Extending Fladuino with support for expressing networks of embedded devices

Ulrik Rasmussen & Philip Lykke Carlsen

February 9, 2010

Project contents

The proposed project is an adaption of a bachelor's thesis from 2009[1] in which a framework called Fladuino is created, for programming the Arduino devices in a declarative functional style. The project is an adaption of the Haskell-embedded domain-specific language *Flask*, which is used for programming sensor networks.

As it is, Fladuino only covers the programming of individual devices. If multiple Arduinos were to communicate with each other in a network (for instance, via bluetooth), possibly with different programs, there would be no support for sanity-checking that the programs actually transmit valid (i.e. type-safe) messages to each other. It would also be up to the programmer to manually establish connections to the other units, and to define a common protocol.

This project aims on turning Fladuino into a system for describing *networks* of communicating machines instead of a system just for describing the dataflow within each machine. The Arduino is also a configurable device, with support for adding numerous modules (called 'shields') which the programs can use. It is also the aim for this project to give Fladuino support for *inferring* the hardware configuration of individual machines, based on the programs that they are going to execute.

Expected result

The result of the project will be a Haskell framework containing an embedded domain-specific language for programming networks of Arduinos in a declarative way. The framework will be capable of generating C-programs which can be compiled and executed on the Arduino, as well as instructions on how to configure each individual machine to conform to the program that is to execute on it. The instructions for the device configurations should be

sanity checked , so it is checked if the required hardware configuration is actually possible.

Motivation

If cheap, embedded systems like the Arduino are to be deployed in vast numbers, it would be an impediment if Fladuino didn't have direct support for networking machines in a declarative, type-safe way.

To ease the deployment and prototyping of larger networks of machines, it would also be of help if the framework could detect impossible hardware configurations, as well as aid in the wiring of the actual hardware modules.

What is new?

To our knowledge, Fladuino is the only framework for reactive programming that allows programming small embedded devices like the Arduino. Other frameworks based on the same paradigm requires a full language runtime, which doesn't fit the limited resources of the Arduino. This adaption would therefore be the only way to program *networks* of Ardunios in a framework for reactive programming.

Project risks

We have no prior experience with the Arduino. We can therefore risk running into a lot of unforeseen challenges which will consume a lot of our time. On the positive side, we both have experience with the Haskell programming language.

Bibliography

[1] Troels Henriksen, Martin Dybdal, and Jesper Reenberg. Fladuino: Controlling embedded devices with functional reactive programming. july 2009.