

組み込みシステムにおけるフラッシュメモリのアクセス性能評価

井邊 研吾[†] 落合 真一[†]
 三菱電機株式会社 情報技術総合研究所[†]

1. はじめに

組み込み機器の高性能化に伴い、OS も高機能な Linux が搭載されることが増えてきている。また、組み込み機器において、OS 及び、データ保存先として、信頼が高い NOR 型フラッシュメモリを用いる場合が多い。このようなことから、Linux のフラッシュメモリへの書き込み性能について検証を行った。

検証は、低機能なソフトウェアとしてのブートローダである u-boot 上からの書き込みと Linux 上からの書き込みの性能を比較して行った。その結果、ブートローダと OS の作りの違いを考慮し、実質の書き込み時間のみを比較したところ、64MB の書き込みに、7 秒もの差があることが分かった。本稿では、この性能差について述べ、その改善策を記述する。

2. NOR 型フラッシュメモリ

NOR 型フラッシュメモリは、NAND 型フラッシュメモリと比較し、1 バイト単位での読み書きが可能という特徴があり、フラッシュメモリ上でプログラムが動作できることから、HDD 等を搭載できない組み込み機器でよく用いられる。

フラッシュメモリのデータの書き込みは、フラッシュメモリが、レディー状態の時に書き込み用のコマンドを入力し、データを書き込み、書き込みが完了するまで待つという一連の処理となる。

今回用いた検証に用いた NOR 型フラッシュメモリの仕様は以下である。

表 1 NOR 型フラッシュメモリ仕様

flash size	128Mbyte
Programming Buffer size	512byte
Sector Erase size	128Kbyte
command type	AMD

3. 本論文の目的

組み込み機器は、データをフラッシュメモリに書き込むことが多くあり、フラッシュメモリへの書き込みの性能向上が求められている。このため、Linux 上からのフラッシュメモリへの書き込み速度を向上させることを目的としている。

4. u-boot と Linux の MTD ドライバ比較

Linux と同様に u-boot にも、フラッシュメモリに書き込む機能を提供しており、この 2 つを比較するため、両方の書き込み処理について調査した。u-boot と Linux とともに、フラッシュメモリの操作を

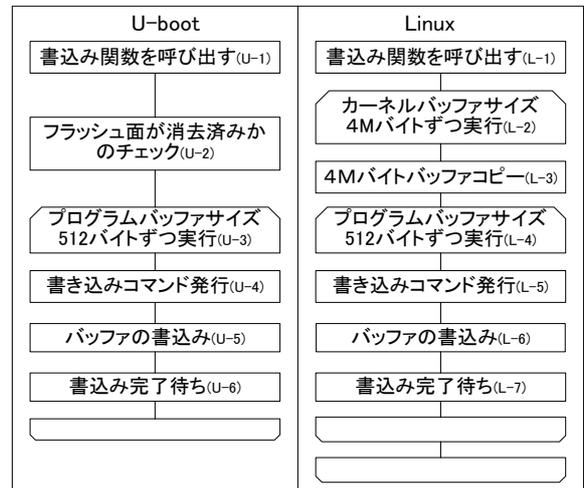


図 1 各 MTD ドライバ処理フロー

するため MTD (Memory Technology Device) ドライバを使用しており、このドライバの構造を図 1 に示す。u-boot は書き込み関数が呼ばれる (図 1 U-1) と書き込み前にフラッシュメモリが消去済みのチェックが行われている (図 1 U-2)。その後、フラッシュメモリの書き込み手順に従ってデータの書き込みを行っている。一方、Linux は、書き込み関数が呼ばれる (図 1 U-1) と消去済みのチェックは行っていないが、Linux 特有のユーザ空間からカーネル空間へのデータコピーが 4M バイトずつ行われる (図 1 L-3) 違いがある。今回、図 1 U-2 の消去済みのチェック部分はフラッシュメモリへの書き込みには関係がないため除外し、残りの各項目を比較し、Linux の高速化が実現できないかを検討する。

5. MTD ドライバ書き込み性能評価

5.1. 測定環境

本評価で用いた H/W 環境及び、S/W 環境を表 2、表 3 に記載する。本評価では、消去済みのフラッシュメモリに 64M バイトのデータを書き込む。書き込みには、u-boot 及び、Linux は既存のコマンドを用いて、フラッシュメモリへの書き込みを実施する。u-boot では、メモリ上に書き込みサイズ分を「0」で埋められた領域を u-boot の cp コマンドでフラッシュメモリに書込んでいる。また、Linux では、Linux 上からフラッシュメモリを操作できる mtd_debug コマンドを用いて、書き込みサイズ分「0」が記述されたファイルを読み込みフラッシュメモリに書き込みを実施する。この各コマンドで書き込み関数 (図 1U-1 L-1) を呼び出している。

なお、u-boot、Linux とともに、メモリ上で動作させており、フラッシュメモリへの影響はない。

Performance Tuning by Flash Memory Partitioning in Embedded Systems
 Kengo Ibe, Shinichi Ochiai
[†] Information Technology R&D Center, Mitsubishi Electric Corporation

表 2 H/W 評価環境

CPU	Freescale PowerPC 800MH シングルコア
メモリ	DDR3 512Mbyte
NOR フラッシュ メモリ	128Mbyte 評価では 64Mbyte のみ使用

表 3 S/W 評価環境

u-boot	2013.04 版
kernel	3.4.51

5.2. 測定方法

図 1 の (U-4, 5, 6) 及び (L-3, 5, 6, 7) の各項目の開始と終了時に時間を記録し、その差分によって処理時間を計測している。取得する時間は、PowerPC に備えられているクロックサイクルでカウントアップする Time Base Register (TBR) を用いて、測定している。今回、供給されるクロックサイクルは 33.3Mhz のため μ 秒精度の測定が可能である。

5.3. 測定結果

u-boot 及び Linux の書き込み時間の測定結果を表 4 表 5 に示す。64MB の書き込みにかかる時間は u-boot で 36 秒、Linux で 43 秒であり、7 秒も差があることが確認できた。各処理の実行時間を比較したところ、書き込みコマンド発行 (U-4) (L-5) 及び、プログラムバッファへの書き込み (U-5) (U-6) は実行時間差がなく、一方で、書き込み完了待ち (U-6) (L-7) には差があり、これが u-boot 及び Linux の書き込み時間の 7 秒の差として現れていることが確認できた。また、(U-6) (L-7) の 2 つの測定結果にも特徴があり、u-boot (U-6) は 170~226 μ 秒のばらつきがあったのに対し、Linux (L-7) では、253 μ 秒の一定であった。

6. Linux の MTD ドライバの性能向上

6.1. 測定結果の考察

5.3 節の結果、(U-6) (L-7) の書き込み完了待ちに差があることが分かったため、書き込み完了待ちの実装方式を確認したところ、u-boot と Linux では書き込み完了の待ち方の違いがあることを確認できた。フラッシュメモリはトグルビットと呼ばれる書き込み完了前にリードすると反転するビットがあり、これによって、書き込み完了を検知することができる。

u-boot では、このトグルビットを 2 回連続で取得し同一の値になれば、完了待ち状態から抜ける方式を取っているため、フラッシュメモリの完了状況によって、(U-6) の待ち時間が変動している。

一方、Linux では、u-boot と同様にトグルビットを確認する方法も用いているが、その前に、一定の udelay を用いた待ちを設けており、この待ち時間が u-boot で待っている最大の 226 μ 秒より長い場合、測定結果に表れている。この一定の待ち時間は、フラッシュメモリの記述されている待ち時間を元に行っているもので、今回の設定値は 256 μ 秒であった。実測と 3 μ 程度の差があるが、この設定値が (L-7) の 1 回の実行時間として現れていると考えられる。

表 4 u-boot の測定結果

項目	1 回の実行時間	実行回数	合計
U-4	1 μ 秒	131072 回 64Mbyte/512byte	131m 秒
U-5	73 μ 秒	131072 回 64Mbyte/512byte	9.6 秒
U-6	平均 200 μ 秒	131072 回 64Mbyte/512byte	26.2 秒
合計	-	-	35.9 秒

表 5 Linux の測定結果

項目	1 回の実行時間	実行回数	合計
L-3	平均 11m 秒	16 回 64Mbyte/4Mbyte	176m 秒
L-5	1 μ 秒	131072 回 64Mbyte/512byte	131m 秒
L-6	73 μ 秒	131072 回 64Mbyte/512byte	9.6 秒
L-7	253 μ 秒	131072 回 64Mbyte/512byte	33.3 秒
合計	-	-	43.2 秒

表 6 Linux 改善後の測定結果

項目	1 回の実行時間	実行回数	合計
L-3	平均 11m 秒	16 回 64Mbyte/4Mbyte	176m 秒
L-5	1 μ 秒	131072 回 64Mbyte/512byte	131m 秒
L-6	73 μ 秒	131072 回 64Mbyte/512byte	9.6 秒
L-7	168 μ 秒 + 平均 37 μ 秒	131072 回 64Mbyte/512byte	26.8 秒
合計	-	-	36.7 秒

6.2. 改善策

6.1 節の考察の結果、Linux の書き込み性能が、u-boot よりも遅い原因は、固定値の待ち時間が、実際のフラッシュメモリの書き込み完了よりも長い設定になっているためである。今回この固定値の設定値を実測値に近づけ短くすることで、Linux の書き込み性能が改善できると考えた。そこで、256 μ 秒の固定値を u-boot で測定した待ち時間の最小に近い 170 μ 秒にし、その後、u-boot と同様にトグルビットによる待ちを行うように変更を加えた。

6.3. 改善結果

6.2 節の変更後再測定を行った結果を表 6 に示す。固定の待ち時間の設定 170 μ 秒に対し、実測では 168 μ 秒であり、トグルビットによる待ち時間は平均 37 μ 秒であった (表 6 L-7)。この値は、u-boot の待ち時間 (U-6) と同程度である。全体の書き込み時間も 43.2 秒から 36.7 秒へ 6.5 秒短縮しており、性能が改善したことが確認できる。

7. おわりに

本稿では、Linux の MTD ドライバに u-boot の実装方式を適用することで性能が向上することを示した。この完了待ちはフラッシュメモリの消去でも発生するため、消去処理への適用の検討を実施する予定である。