

Lego MindStorms制御プログラムの対話型開発・実行環境

開発者：湯浅 太一

XS Lisp で自作ロボットららくコントロール



Why Lisp?

- 対話型プログラム開発環境
- 部品動作を確認しながらプログラム構築
- データ, 式, プログラムがすべてS式
- 小さくても高機能 (関数閉包, 高階関数, 非局所的脱出, 末尾再帰呼出しの最適化, ...)

ダウンロードサイト

<http://www.xslisp.com>

XSの特徴

- 対話型プログラム開発環境
 - read-eval-print ループ
 - 対話的に関数を定義, 再定義
 - 適切なエラーメッセージとバックトレース表示
 - 関数トレース機能
- 実行系はRCX 内で自律的に動作
 - オブジェクトの動的割り付け, ごみ集め
 - 末尾再帰的なインタープリタによる実行
 - プログラムエラーやスタック・バッファオーバーフローに対して頑健
 - 端末機割り込み
- ロボット制御に十分なプログラム記述能力
 - SchemeベースのLisp言語
 - Legoデバイス(モータ, センサ, ランプ等)のためのインターフェイス
 - イベント待ち, タイマ, 非同期イベント割り込み

プログラム例: 障害物回避

```
(begin
  (speed :a (speed :c :max-speed)))
  (let loop ()
    (motor :a (motor :c :forward))
    (wait-until (or (touched? 2)
                    (pressed?))))
    (if (pressed?)
      (motor :a (motor :c :off))
      (begin
        (motor :a (motor :c :back))
        (sleep 5)
        (motor
         (if (= (random 2) 0) :a :c)
         :forward)
        (sleep 5)
        (loop))))
  )))
```

