

関数型言語を用いた組み込みマイクロプロセッサ用フレームワークの開発

材料技術室 石川隆朗

Development of Framework for Embedded Microprocessor
Using Functional Programming Language.

Takaaki ISHIKAWA

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のフレームワークの開発を行っている。平成 28 年度から平成 30 年度までの 3 カ年の研究であり、本年度はマルチタスク機能の開発を行った。Common Lisp 上の Bordeaux threads 上に関数型プログラミングに適したメッセージ通信機構を構築した。応用例として後出しジャンケンゲームの作成を行い、有用性の確認を行った。

1. はじめに

IoT(Internet of Things)に対する注目が高まっている。IoT は様々なセンサを利用し情報を集め、様々な行為を行う。その中で、組み込みマイクロプロセッサは重要な地位を占めている。今後、IoT 機器の発展に伴い、組み込みマイクロプロセッサ上で AI 等の複雑なプログラムを動作させる需要が大きくなることが予想される。

組み込みマイクロプロセッサ開発は一般的に C 及び C++言語によって行われる。これらの言語は GC 等現代的な言語が持つ機構を備えていないため開発効率が高いとは言えない。そこで、平成 25 年度から 27 年度まで「GC(Garbage Collector)を備えた高級言語による組み込みマイクロプロセッサの開発フレームワークの作成」を行い、開発の効率性を高めることに成功した。

さらなる効率化を求めるにあたり、プログラム開発におけるデバッグにかかる時間を短縮することを考えた。プログラム開発におけるデバッグ時間の割合は非常に大きい。この時間を減らすことでプログラミングに関わる時間は大いに短縮できると考える。加えて、組込み開発はモーターの誤動作、配線のショート等の事故を起こすためバグのない信頼性の高いプログラムの開発は重要である。

一般にプログラミング言語の堅牢性と記述力の高さはトレードオフの関係にある。関数型プログラミングはプログラムを構成する要素とし

て関数を中心として考える手法である。その歴史は長く、半世紀以上も前からプログラムの記述法に関する研究が行われており、プログラミング言語の堅牢性を保ったまま記述性を高めることに成功している。

本研究では組み込みマイクロプロセッサ上で動作する関数型プログラミング言語処理系を作成する。また、マルチタスク制御等、OS 機構を含むフレームワークを開発する。

本年度は開発するフレームワークで用いるマルチタスク機構のメッセージ手法の選定を行い、PC 上で動作する Common Lisp 上の Bordeaux threads 上に実装を行い評価した。

2. 過去の成果物

「GC を備えた高級言語による組み込みマイクロプロセッサ開発フレームワークの作成」では Scheme 言語のサブセット言語処理系を作成し、ルネサスエレクトロニクス株式会社製 RX シリーズマイクロプロセッサ上に実装した。これにはマルチタスキング等の OS 機構もフレームワークとして組み込まれている。

作成した処理系を用い様々な作例を作成した。6-queen のソルバ、リバーシゲームのソルバ(写真 1)、ギターの声で動作するお茶運び人形が作成された。リバーシゲームは 8×8 のフルカラー LED をダイナミック点灯で動作させ、LED の点灯、キー入力によるカーソルの移動は、マイク

ロプロセッサが次の手を考えている間も同時並行的に動作する。これら全ての動作が500行程度のプログラムで記述できた。ソフトウェアの作成にかかった時間は8時間程度であった。ギターで動作するお茶運びロボットは数十行で記述できソフトウェア作成にかかった時間は1時間程度であった。

非常に効率よくソフトウェアを作成できることが実証された。

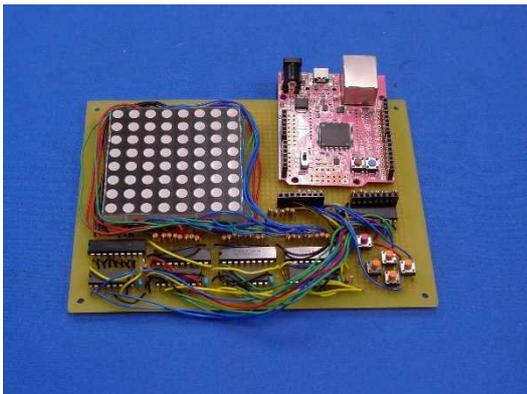


写真1 リバーシゲーム

3. タスク間通信機構の選定

作成した作例、特にリバーシゲームが容易に作成できた理由として、マルチタスク制御が簡単に記述できたためだと考える。

そこで、本年度はマルチタスク制御の情報共有機構に注目し、組み込み開発向け関数型プログラミング言語に適した。情報共有機構の選定を行った。

本処理系では Erlang 言語のメッセージパッシング機構を利用した。メッセージ通信機構を Haskell 言語の MVar 機構と同等の機構を用い実装した。実装に用いた atomic 制御機構、MVar 機構はメッセージパッシング機構とは独立してプログラミング言語上で利用できるようにした。メッセージパッシングという名前から、オブジェクト指向のメッセージパッシングが想像できるが、同様の表現ができるよう、タスクオブジェクトにメッセージを送信するという形の構文を用いた。

PC 上の SBCL (Steel Bank Common Lisp) 上の Bordeaux thread 上に実装した。

4. 作例

作例として「後出しじゃんけんゲーム」を作成した。後出しじゃんけんゲームは、2人用のゲームであり、二人は親と子に分かれる。「じゃん」「けん」「ぽん」「ぽん」のリズムでゲームを行う。最初の「ぽん」で親が「グー」「チョキ」「パー」のいずれかを出す。子は2回目の「ぽん」で親の出した手に負ける手を出す。リズムよく負ける手を出せなかったら子の負けとなる。テンポは少しずつ速くなっていく。

PC が親となり、人間がプレイヤーとなるものを作成した。

このゲームは時々刻々と状態が変化し、それに伴い行わなければいけない動作も変わっていく。それらの動作をうまく分離して記述できないとプログラムの見通しが悪いものとなる。関数型プログラミング手法を用いることによってその問題を解決することができた。本プログラムは元々図 1(a) に示すブロック図のようなメッセージ通信を行うよう記述していったのだが、最終的には図 1(b) のような形になった。このような構造の変化にも容易に対応することができた。このことは製品の最終的な形態を初期に規定せず開発を進めるラピッドプロトotyping 開発に非常に有用である。

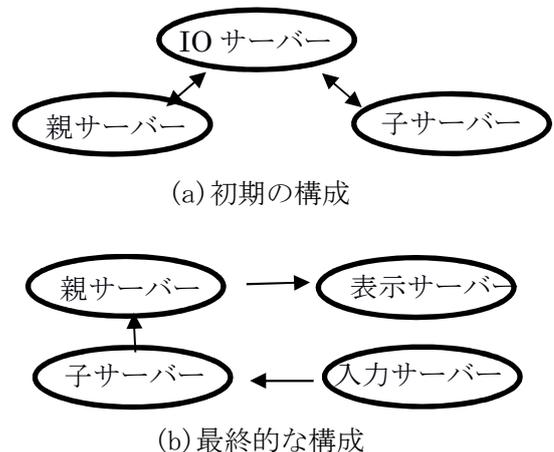


図1 メッセージ通信の変遷

5. まとめ

組み込み開発用関数型プログラミング開発フレームワークを開発するに当たって、本年度は、マルチタスキング機構の情報共有機構に着目し、検討を行った。PC 上にマルチタスク機構を構築し、作例として後出しじゃんけんゲームを作成した。