gibt Ihnen die Möglichkeit einzelne oder mehrere Bausteine zu exportieren um anschliessend in einem anderen Projekt wieder zu importieren. Öffnen Sie das Projekt, inwelchem Sie den Funktionsblock `EdgeDetector` verwirklicht haben. Exportieren Sie den Baustein mithilfe des Kontextmenus. TwinCAT PLC Control legt dazu eine .exp-Datei an.

Erstellen Sie ein neues Projekt für diese Aufgabenserie und importieren Sie den Funktionsblock unter Projekt – Importieren...

[Aufgabe 2] (*Enumeration*) In dieser Aufgabenserie schreiben Sie eine Applikation die einen Tresor simuliert, der sich öffnet sofern die richtige Tastenkombination gedrückt wird. Diese Anwendung ist ein typisches Beispiel für einen endlichen Automaten, welche oft mit einer SPS umgesetzt werden. Das hier behandelte Beispiel ist dargestellt in der Abbildung 1.

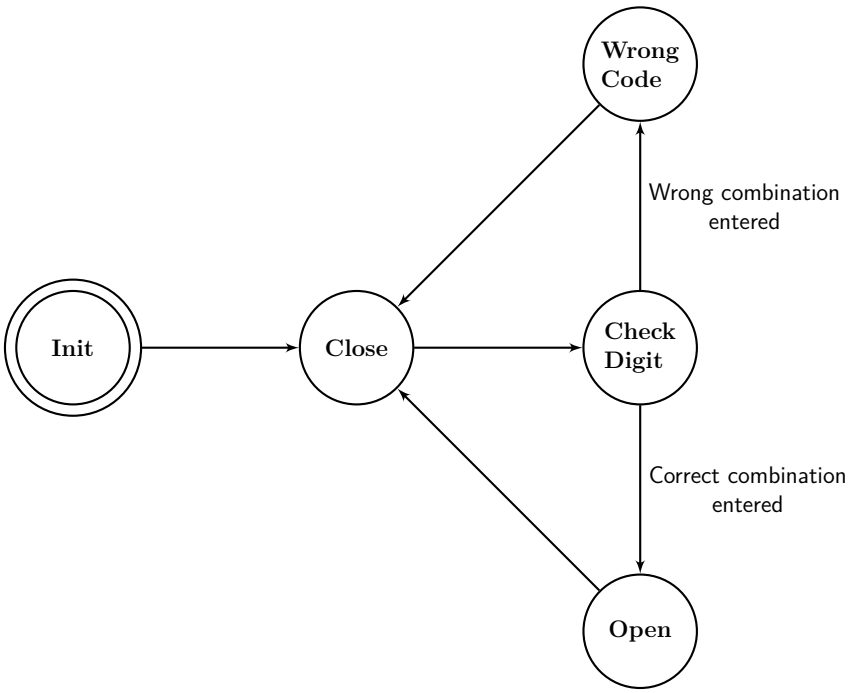

Abbildung 1: Endlicher Automat

Innerhalb der TwinCAT PLC Control Umgebung können auch Aufzählungen (Enums) deklariert werden. Dies geschieht ähnlich wie bei den Strukturen unter den Datentypen, mit dem Unterschied, dass das die Tags `STRUCT` und `END_STRUCT` entfernt werden müssen. Erstellen Sie eine Aufzählung `SafeState` mit allen nötigen Einträgen für die Umsetztung des hier behandelten endlichen Automaten.

[Aufgabe 3] (`CASE`-*Anweisung*) Die unter dem Namen Switch geläufige Anweisung lautet `CASE` in ST. Instanzieren Sie innerhalb Ihres `MAIN`-Programms eine `currentSafeState` Variable des Typs `SafeState`. Erstellen Sie eine `CASE`-Anweisung für diese Variable und ihre möglichen Zustände.

[Aufgabe 4] (*Endlicher Zustandsautomat*) Für die Vervollständigung Ihrer Anwendung fehlen noch die folgenden Teile:

- Globale Variablen für die Taster und Lämpchen
- Eine Instanz des `EdgeDetector` Funktionsblocks für jeden Taster
- Ein Array `codeArray` des Typs `BOOL` mit 4 Einträgen für den Geheimcode des Tresors
- Eine `currentIndex` Variable des Typs `INT` um auf das `codeArray` zuzugreifen
- Ein Zähler `correctCounter`, der die richtig eingegebenen Tastenanschläge zählt
- Und eine Konstante `lightDuration`, initialisiert auf 5 Sekunden, für die Brenndauer der Lämpchen

Mittlerweile haben Sie alle nötigen Teile für die Fertigstellung des endlichen Automaten. Verwirklichen Sie noch die folgenden Anweisungen für die einzelnen Zustände innerhalb der `CASE`-Anweisung:

- `SafeStateInit`: Füllen Sie das `codeArray` mit Kombinationen von `TRUE` und `FALSE`. `TRUE` steht für eine Betätigung der blauen Taste, `FALSE` für die gelbe. Zum Beispiel: `TRUE, FALSE, FALSE, TRUE` bedeutet, dass der Benutzer blau, gelb, gelb, blau betätigen muss um den Tresor zu öffnen. Springen Sie weiter zu `SafeStateClose`.
- `SafeStateClose`: Setzen Sie alle Zähler und Lämpchen zurück. Springen Sie weiter zu `SafeStateCheckDigit`.
- `SafeStateCheckDigit`: Handeln Sie nur wenn eine `risingEdge` detektiert wird. Wird die im Array erwartete Taste detektiert, inkrementieren Sie den `correctCounter`. Inkrementieren sie auch den `currentIndex`. Passen sie auf, dass sie die Inkrementierungen an der richtigen Stelle vornehmen. Wiederholen Sie diesen Schritt vier Mal. War die Codeeingabe erfolgreich soll das Programm nach `SafeStateOpen` springen, sonst nach `SafeStateWrongCode`.
- `SafeStateWrongCode`: Lassen Sie das rote Lämpchen für die Länge `lightDuration` aufleuchten und transitieren Sie anschliessend zurück zu `SafeStateClose`.
- `SafeStateOpen`: Lassen Sie das grüne Lämpchen für die Länge `lightDuration` aufleuchten und transitieren Sie anschliessend zurück zu `SafeStateClose`.

Wenn Sie die Verknüpfung der Hardware mit der Software vorgenommen haben, sind sie

bereit die Grundfunktion Ihres Programms zu testen. Sobald Ihre Anwendung funktionsfähig ist, sollten Sie die folgenden Zusatzfunktionen einbauen:

- Wird für den blauen Taster ein `doubleTap` detektiert, soll die Codeeingabe abgebrochen werden können. Das heisst, es müssen nicht 4 Tastenanschläge detektiert werden bevor mit einer neuen Eingabe gestarten werden kann.
- Der Benutzer soll den Code für den Tresor von Hand umprogrammieren können. Um diese Funktion zu aktivieren, muss sich der Tresor im geöffneten Zustand befinden während beide Tasten gleichzeitig betätigt werden. Das Programm soll an dieser Stelle zu einem weiteren Zustand springen, welche eine vierstellige Codeingabe erwartet.