Utilizing Sequential Function Charts to Specify Hardware-in-the-Loop Tests

(Master Thesis)

MARCO NICOLINI

Motivation

Testing is crucial to avoid system failures. This is especially important where failures are safety-critical or expensive. Concerning the V-model for software engineering, Hardware-in-the-Loop (HiL) testing comes after software and before system tests. The HiL Simulator simulates the environment to the System Under Test (SUT). It reads the SUT's output and provides the input for the SUT. A PLC programmer is not required to learn a special test specification language, if the HiL tests can be defined with the programming languages for PLC programming standardized in IEC 61131-3.

State of the art

There exist different techniques to specify HiL tests. The two main components for HiL testing are specification of test steps and acceptance criteria. Some tools have a plant model as additional component. The languages for the test specification are textual or graphical. Some use 3D models to visualize the test. The HiL Simulator must fulfill real-time requirements to allow for accurate testing. Real-Time Android (RTAndroid) extends Android by real-time capabilities. Twistturn provides a Soft PLC using RTAndroid and can be used as HiL Simulator. It supports the PLC languages Structured Text (ST) and Function Block Diagram (FBD) for programming.

Objective

The objective of this thesis is to add support for Sequential Function Charts (SFCs) to Twistturn and use them to define HiL tests. Therefore, editing and executing them in Twistturn is necessary. The HiL test steps and plant model should be defined in SFC, as well as acceptance criteria. The SFC language should be extended to fulfill the following requirements: In a SFC step for acceptance criteria, the user can describe logical parts of a signal, as well as adjust the signal tolerances. Examples for logical parts of signals are ramp or pulse-width modulation signals.

Approach

The SFC support will be added based on an evaluation of existing SFC editors and other graphical editors. The existing FBD editor can be used as foundation for the SFC editor. ST will be used for the translation from SFC to execution code. The actions and conditions for SFC can be defined in ST and FBD.

The extension of SFC will come as a library with functions that can be used in the SFC steps' actions. There will be functions for defining signal parts, as well as functions for adjusting the tolerances. The latter will only be available for the acceptance criteria and not for the test steps.

Advisor: David Thönnessen
11.05.2017

Informatik 11
Embedded Software
RWTH AACHEN UNIVERSITY