

フラッシュ ROM 上で動作する組み込み OS の動的更新方法

石井 光 横山 孝典 志田 晃一郎 兪 明連

東京都市大学

1 はじめに

近年、組み込みシステムの高機能化に伴い、組み込みシステムに求められる処理が複雑になり、組み込みコンピュータに搭載されるソフトウェアの規模が増大している。そのため、製品出荷後にソフトウェアの不具合に対応したり、機能を追加したいという要求が増えている。

これまで、自動車や家電などワンチップマイクロコントローラを用いた組み込みシステムの多くは、書き換え不可能なマスク ROM を使用していたため、ソフトウェアを修正するには機器の回収が必要だった。しかし最近では、組み込みシステムに書き換え可能なフラッシュ ROM が用いられるようになり、機器を回収することなくネットワークを介してソフトウェアを更新することが期待されている。

フラッシュ ROM 上のソフトウェアを更新するシステムとして RLL[1]が提案されている。RLL は稼働中の組み込みシステムのソフトウェアをネットワークを通して更新することができる。

しかし、RLL が対象としているのは、内蔵 ROM のみでなく外付けフラッシュ ROM をも搭載したシステムである。具体的には、内蔵 ROM に OS と RLL Agent を搭載し、外付けフラッシュ ROM にアプリケーションを搭載する。書き換え可能なのは外付けフラッシュ ROM 上のアプリケーションのみで、OS の書き換えはできない。また、内蔵 ROM、RAM のみで動作するワンチップマイクロコントローラに適用するのは困難である。

そこで、我々はワンチップマイクロコントローラにも適用可能な OS の更新方法の開発に着手した。更新を行うプログラムを OS とは独立に動作可能とするとともに、更新プログラムを RAM 上で動作させることにより、内蔵フラッシュ ROM 上の OS を書き換え可能とした[2]。

しかし、当初開発したシステムでは OS の一部を変更する場合でも OS 記憶領域全体の書き換えを必

要とした。この問題を解決するため、本論文では、OS の記憶領域のうち必要最小限の領域のみの書き換えで対応可能な、OS の部分更新手法を提案する。

2 全体構成

全体構成を図 1 に示す。本システムは、サーバ(ホストコンピュータ)とターゲット(組み込みシステム)から成る。サーバ側に転送プログラムと組み込み OS の更新データを置く。ターゲット側には更新プログラムを置く。

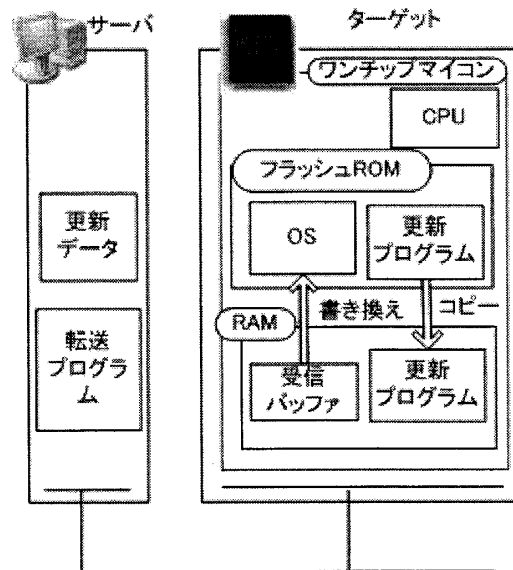


図 1 全体構成図

3 更新プログラム

更新プログラムは、ローダ、通信モジュール、書き換えモジュールから構成される。それらの機能を以下に示す。

(1) ローダ

更新プログラムを RAM 上で動作させるため、フラッシュ ROM 上に記憶されていた更新プログラムを RAM にコピーする必要がある。その処理をローダで行う。

(2) 通信モジュール

ネットワークを介してデータを送受信するための通信処理を行う。更新データを一時的に蓄える受信バッファは RAM 上に確保する。

(3) 書き換えモジュール

OS の書き換えを行うモジュールである。受信バッファに格納されたデータをフラッシュ ROM 上の指定された番地へ書き込む。

4 OS の構成

OS を部分的に更新可能とするため、OS のプログラムを複数のセクションに分け、それをフラッシュメモリの各ブロックに割り当てる。また、書き換えを行っても新たにアドレス解決をする必要をなくするため、OS を構成する関数の呼び出しを、ジャンプテーブルを介した関節呼び出しとする。以下その詳細を説明する。

(1) セクション分け

フラッシュ ROM 上のデータの消去は決められたブロックと呼ばれる単位でしか行うことができない。そこで、図 2 に示すように、OS を構成する関数を各ブロックに対応させたセクションに分けて配置し、ブロック単位で書き換えるようにする。

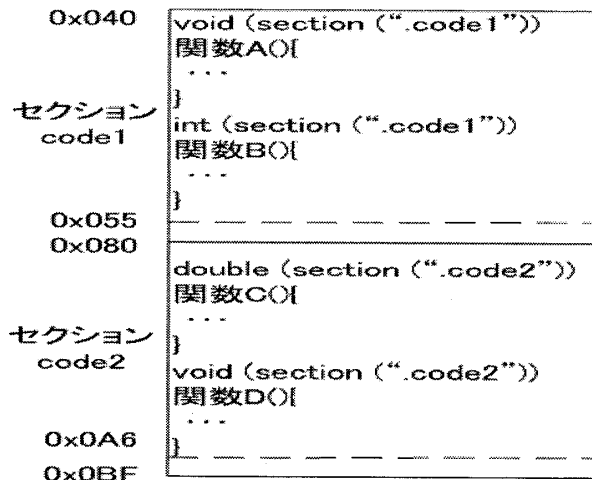


図 2 セクション分け

(2) ジャンプテーブル

更新により関数のアドレスが変化すると、関数の呼び出し側のコードについてもアドレス解決をやり直す必要があるため、その更新を行わなければならない。そこで、ジャンプテーブルを用いた間接呼び出しを採用することで、呼び出し側のアドレス解決のやり直しを不要とする。

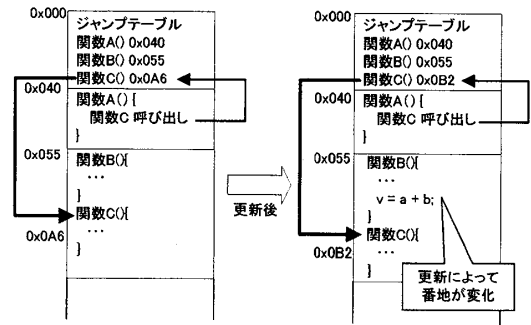


図 3 ジャンプテーブル

ジャンプテーブルを図 3 に示す。ジャンプテーブルは呼び出す関数のアドレスにジャンプする命令を記憶する。そして、関数呼び出し側は、ジャンプテーブルのアドレスを呼び出すようにする。更新時には、ジャンプテーブルのジャンプ先アドレスを修正するのみのみでよく、呼び出し側の修正は不要になる。

(3) 部分更新

セクション分けとジャンプテーブルを採用することにより、OS を部分的に更新することが可能になる。すなわち、図 4 に示すように、複数あるセクションのうち更新対象関数を含むセクションに対応するブロックと、ジャンプテーブルを記憶しているブロックのみを更新すればよい。

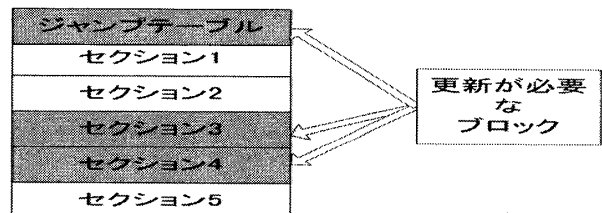


図 4 部分更新

5 まとめ

フラッシュ ROM を搭載したワンチップマイクロコントローラ向けの OS の更新手法を提案した。本手法は、フラッシュ ROM 上の組み込み OS をオンラインで部分的に書き換え可能としている。

参考文献

[1] Remote Link Loader
<http://www.toppers.jp/rll-download.html>
 [2] 樋上真人, 横山孝典, 志田晃一郎, 兪明連,
 動的更新が可能な組み込み OS に関する研究,
 情報処理学会第 71 回全国大会論文集, 1L-3,
 2009