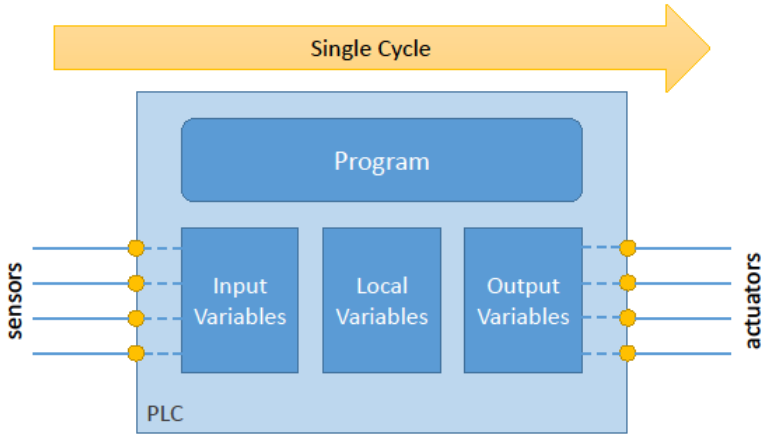
Bachelor/Master Thesis

Bounded Model Checking of Programmable Logic Control Software

Introdcution to the Problem

Programmable Logic Controllers (PLCs) are devices commonly used for industrial automation tasks, e. g. monitoring the output flow of a tank and controlling appropriate valves. Being used for repeatedly performing the same task, PLCs adopt a cyclic execution mode, i. e. reading inputs (typically connected to sensors), executing the main program, writing outputs (typically connected to actuators) and starting all over again.



PLC CYCLE

In this context, one property of interest is whether it is possible to reach undesired behaviour, e. g. division by zero. Bounded Model Checking (BMC) is a sound (but not complete) technique for checking reachability of such behaviour (bounded to k instructions). In the PLC domain bad behaviour often relates to the valuation at the end of a cycle instead of the intermediate values during program execution. Unfortunately, checking the program's safety for k instructions does not state how many cycles are safe to execute since paths through the program have different lengths.

Task

To solve the problem at hand, a procedure for bounded checking of PLC programs will be developed where k refers to the number of cycles (instead of instructions or larger blocks). To this end several challenges must be faced:

- Finding a suitable logical characterisation of the program (for use in an SMT solver) which supports incremental checks
- Handling of paths of different length during a cycle, esp. loops
- Implementing a prototype in the ARCADE.PLC framework

Prior Knowledge

Necessary	Helpful
Interest in formal methods	Lecture: Formal Methods for Logic Control Software
Be able to work independently & self-reliant	Lecture: Static Program Analysis
Java Programming	Lecture: Satisfiability Checking

Contact

Dimitri Bohlender, M. Sc. RWTH
bohlender@embedded.rwth-aachen.de