Abstract

Dataflow-based programming is an interesting software development methodology. In contrast to the widely used imperative paradigm, it does not focus on the individual instructions of the program, but the data which is being processed instead, along with its transformations and flow. In this model, we represent the software as a network of logical components executed in parallel, which communicate with each other via message passing. This approach is a great fit for the signal processing tasks commonly encountered in embedded systems. The goal of my Thesis, is to design and implement a framework which enables the development of dataflow-based software in embedded environments, mainly for usage on modern microcontrollers. The implemented system supports the creation and runtime instantiation of pre-defined logical components and the dynamic configuration of the dataflow network. The system configuration is handled by a management layer (engine) accepting external commands, which is capable of controlling the execution of the components and changing the logical connections between them at runtime.

In this document, I summarize the literature of the subject and create a model of the structure and behaviour of the system using UML tools. Finally, I describe the implementation process of the main software layers, paying special attention to the functional testing aspect.