Lego MindStorms 制御プログラムの対話型開発･実行環境

― XS Lisp で自作ロボットらくらくコントロール ―

1. 背景

わが国でもブロック玩具でよく知られている Lego 社が，MindStorms とよばれるロボットシステムを提供している．このシステムは，8 ビットのプロセッサを備えた RCX と呼ばれる「ブロック」にプログラムを与え，Lego 社が提供するモーターや各種センサー，汎用のブロックを組み合わせることによって，さまざまなロボットを構築することができる．小学生でも自分でプログラムを書いてロボットを制御することができ，比較的安価である（RCX，700 個以上の部品，付属ソフトウェア込みのセットが3万円以下）ことや，関連書籍が数多く出版されていることなどから，子供の玩具，大人のホビー，さらにはプログラミングや機械工学の学習教材としても利用されている．

Lego 社が提供している MindStorms 用のプログラム開発環境では，きわめてシンプルなビジュアル言語を使用してプログラムを作成する．そのシンプルさのために，子供でも利用できると言われている．その反面，ビジュアル言語という特殊なクラスの言語であるために，現実のプログラミング言語と大きくかけはなれており，また機能の制約が大きい（関数や変数の利用が制限されているなど）ために，本格的なプログラミングには適さなかった．

2. 目的

プログラミングの初心者でも容易に利用でき，高度な制御プログラム開発にも利用でき，しかも対話的に MindStorms 制御プログラムを開発できるプログラミング言語を設計し，その言語で記述されたプログラムを対話的に開発するための総合的なプログラム開発･実行環境を構築する．

3. 開発の内容

上記の目的を達成するために，MindStorms 用の Lisp 系言語を設計し，その処理系を開発した．言語を Lisp 系にしたのは，次の理由による．

- Lisp 系言語は，対話型プログラム開発環境に適しており，実際に従来の大多数の Lisp 処理系は対話型である．
- 部品の動作を確認しながらプログラムを構築することが容易である．
- データ，式，プログラムがすべてS式で表現されている．RCX に搭載されている RAM メモリは32K バイトしかないので，S式の統一表現は，メモリ空間を有効利用するために重要である．
- 比較的小規模な言語仕様であっても，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など，高度な言語機能を提供できる．

RCX の小さなメモリ空間で動作するために，設計した言語とその処理系もきわめて小さいものでなければならない．このために，今回開発したシステムを XS（eXtra Small）と名づけた．

XS システムの主要な特徴を以下に示す．

- 対話型プログラム開発環境である．
  - ♦ Lisp 特有の read-eval-print ループを基本とする．
  - ♦ 対話的に関数を定義し，再定義することが可能である．
  - ♦ 適切なエラーメッセージとバックトレースを表示する．
  - ♦ デバッグのための関数トレース機能を提供する．
- 実行系は RCX 内で自律的に動作する．
  - ♦ データをメモリに動的に割り付け，不要データはごみ集めによって自動的に回収して再利用する．
  - ♦ 末尾再帰的なインタープリタによってプログラムを実行する．
  - ♦ 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である．
  - ♦ 端末機割り込み機能を持つ．
- ロボット制御に十分なプログラム記述能力を提供する．
  - ♦ Scheme ベースの言語である（ただし一級継続はない）．
  - ♦ 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を持つ．
  - ♦ イベント待ち，タイマ，非同期イベント割り込みの機能を提供する．

図1に，XS システムの概要を示す．ユーザがフロントエンド PC 上で Lisp の S 式を一つ入力すると，XS のフロントエンドサブシステムがこれを読み込み，簡単な前処理を行った後，フロントエンド PC と RCX との唯一の通信手段である赤外線通信を使って RCX にダウンロードする．これを RCX 内の XS 実行系が評価（実行）し，結果をフロントエンドに送信する．フロントエンドではこの結果をディスプレイに表示してユーザからの次の入力を待つ．この一連の繰り返しは，通常の Lisp 処理系における read-eval-print ループと同様のものである．ただし XS の場合は，評価する実体が，入出力を担当するフロントエンド PC ではなく，赤外線通信で交信する RCX 上で動作する．

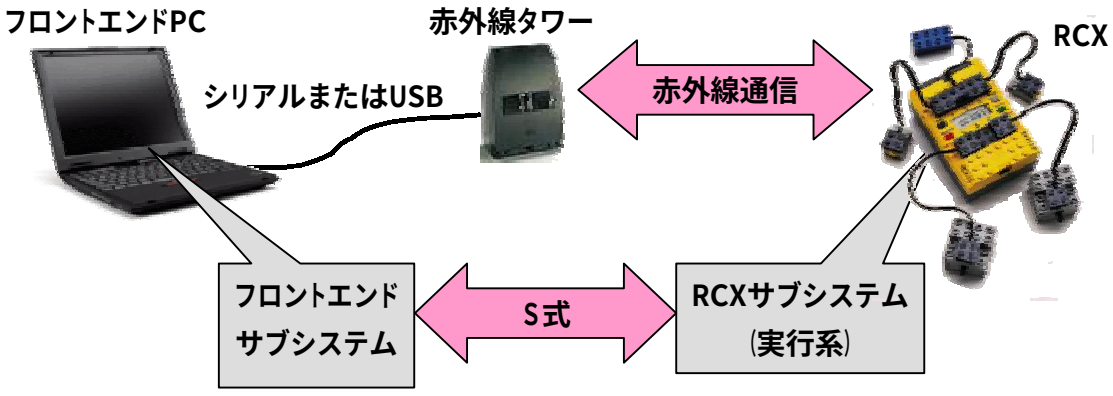

図1　XS システムの概要

XS システムは C 言語で記述されている．フロントエンドサブシステムは通常の C コンパイラを使ってコンパイルするが，RCX サブシステムは，RCX に搭載されているプロセッサであ

る Hitachi H8 用の GNU クロスコンパイラを使って実行形式に変換し，赤外線通信によって RCX にダウンロードする．RCX サブシステムは，BrickOS というオープンソースの MindStorms 用ＯＳの上で動作する．

現時点では，XS のフロントエンドサブシステムは，Windows および Linux で動作する．Windows 版は Windows XP で，Linux 版は RedHat および Debian でそれぞれ動作確認ずみである．MindStorms には，フロントエンド PC と RCX との間の赤外線通信を行うデバイスとして，Lego 社提供の赤外線タワーを使用するが，これには，PC とシリアルで接続するものと USB で接続するものがある．Windows 版ではその両方が利用可能であるが，Linux 版は USB デバイスのドライバが提供されていないため，シリアル方式のみをサポートしている．

４．従来の技術（または機能）との相違点

MindStorms 用には，Lego 社が提供する開発環境の他に，さまざまな言語をベースにした開発環境が Lego 社以外から提供されている．NQC（Not Quite C）というＣ言語に類似のプログラミング言語は，基本的に Lego 社の開発環境のテキスト版である．この言語は，MindStormsのファームウェアの制約から，関数の概念はあるものの，関数相互の呼び出しができないなど，プログラミング言語としては不備である．これに対して，本プロジェクトで設計した Lisp 系言語は，標準的な Scheme 言語よりは機能が劣るものの，実用的なプログラミング言語として十分な機能を備えており，本格的なロボットプログラムの作成が可能となる．

ファームウェアを前述の BrickOS と入れ替えれば，C 言語による制御プログラムの作成が可能である．しかし OS の保護機構が弱く，プログラムにエラーがあれば容易に OS がクラッシュするので，初心者はもちろん，熟練したプログラマにとっても利用するのは困難である．また，C 言語はコンパイラ指向の言語であり，簡単な制御テストをするにも，完結したプログラムを作成し，それをコンパイルして RCX にダウンロードするという面倒な手間を必要とする．これに対して，本プロジェクトで開発中の実行系は，OS とは独立した実行メカニズムを備えており，ユーザプログラムのエラーによって OS がクラッシュすることはない．エラーが発生した場合は，それを実行系が検出し，適切なエラーメッセージを表示するとともに，エラーからの実行回復を自動的に行う．また，対話的なプログラミング環境を提供しているので，ユーザプログラムの構成要素ごとのテストが容易に行える．

本プロジェクトと類似の目的で開発されたプログラミング言語および開発･実行環境として，Lisp 系言語をベースとした LegoLisp と LegoScheme がある．これらの言語では，プログラムはフロントエンド PC で動作し，適宜 RCX に制御命令を送信する．しかし RCX の唯一の通信手段である赤外線通信は通信可能距離が短いうえに指向性が強いために，この方式で移動するロボットを制御するのはきわめて困難である．また RCX に接続した各種センサーからの情報に基づいた制御を行う場合は，センサーからの情報をいったんフロントエンドに取り込み，その情報に基づいて制御命令をフロントエンドから RCX へ発行する必要があるが，赤外線通信が低速であるために，適切な時間内にロボットが反応することは困難であった．これらの問題を解決するために，本プロジェクトで開発する実行環境では，プ

ログラムを RCX にダウンロードし，RCX 上で動作する実行系によってプログラムの実行を行う．

5．期待される効果

以上の特徴をもったプログラム開発･実行環境を提供することによって，次の効果が期待できる．

- RCX のきわめて厳しいメモリ制約の下でも，比較的高度なプログラムの開発が可能になり，成人にとっても取り組む価値のある知的な娯楽を提供できる．
- プログラミング言語自身は簡潔であり，プログラミングの初心者にも比較的容易に習得し，制御プログラムを開発することができる．
- プログラミング入門の適切な教材が提供できる．
- さまざまな部品を組み合わせて自由な発想でロボットを製作することが容易となり，ロボット工学などの教材としても利用しやすくなる．

6．普及（または活用）の見通し

開発したシステムは，オープンソースのソフトウェアとして，下記の URL から無償で公開している．今後，プログラミングや機械工学の教材として活用していく予定である．すでに，国内外のいくつかの機関から，教材として検討を開始しているとの連絡を受けている．現在は，関連ドキュメントとしては，リファランスマニュアル（英文），設定ガイド(英文)，発表論文（英文および和文），発表用スライド（英文）を公開しているが，教育用の教材として有効利用するためには，適切なテキストブックが必要であり，海外の教育･研究者も交えたコミュニティで，教育利用の経験をふまえながら，検討を進めていきたい．

7．開発者名

湯淺 太一（京都大学 大学院情報学研究科 yuasa@kuis.kyoto-u.ac.jp）

（参考）開発者 URL

http://www.xslisp.com