



小規模組込みシステム向けFRP言語における更新間隔の切り替え機構

東京工業大学 情報理工学院 渡部研究室
横山 陽彦・森口草介・渡部卓雄

概要 (WIP)

目的 小規模組込み向けFRP言語における省電力効果の高い間欠実行の実現

手法

- ・ タイミング注釈の拡張
- ・ 更新間隔の動的変更機構の導入

貢献

- ・ 「あるタイミングを起点に時変値の更新間隔が切り替わる」動作が記述可能
- ・ 不必要な更新を減らすことによる省電力化の実現

PbEmfrp [1] / EvEmfrp

小規模組込みシステム向けFRP言語 Emfrp [2]の拡張

- ・ **時変値更新タイミング**を型に含む
 - ・ 周期的タイミング (φ, T): (開始時刻, 周期)
 - ・ タイミング合成, 部分化
 - ・ 割り込みタイミング $\$a$: タイミングラベル a
- ・ プログラム全体のタイミング注釈を元にシステム全体の時変値更新周期を合成, 更新処理中でない場合スリープモードに移行することで**省電力化**を図る.
 - ・ 従来のEmfrpではポーリングによる更新処理
- ・ @last[n]による直前値参照が可能
- ・ Emfrpと同様に使用メモリ量の静的な決定が可能

プログラム例

```
module BtnBlink # ボタンを押すとLEDの発光状態が変わる
in btn_in (False) : Bool '(0, 5),
# 5ms周期で更新される時変値
intr : Bool '$a # 割り込みで更新される時変値
out blink_led : Bool '(0, 5),
led : Bool '$a

node btn = btn_in@last && btn_in
node init[False] blink_led = if btn then
not blink_led@last
else blink_led@last

node led = not intr # 同じ割り込みタイミング$aで更新
```

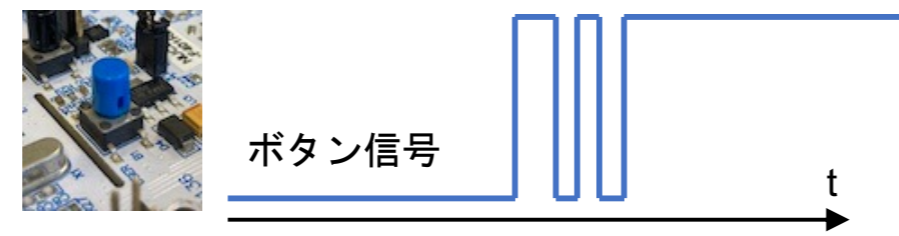
実行時の振る舞い

- ・ 周期イベントと被周期イベントは同時発生しないと仮定
- ・ 割り込み\$aは静的に生成されるスケルトンコード内にて発行
- ・ システム全体の更新周期の際に, 更新すべきノード群を選択し, それらのみを更新

例題: 押下判定 (チャタリング除去)

問題設定

- ・ マイコンにタクトスイッチが接続されている.
- ・ スwitchを押した直後からある程度の時間は信号が乱れ(チャタリング), その後安定する.
- ・ ボタンの適切な押下判定をしたい.



解決策①

- ・ 適切な間隔でサンプリングし続け, 2回連続でHighならば押下と判定.
→ ほとんどの期間で押下なし. 必要以上にシステムがWakeupする.

解決策②

- ・ 最初の立ち上がりエッジを割り込みで検出, 一定時間後にもう一度センシングして, Highならば押下と判定する.
→ 既存のEvEmfrpでは「一定時間後にもう一度センシング」するプログラムの記述ができない

提案手法: タイミング注釈の拡張

タイミング定義(タイミング名:制約)をモジュールに記述可能

```
module Sample
in x : Int at A out y : Bool at A
timing A : Interrupt[50 ms]
node y = x > 0
```

タイミング注釈を「制約」と解釈して記法を拡張

- ・ **Interrupt[n]**: 少なくともnミリ秒間隔の割り込み
- ・ **Peripheral[n]**: 少なくともnミリ秒間隔の割り込みを発生させる外部デバイス
- ・ **t -> (p, T) | ... | t' -> (p', T')**
 - ・ t, t': 切り替えの起点となるタイミング名
 - ・ (p, T), (p', T'): 相対周期タイミング(pとTは ∞ を含む自然数定数値)
 - ・ 起点となるタイミングが発生するたびに周期タイミングが切り替わることを意味する.
- ・ **When b**: ノードbが真になったタイミングを表す
- ・ **init**: プログラム実行開始を表すタイミング名

拡張記法による例題の記述

```
module BtnOn # 押下判定モジュール
in fst : Unit at Ta,
snd : Bool at Tb
out btn : Bool at Tb
timing Ta : Interrupt[5ms],
Tb : init -> ( $\infty, \infty$ ) | Ta -> (10ms,  $\infty$ )
node btn = snd
end

module Toplevel # トップレベルモジュール
in btn_intr : Unit at Ti,
btn_in : Bool at Tx
out blink_led : Bool at Tx
timing Ti : Peripheral[5ms], Tx : Interrupt[1ms]

# サブモジュール利用
newnode btn = BtnOn(btn_intr, btn_in)

node init[False] blink_led : Bool at When btn
= not blink_led@last
end
```

タイミングTaの10ms後に発生するone shotなタイミングを表す

周期タイミングが発生する前にもう一度Taが発生すると, その時点から10ms後に周期タイミングが発生する

割り込み信号を発生させるデバイスの指定

Interrupt制約な入出力ノードはランタイムが適切に入出力コールバック関数(C言語)を呼び出す

blink_ledはbtnがTrueの時だけ更新される(ノイズによってボタン信号が一瞬だけ立ち上がるとbtnはFalseになりうる)

When btnとInterrupt[1ms]に整合性があるか静的に検査する予定

関連研究

・ mTask [3]

純粋関数型言語Clean上に定義されたIoT機器向けDSL. ペリフェラルのリフレッシュレート(更新周期)を基にスリープタイミングを決定する. 周期は定数値.

・ SynchronVM/API [4]

関数型言語による組込みシステムのプログラミング環境. TinyTimberカーネルに由来するタイミング注釈(遅れ, 締め切り)を利用した同期的メッセージパッシングをサポート.

似ている研究, 言語機構ぜひ教えてください!

今後の方針

- ・ タイミング(制約)の意味の形式化
- ・ 明らかな制約違反の検出
- ・ スケジューリング(相対的な時間を経過時刻に変換, スリープタイミングの決定)
- ・ 省電力効果の評価

[1]辻 et al. 小規模組込みシステム向けFRP言語における周期的タスクの記述方式. 日本ソフトウェア科学会第38回大会, 2021.
[2] Kensuke Sawada, et al., Emfrp: A Functional Reactive Programming Language for Small-Scale Embedded Systems, MODULARITY 2016
[3] Crooijmans, S. et al. Reducing the Power Consumption of IoT with Task-Oriented Programming, TFP 2022.
[4] Abhiroop Sarkar, et al. Synchron - An API and Runtime for Embedded Systems. ECOOP 2022