EmbedFX: New Programming Language for Embedded Systems with Asynchronous Effects

Danel Batyrbek　Yukiyoshi Kameyama

While embedded systems are recognized to be important for the proper functioning of our society, it is extremely difficult to write safe and reliable software because of inherent limitations. Among the ongoing research of embedded software, several programming languages have been proposed, improving on specific aspects of the C programming language. However, those languages continue to occupy a niche of research languages, without mass adoption of the wider community. EmbedFX is a new programming language specifically designed for embedded systems. The goal is to make development for embedded systems simpler, without foregoing the benefits of the expressive nature of C. EmbedFX draws inspiration from OCaml and incorporates asynchronous effects, a variation of traditional effect handlers, to represent the asynchronous nature of embedded software. EmbedFX additionally retains many of OCaml's powerful features such as static type checking, expression-oriented syntax, and immutability. The language transpiles to C and leverages the FreeRTOS API to implement its asynchronous effects.

1 Introduction

Programming for embedded systems comes with unique challenges, such as limited resources, the necessity for real-time responsiveness, and the complexity of handling concurrent operations. Historically, C and C++ have been the go-to languages for embedded programming because of their ability to directly interface with hardware, though this often means sacrificing the advantages of high-level languages. Drawing inspiration from the functional programming language OCaml, our goal is to offer a high-level development experience while producing efficient code for microcontrollers with EmbedFX —a statically typed, expression-oriented, and immutable language that includes built-in support for asynchronous effects.

2 Language Features

EmbedAsync incorporates several key features from OCaml, adapted specifically for the embedded systems field:

1. Ensures type safety at compile time, minimizing runtime errors and improving code reliability.
2. Promotes a concise and expressive coding style, making the development process more straightforward.
3. Encourages the use of immutable data structures to prevent unintended side effects and foster safer code.

2.1 Asynchronous Effects

A significant innovation in EmbedFX is the integration of asynchronous effects [1], which allow developers to manage asynchronous operations in

EmbedFX: New Programming Language for Embedded Systems with Asynchronous Effects.

Danel Batyrbek, Yukiyoshi Kameyama, 筑波大学 情報理工学位プログラム, Department of Computer Science, University of Tsukuba.

a natural and efficient manner. These effects enable the creation of complex asynchronous control flows that are both type-safe and capable of tracking user-defined operations without compromising asynchronicity, offering a streamlined approach to handling concurrent tasks common in embedded systems.

Asynchronous effects in EmbedFX are designed to:

- Maintain clear dependencies between asynchronous tasks, ensuring proper communication.
- Utilize non-blocking operations to preserve system responsiveness, which is crucial for real-time embedded applications.
- Leverage the language's static type system to enforce correct usage of asynchronous effects, reducing the likelihood of errors.

2. 2 Transpilation to C and FreeRTOS Integration

To maintain compatibility with a broad range of embedded hardware, EmbedFX transpiles to C. This strategy allows developers to take advantage of the language's high-level abstractions while generating C code that is well-suited for resource-constrained environments. The transpilation process is carefully designed to produce C code that not only benefits from existing libraries but also allows for further modification and patching after transpilation.

EmbedFX integrates with FreeRTOS, a widely-used real-time operating system for embedded devices, to implement the semantics of its asynchronous effects. By utilizing the FreeRTOS API, EmbedFX ensures that its asynchronous operations are both efficient and reliable, fully leveraging FreeRTOS's capabilities for task scheduling, synchronization, and inter-task communication.

3 Conclusion

EmbedFX seeks to enhance the landscape of programming languages for embedded systems by merging the expressive power and safety features of OCaml with advancements in programming language research. Its goal is to elevate the developer experience beyond that of traditional C. Despite this, the ability to transpile to C ensures compatibility with existing C codebases, while also simplifying the use of the FreeRTOS API in a type-safe and reliable way.

4 Future Work

Future developments for EmbedFX will aim to offer stronger guarantees and a more ergonomic design, including features like the ability to define atomic operations and prevention of most race conditions, to ensure a safer and more reliable development experience for embedded systems. The generated C code is currently produced conservatively, so optimizing it for readability is a key area for improvement. Additionally, while the current implementation of EmbedFX is tailored for the ESP32, expanding support to other microcontroller families is under consideration.

参考文献

[1] Ahman, D. and Pretnar, M.: Asynchronous effects, *Proceedings of the ACM on Programming Languages*, Vol. 5, No. POPL(2021), pp. 1–28.