

環境反応型イルミネーション装置の開発 ～関数型プログラミングをサポートする組み込み開発フレームワークの開発～

材料技術室 石川 隆朗

Development of Illumination Devices Reacting Environment Condition ～ Development Embedded Framework Supporting Functional Programming ～

Takaaki ISHIKAWA

音、光等の周囲の環境情報を様々なセンサで読み取り、それらに対してインテリジェントな反応を示すイルミネーション装置の開発を目的とした関数型言語を用いた組み込みマイクロプロセッサ開発用フレームワークの作製を行った。リバーシ・ゲーム・イルミネーション等の試作イルミネーションを作成し高い開発効率を持つことを示した。

1. はじめに

デパートのショーウィンドウには様々なイルミネーション装置が飾られており、我々の目を楽ませてもらえる。また、観光地はイルミネーションによりライトアップされ、イルミネーション自体が観光の目玉となる場合も多い。このようにイルミネーションは非常に市場性の高い商品である。

従来のイルミネーション装置はただ単純に発光するだけか、繰り返し点滅などの単調な動作を行うものであった。近年、PC等電子機器の発達により、音や光、温度等の周囲の環境情報を読み取りインテリジェントな反応を示すイルミネーション装置も増えてきている。それらは一般的にPCのような大きく大電力を要求する機器によって制御されている。イルミネーションの制御装置はそれ自身が目立たないよう小型で、電源が無くても電池で動くよう省電力で、24時間風雪に耐え自立稼働することが望ましい。これらの要求は組み込み用マイクロプロセッサを用いることで解決される。

一般的に組み込みマイクロプロセッサ開発はC/C++言語を用いて開発される。C言語はGC (Garbage Collection)がないなどプログラミングを容易に行うための機構がなく、プログラム開発は困難である。

イルミネーション装置のようなアイデアが重要視される商品は、試作品を作っては改良し、さらに新しいバージョンの試作品を作成しさらに完成度を上げていくサイクルを繰り返すラピッドプロトタイプ開発が有用である。しかしC言語はこのような開発を行う柔軟性に乏しい。

また、モーターやサーボ等を制御する際にプログラムが誤動作を起こすとハードウェアの損傷につながる。そのためバグが発生しにくいプログラミングが必要である。

関数型プログラミングは1930年代のラムダ計算の考案、1958年のLISPの登場など古くからの歴史があり、バグの発生しにくい頑健なプログラミングと柔軟な記述を両立している。

本研究では関数型プログラミングの頑健性、記述力の高さに注目し、組み込みマイクロプロセッサ開発を関数型プログラミング言語で行うことができるフレームワークを開発した。開発したフレームワークを用い試作イルミネーション装置を作成し本フレームワークの有効性を調査した。

本フレームワークはインテリジェントなイルミネーション装置をターゲットとしたが、一般的なIoT機器の開発にも利用できる。

2. 関数型言語フレームワーク

以下の特長を持つ関数型言語処理系をルネサスエレクトロニクス社製マイクロプロセッサRXシリーズを搭載したマイコンボード上に実装した。

2.1 関数型言語処理系

頑健かつ柔軟なプログラムの記述を行う上で以下の要素が必要だと考えた。

- クロージャを持ち、それが第一級のオブジェクトであること
- レキシカル・スコープを持つこと
- GCを持つこと

クロージャとは関数(動作)とデータの組である。第一級のオブジェクトとは変数として束縛, 名前で指し示して利用することが可能なオブジェクトのことである。クロージャが第一級のオブジェクトであることによって高階関数が利用できる。また, データに動作を紐付けることによってオブジェクト指向的な考えでプログラムを作成することが可能になるなど, プログラムの記述力に柔軟性が増す。

レキシカル・スコープは変数の定義領域, 指し示す内容が単純かつ明確にわかり, 変数内容の誤認が起こりづらく頑健なプログラミングに有効である。

GCがあることによりオブジェクトは永続的に存在するように見え, プログラムはオブジェクトの寿命を考慮せずプログラムを記述することができる。これによって, 容易にプログラムを記述だけでなく頑健に作成できる。

これらの特長を持つプログラミング言語として Scheme 言語に着目し, Scheme 言語の subset の処理系を開発した。Atsushi Moriwaki の mini-scheme と同等の動作を行う VM (Virtual Machine) を開発した。その VM 上で動作するバイトコード列を生成する Scheme コンパイラを Haskell 言語で記述した。

2.2 OS機能

OSのない組み込みマイクロプロセッサ上で動作させるためフレームワーク中にOS機能を含めた。

複数の仕事を並行して行うためマルチタスク機能を付与した。

本フレームワークのマルチタスク機能は Unix や Mac OS X, Windows 等で採用されているプリエンプティブ・マルチタスクではなく, 協調型マルチタスクを採用した。

OSが実行タスクから強制的に制御を奪うプリエンプティブ・マルチタスクはCPUのレジスタ, プログラムカウンタ等をタスクごとに記録, 保持しなければいけないためコストが大きい。協調型マルチタスクは保持しなければならないコンテキストが少ないためコストが小さくなる。

協調型マルチタスクはタスクがOSに実行権を交換しない場合, 並行動作を行うことができないが, 本フレームワークではVMが強制的に実行権を奪うため粒度の小さいマルチタスクが可能である。

本フレームワークの処理系は mark and sweep 法の GC を備えている。

C 言語を利用する組み込み OS はデータの保持にメモリのアドレスを用いたスタック・データ構造を使う。このスタックはタスク一つ一つに対して個別に必要で, 各タスクがどれだけのスタック・メモリ容量が必要かの見積もりは困難である。

本フレームワークは全てのタスクが一つの GC メモリ領域を利用するため, 個別のタスクのメモリ使用量を気にせずプロセッサのメモリ全てを利用することができる。

PWM 制御やタイマ等のマイクロプロセッサ周辺装置の利用も可能である。

3 試作イルミネーション装置

3.1 5-queenイルミネーション

n-queen は上下左右斜め方向に自由に動けるチェスのクイーン n 個を $n \times n$ の盤上にお互いの効き筋にいないように配置できる盤面を探索するもので, $n=8$ としたエイト・クイーンが有名である。

$n=5$ とした 5-queen のソルバ(解法器)イルミネーションを本フレームワーク上に実装した。(写真 1)

本イルミネーションは 5-queen の解を LED マトリクス上に表示するものである。タクトスイッチにより解の巡回表示を行う。

このイルミネーションの動作には以下の機能が必要となる。

- 5-queen のソルバ
- LED マトリクスのダイナミック点灯
- チャタリング除去を行ったスイッチ入力

これらの動作を時分割の並行動作で行う。これら全ての機能を 130 行で記述することができた。

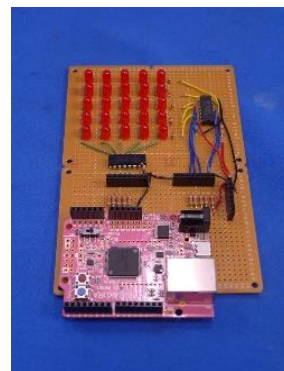


写真 1 5-queenイルミネーション

3.2 リバーシ・イルミネーション

8×8のフルカラーLEDマトリクス上で動作を行うリバーシ・イルミネーションを作成した。(写真2)

リバーシの解法アルゴリズムはミニ・マックス法を採用した。コンピュータの思考中にタクトスイッチによりカーソルの移動ができる。

5-queenイルミネーションと同様に

- ・リバーシのソルバ
 - ・フルカラーLEDマトリクスのダイナミック点灯
 - ・チャタリング除去を行ったスイッチ入力
- の並行動作を行う。

以上の機能を500行のプログラムで記述できた。開発時間は8時間程度であった。

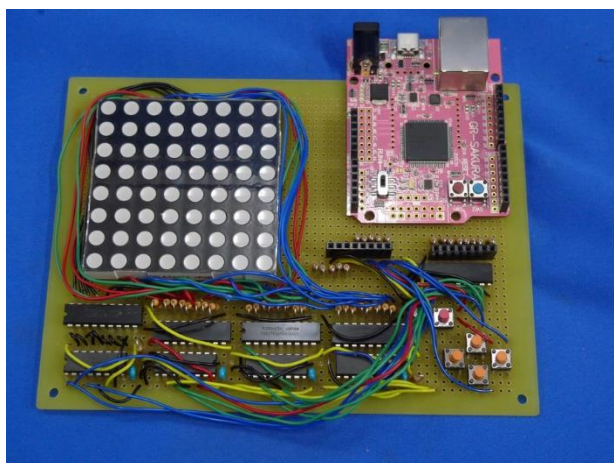


写真2 リバーシ・イルミネーション

3.3 お茶運び自動人形

エレキギターの音声で動作するお茶運び自動人形を作成した。(写真3) エレキギターの音程の高低で前進、旋回の動作を行う。



写真3 お茶運び自動人形

3.4 PCで操作する模型自動車

PCで操作する模型自動車を作成した。(写真4) PCはシリアル通信で制御を行う。車体は1/10の電動RCカーを利用している。

サーボ、スピードコントローラのPWM制御、シリアル通信はマイクロプロセッサの周辺機器を利用し、ハードウェアで制御を行っている。

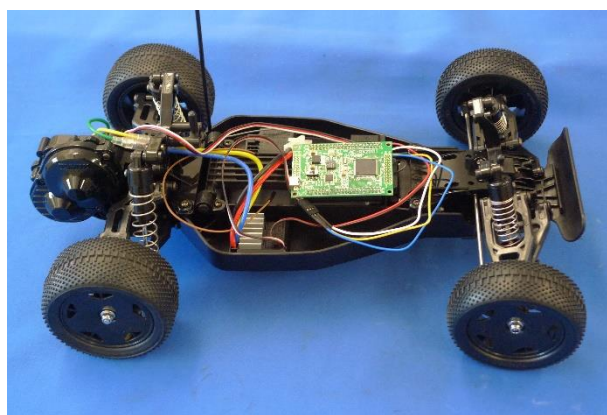


写真4 PCで操作する模型自動車

4. まとめ

音等の周囲の環境情報を読み取りインテリジェントな反応を示すイルミネーション装置を開発するための関数型言語を用いた組み込みプロセッサ開発フレームワークの作製を行った。また、そのフレームワークの有用性を確認するために様々な試作イルミネーション装置の作製を行った。

本フレームワークを使用することによりリバーシ・ゲームのAIが8時間で開発できることが確認された。本フレームワークを用いることにより、組み込みマイクロプロセッサ開発が効率よく行うことができ、ラピッドプロトタイピング開発を迅速に行うことができることが期待される。

本フレームワークはインテリジェントなイルミネーション装置の開発に主眼を置いたが、今後IoT機器の開発への利用を行うことも検討している。