

Complete Language Support and Error Detection of Sequential Function Charts in Twistturn

(Bachelor Thesis)



MAX BÖCKER

Motivation

Hardware-in-the-Loop (HiL) testing is an important step in the development of Programmable Logic Controllers (PLCs). To ensure that PLC programmers are not required to learn a tool-specific test case specification language, test cases should be created in PLC programming languages, such as Sequential Function Chart (SFC). To enable HiL testing in SFC, an editor is required that supports the full functionality of SFC. SFC programs consist of steps at which actions are executed and conditional transitions that connect individual steps. Furthermore, there is the possibility to create alternative and parallel paths with the help of branches and joins.

State of the Art

Twistturn is a HiL testing environment which allows the user to create test cases in PLC languages. One of these languages is SFC. Twistturn currently supports creation, editing, and execution of SFC programs as well as their use for HiL simulations. The editor is not able to manage alternative and parallel branches, nor is it possible to use more than one SFC per project. Since SFCs are not validated before execution, it is not possible to inform the user if they start an erroneous program. Furthermore, Twistturn cannot detect the equality of SFCs, so they are recompiled each time they are executed.

Objective

The objective of this thesis is to extend the SFC editor of Twistturn in a way that it supports missing language elements, for example alternative and parallel branches. The next step is to adapt the execution environment to be capable of executing two or more SFCs in one project. SFCs only need to be recompiled in the event of change and therefore Twistturn should be able to detect the equality of SFCs. Since an SFC with an invalid structure, for example two transitions directly following each other, leads to undefined behavior, the user should be informed if they try to execute a faulty SFC.

Approach

It must be analyzed how the missing language elements can be inserted into the existing implementation. In addition, it must be evaluated how Twistturn can be enabled to execute several SFCs simultaneously. The structure of valid SFCs must be defined so that a routine can be used to validate an SFC. To enable Twistturn to detect the equality of SFCs, hash values are to be calculated for them and used as a basis for comparison.