Hes·so
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences and Arts Western Switzerland

Travail de Master of Science HES-SO en Engineering

ESPeclal : an Embedded Systems Programming Language

Christopher Métrailler

Professeur responsable : **Pierre-André Mudry**

HES-SO Valais

DESCRIPTION

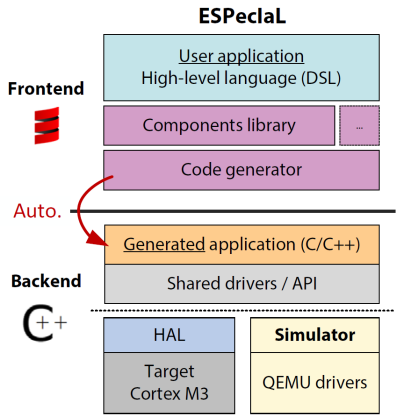
Nowadays embedded systems, available at very low cost, are becoming more and more present in many fields such as industry, automotive and education. This project presents a prototype implementation of an embedded systems programming language.

A high-level language has been developed to build embedded applications, based on the dataflow paradigm. Using ready-to-use blocks, the user describes the block diagram of his application, and its corresponding C++ code is generated automatically, for a specific target embedded system.

As a proof of concept, the embedded systems programming language is developed using an internal Domain Specific Language (DSL) in Scala. With the help of this high-level language, embedded applications can be built with ease, avoiding the use of low level programming languages like C or C++. This prototype language helps non programmer users to build portable applications for embedded systems.

GOALS

1. Specification of the embedded programming language, based on the dataflow paradigm.
2. Implementation of the prototype language using a custom dataflow DSL. The language is developed in Scala in an internal DSL. Several components / blocks have been developed to build the application logic and to access to specific microcontroller peripherals, such as Gpio, external interrupts, PWM outputs or analog inputs.
3. Setting up a testing environment to validate the generated C++ code.
   Tests can be made on the real target or in a custom version of the QEMU emulator.



Architecture overview

ARM 32 bit CORTEX M3™

STM32F103
72 MHz

Target

RESULTS

1) Sample application

The behavior and the block diagram of an application is described using the developed dataflow DSL. Ready-to-use components are available in the framework. The user can connect these blocks together to build the application.

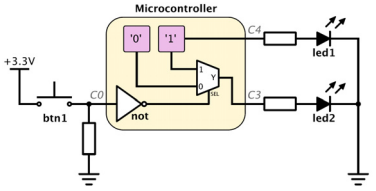
User application specification (DSL)

```
val not = Not()
val mux = Mux2()
Stm32stkIO.btn1.out --> not.in

val cst1 = Constant(true).out
mux.out --> Stm32stkIO.led1.in
cst1 --> Stm32stkIO.led2.in

Constant(false).out --> mux.in2
not.out --> mux.sel
cst1 --> mux.in1
```

Corresponding visual representation




```
while(1) {
   // 1) Read inputs
   bool in_C0 = in_cmp02.get();
   // 2) Loop logic
   uint8_t out_cmp01 = !in_C0;
   uint8_t sel_cmp03 = out_cmp01;
   bool out_cmp03;
   if(sel_cmp03 == 0)
      out_cmp03 = false;
   else
      out_cmp03 = true;
   // 3) Update outputs
   out_cmp05.set(out_cmp03);
   out_cmp06.set(true);
}
```

Generated C++ code for the target

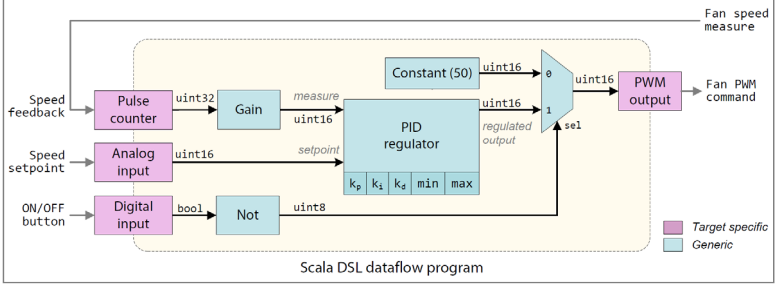
The developed framework generates automatically the C++ code corresponding to the dataflow application (DSL).

The code can be used out of the box :

- On the target embedded system
- Or it can be simulated in QEMU

2) Real-world regulation application

Speed regulation of a PC fan. Application developed with the DSL, without C/C++ code.



Scala DSL dataflow program

CONCLUSION

With the help of this prototype Domain Specific Language (DSL), embedded applications can be built with ease using an high-level language. Ready-to-use components are available in the framework. The user can connect these blocks together to build applications. Low-level C/C++ code is no more necessary. It is generated automatically.

The model / the block diagram of the application is written using a custom dataflow DSL and the generated C++ code can be used out of the box - without modification - to program the embedded target. One ARM Cortex M3 board is supported at this time. The generated application can also be tested automatically using a Scala test case, in a modified version of QEMU.