

# 関数リアクティブプログラミング言語による 小規模組込みシステムのプログラミング

東京工業大学 情報理工学院 情報工学系 渡部研究室



組込みシステムは、センサやスイッチ、通信ポートを介した外界からの入力に反応し、モータやディスプレイ等の機器を制御します。このような、外界からの入力に反応し続けるような動作をするシステムを**リアクティブシステム**と呼びます。組込みシステムの他にも、GUI、Webアプリケーション、ゲーム等はリアクティブシステムとみなすことができます。

一般にリアクティブシステムに与えられる入力は順序やタイミングが予測できません。そのため、ポーリングや割り込み、イベントシステム、状態機械と呼ばれる手法がリアクティブシステムのプログラミングにおいて現在一般的に用いられています。ところが、これらの手法を用いるとプログラムが分断されて非常に見通しが悪くなり、作成や保守のコストが増大します。

**関数リアクティブプログラミング(FRP)**は、副作用のない式のみを用いてリアクティブシステムのプログラムを記述しようという考え方です。これによってプログラムの見通しをよくなり、保守コストを下げようという目的で研究が行われています。

関数リアクティブプログラミングによってリアクティブシステムの動作が見通しよく書けるのは、**時変値(Time Varying Value)**と呼ばれる仕組みによります。これは時々刻々と変化するデータをそのまま表現する機構です。例えば温度センサを表す時変値を `temp` とし、ファンのON/OFFを表す時変値を `fan` とします。気温が30度を超過している間のみファンを回すプログラムは以下の1行で書くことができます。

```
node fan : Bool = temp > 30
```

従来のプログラミング手法では、例えば気温の変化に追随するには、ポーリングによって温度センサによる測定を繰り返すか、あるいはセンサからの割り込みによって変化を知る必要がありました。FRPでは、センサの測定値やアクチュエータの状態を時変値として表現することでプログラムを簡潔に記述することができますようになります。

渡部研究室では、特にマイクロコントローラ等を用いた小規模組込みシステムを対象とした、FRPによる開発支援についての研究を行っています。これまでに、オリジナルのFRP言語Emfrp[1]等の開発を通して、FRPが小規模組込みシステム開発に有用であることを明らかにしてきました。現在までに、並行計算モデルであるアクターモデルとFRPの融合による非同期性の表現と分散システムへの対応[3]、文脈指向プログラミング機構[2]や状態記述機構[6]の導入によるモジュール性の強化、マルチコア向け実装方式による応答性向上[4]、およびメモリサイズの静的見積もりが可能な再帰的データ型の導入による表現力の強化[5]などについて研究を行っています。

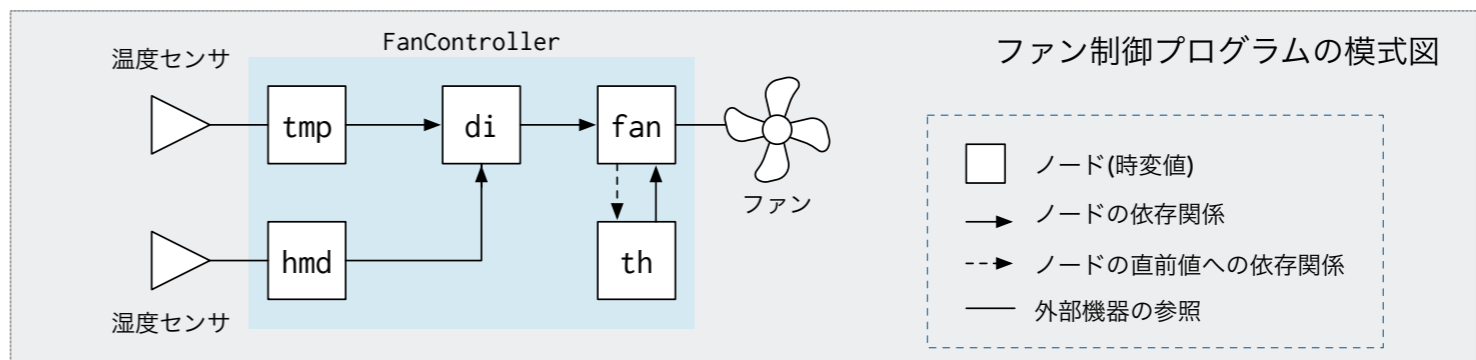
```
module FanController # module name
in tmp : Double,     # temperature sensor
   hmd : Double     # humidity sensor
out fan : Bool,     # fan switch
   di : Double      # discomfort index
use Std

# discomfort (temperature-humidity) index
node di = 0.81 *. tmp +. 0.01 *. hmd *. (0.99 *. tmp -. 14.3) +. 46.3

# fan switch
node init[False] fan = di >=. th

# threshold (w/ hysteresis offset)
node th = 75.0 +. (if fan@last then -.0.5 else 0.5)
```

FRP言語Emfrpによるファン制御プログラムの例



- 主な発表論文
- [1] K. Sawada and T. Watanabe, Emfrp: A Functional Reactive Programming Language for Small-Scale Embedded Systems, CROW 2016, pp. 46-54, ACM, Mar., 2016. DOI:10.1145/2892664.2892670
  - [2] T. Watanabe, A Simple Context-Oriented Programming Extension to an FRP Language for Small-Scale Embedded Systems, COP 2018, pp. 23-30, ACM, Jul., 2018. DOI:10.1145/3242921.3242925
  - [3] K. Shibanaï & T. Watanabe, Distributed Functional Reactive Programming on Actor-Based Runtime, AGERE 2018, pp. 33-42, ACM, Nov., 2018. DOI:10.1145/3281366.3281370.
  - [4] Y. Sakurai & T. Watanabe, Towards a Statically Scheduled Parallel Execution of an FRP Language for Embedded Systems, REBLS 2019, pp. 11-20, ACM, Oct., 2019. DOI:10.1145/3358503.3361276.
  - [5] 横山・森口・渡部, 小規模組込みシステム向けFRP言語に対する再帰的データ型の導入, PPL 2020, Mar., 2020.
  - [6] 松村・渡部, 組込みシステム向けFRP言語における状態依存動作のための抽象化機構, 情報処理学会論文誌(プログラミング), 2020 (掲載予定).