特集号
招待論文

スマートフォン省電力化への取り組みと成果
―アプリケーション省電力化への誘い―

丸木 勝也[†1]　曽根田 裕郁[†1]　日原 圭祐[†1]　陣之内 宏基[†1]　江角 直起[†1]

[†1] シャープ（株）

ユーザが Android スマートフォン（以降, スマートフォン）を購入するときに重視する項目として, 電池持ちが挙げられる. 筆者らは電池持ちを長くするために, 低電力で表示可能な IGZO 液晶を搭載したスマートフォンを開発した. また, スマートフォン購入後も電池持ちが長いことをユーザに実感してもらうため, アプリケーションの動作を調査し, 消費電流の改善を行った. アプリケーションのどのような動作が端末全体の電池持ちに大きく影響するのかを述べる.

1. はじめに

MMD研究所の調査によると，スマートフォン購入時に重視する項目として，電池持ち（40.9%）が挙げられる[1]. これは，スマートフォンの電池持ちに関してユーザが満足していないことの裏返しと考えられる. 筆者らは低電力な周辺装置（以降，デバイス）を搭載し，最適なソフトウェア制御を行った電池持ちが長いスマートフォンを提供することで，ユーザの満足度を向上させようと考えた.

また，スマートフォンを購入したユーザは，アプリケーションを自由にダウンロードして使用することができる. 購入後も電池持ちが長いことをユーザに実感してもらうためには，アプリケーションの動作が電池持ちにどの程度影響を与えているか知る必要がある. しかし，それらは目に見えるものではないため，特殊な調査を必要とする.

本稿では，まず，当社スマートフォンの電池持ちを改善するアプローチについて説明する. 次に，電池持ちを長くするために，アプリケーション開発者と行っている取り組みを紹介する.

スマートフォンユーザの
電池持ちに対する不満を
解消したい

2. 電池持ちを改善するための設計思想

電池持ちを改善するためには, どこに問題があるのか, 体系的に整理する必要がある. 筆者らは，電池持ちの改善には，ハードウェアデザイン，システムアーキテクチャ，アプリケーションの調査といった積み上げが必要と考えた（図1）.

はじめに，スマートフォン開発の上流工程にあるハードウェアデザインは，電池持ちを改善するためにシステム全体の消費電流を考慮し，最適な低電力デバイスを選択する. 選択するデバイスによって，消費電流は大きく変動し，選択するポイントとして，デバイス動作中の消費電流，起動時間，終了時間，電力効率，温度特性などが挙げられる.

次に，実装された低電力デバイスを正しく制御するためのシステムアーキテクチャが必要である.「電池持ちの長いスマートフォン」を実現するためには,

- ユーザがスマートフォンを操作するとき，実装したデバイスを一定時間以内に応答させること

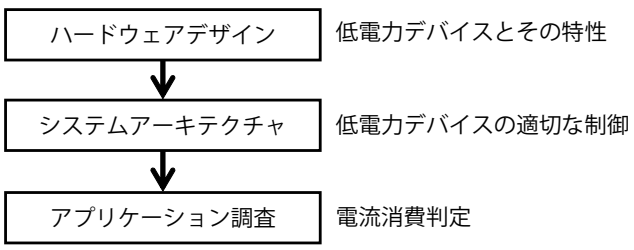


図 1　電池持ちを改善するための設計思想

• ユーザがスマートフォンを操作していないとき，必要なデバイスのみ動作させること

上記2点が必要であり，第3章に詳細を説明する．

最後に，アプリケーションの動作次第で，電池持ちが大きく左右されるため，アプリケーションの動作を調査する際に筆者らがどういった取り組みを行っているか，第4章に説明する．

3. IGZO 液晶とデバイス制御による省電力化

液晶ディスプレイとCPUの消費電流は，スマートフォン全体の消費電流に占める割合が高い．筆者らは，電流削減効果の高い液晶ディスプレイとCPU制御に注目した．本章では，まず，ハードウェアデザインの取り組みとして，IGZO液晶ディスプレイについて3.1節で述べる．次に，システムアーキテクチャの取り組みとして，パフォーマンスを維持しながら消費電流を削減する手段について3.2節で述べる．

3.1 IGZO について

IGZOとは, In（インジウム）, Ga（ガリウム）, Zn（亜鉛）で構成される酸化物半導体である．IGZOは，液晶ディスプレイなどに広く応用されている非結晶半導体のアモルファスシリコンに比べ，20 ～ 50倍という電子移動度の高さで電流を流すことができる．電子移動度の高さにより，IGZOを採用した液晶ディスプレイは，従来よりも薄膜トランジスタの超小型化と配線の細線化を実現でき, 1画素あたりの透過量を高めることができた. つまり，1画素あたりのバックライト透過率が向上し，従来と同じ明るさで消費電流を削減することができる．

また，IGZOは電流がOFFの状態でも一定期間データの書き換えをせずに画像を保持できる．従来の液晶ディスプレイは静止画表示中，1秒間に60回の画像データを転送していたが，IGZOは1秒間に1回の転送で済む．したがって，転送するデータ回数が少なくて済むため，電流を多く消費する液晶ディスプレイへの描画回数を減らすことができる．

また，静止画だけでなく，ワンセグ放送は画像のフレームレートが1秒間に15回，動画コンテンツは1秒間に30回であり，1秒間に60回の画像データ転送を必要としないため，消費電流を大幅に削減することが可能となる（表1）．

IGZO液晶の消費電流を削減する手段として，当社スマートフォンの設定メニューの中に，「省エネ液晶ドライブ（もしくは，「なめらかスクロール」）」を搭載した．設定を有効にすることで，1秒間に60回の画像データ転送を必要とする画面について，1秒間に30回の転送回数で動作する．1秒間に30回の転送回数でも十分快適に使うことができる場面は多く，電池持ちが長いことも実感できる．

3.2 消費電流とパフォーマンスについて

消費電流を削減するためには，CPUが動作しない時間を可能な限り増やすことが重要である．一方で，スマートフォンを快適に動作させるためには，大きな処理を実行するとき，高速なクロック周波数で動作し，複数のプロセスを並列処理可能なマルチコアCPUを必要とする．つまり，パフォーマンスを上げようとすると，CPUクロック周波数が高くなり，消費電流を多く消費する．「快適に使用できること」を実現しつつ，CPUの消費電流を削減することは容易ではない．特効薬はなく，無駄な処理を可能な限り減らし，CPUが動作しない時間を増やす地道な作業が必要である．

CPUクロック周波数を高くする要因として，IGZO液晶を含む周辺デバイスの制御，アプリケーションの動作仕様，Android OSの処理量が挙げられる．具体的には，周辺デバイスの制御にはデバイスからの要求に対して，一定時間以内の応答が求められる．アプリケーションはプログラミングに応じて，CPUクロック周波数を高く維持することが可能となる．Android OSは，OSそのものが定期的に大きな処理を実行するため，バックグラウンドでCPUクロック周波数が高くなる．これら複数の要素を考慮しながら，複数のCPUとクロック周波数を制御する（図2）．

図2で挙げている要素に対する消費電流削減例を図3に示す．ある操作を行ったとき，デバイスが動作する理想の消費電流波形をa.のデータとする．

b.のデータは操作終了後，デバイスを停止し，消費電

表1　IGZO について

IGZO は転送するデータ回数が少なくて済むため，電流を多く消費する液晶ディスプレイへの描画回数を減らすことができる

	1 秒間に転送するデータ回数	
	非 IGZO	**IGZO**
静止画	60 回	**1 回**
ワンセグ	60 回	**15 回**
動画	60 回	**30 回**

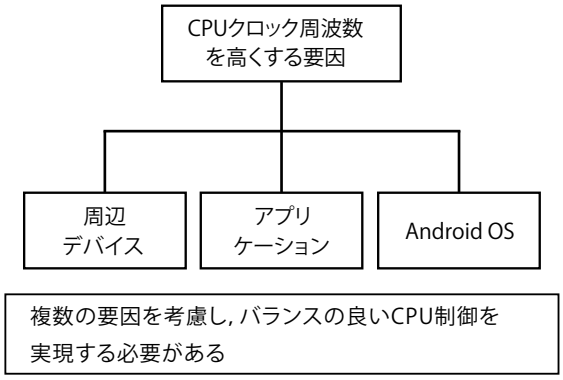


図 2　CPU クロック周波数を高くする要因について

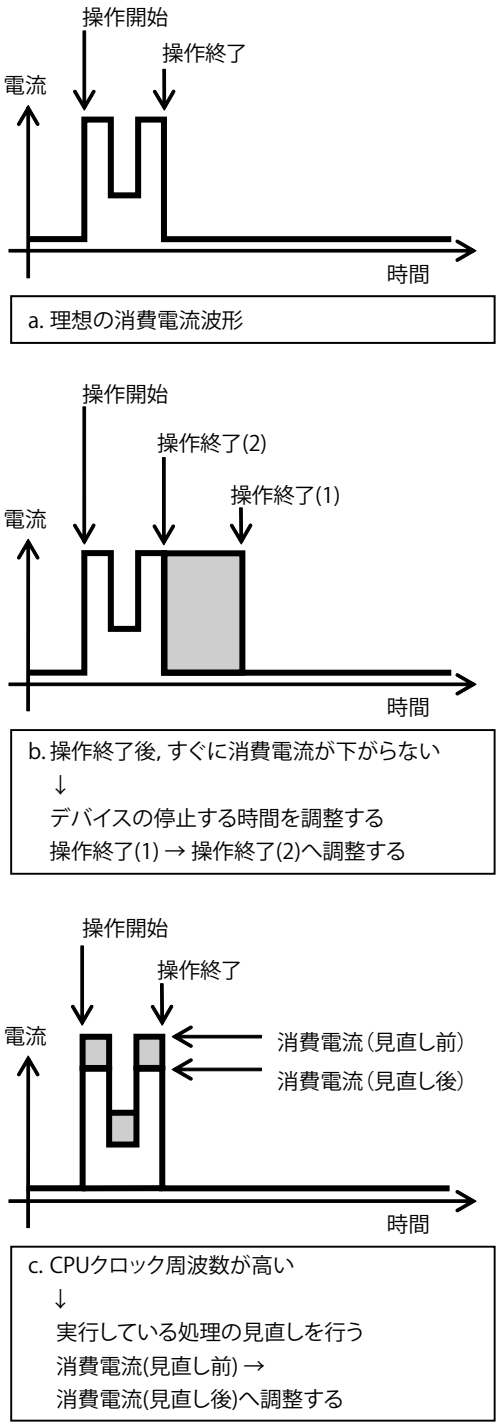


図 3　消費電流削減例

流を削減する一例である．デバイスは，連続的動作に備えて停止するまでの時間を長く設定する場合がある（図中の灰色部分）．停止するまでの時間を調整し，連続動作も可能な調整を行い，a.の理想の電流波形に近づける（操作終了(1)から操作終了(2)へ調整する）．

c.のデータは操作中，高いCPUクロック周波数で動作し続けている一例である．処理が必要なとき（図中の灰色部分）は，無駄な処理が存在するため，CPUクロック周波数が高くなり，消費電流が増加する．実行している処理の見直しを行い，a.の理想の電流波形に近づける（消費電流（見直し前）から消費電流（見直し後）へ調整する）．

4. アプリケーション開発者に対する指針

4.1　注目すべきアプリケーションの動作

スマートフォンを購入したユーザは，好きなアプリケーションを自由にダウンロードして使用することができる．購入後も電池持ちが長いことを実感してもらうために，アプリケーションの動作が電池持ちにどの程度影響を与えているか調査した．これまで，アプリケーション開発者と筆者らが同じ観点から調査するための指針がないため，まず指針を作成することを考えた．次に，作成した指針に従って，電池持ちへの影響を調査し，電流増加につながる処理が抽出されれば, 協力して対策を行う．筆者らは，この地道な取り組みを通して，当社製スマートフォンだけではなく世界中のスマートフォンの電池持ちを長くし, ユーザ満足度の向上に貢献したいと考える．

一方で，アプリケーションは，操作性を高め，より使いやすく，多様なグラフィックスデザインを使用し，より分かりやすく，ユーザのあらゆる満足度を向上させるために存在する．したがって，ユーザがアプリケーションを操作しているときの電池持ちを長くするための仕様変更は，実現したいことを損う可能性があり，アプリケーション開発者に依頼することは難しい．そこで，ユーザが使用しておらず，バックグラウンドでアプリケーションが動作しているときの挙動に注目した．調査したところ，アプリケーションによって，操作していないときでもバックグラウンドで消費電流を使用することが分かった．筆者らは，ユーザがスマートフォンを使っていないときに勝手に電池持ちが悪くなっていることに不満を感じていると考えた．そのためには，いかにバックグラウンドでの動作を少なくするかが重要であり，次節で具体的な取り組みについて説明する．

操作していないときでも バックグラウンドで電池を消費する アプリが存在する

4.2　取り組みフロー

「電池持ち改善」取り組みフローを図4に示す.

筆者らは，7つのステップにより，アプリケーション開発者とアプリケーションの省電力化に取り組んでいる.

4.2.1　7つのステップを進める前に

アプリケーションの省電力化を進める上で重要なことは，電池持ちに影響を与える処理が仕様通りの動作か開発者に確認した上で，対策案の検討を一体となって進めることである．アプリケーションの動作仕様を確認することなく，急いでアプリケーションの省電力化を進めようとした場合，省電力化は達成できたとしてもアプリケーションが提供するサービスを損なう結果に繋がってしまうこともあり得る．7つのステップを繰り返す回数に決まりはなく，地道に必要な回数を繰り返していくことが，開発者との信頼を構築し，アプリケーションの省電力化への近道になることをご理解いただきたい.

4.2.2　アプリケーションの選定

ステップ①では，調査するアプリケーションを選定する．多くのユーザへ改善効果が分かるように

- ユーザから得られた情報
- 人気の高いアプリケーションランキング情報
- 口コミ，掲示板による情報
- マスメディアで注目されている情報

上記の情報を基に選定する.

4.2.3　消費電流増加につながる処理の抽出

ステップ②では，選定したアプリケーションについて，消費電流増加につながるバックグラウンド動作を調査する．具体的には，

- アプリケーションをインストールするだけで発生する場合
- アプリケーションをインストールするだけでは発生せず，ログイン処理実行後に発生する場合

上記のような利用シーンでの1次解析を行う.

電池持ちが短くなる例を図5に示す．ある操作を行ったときの本来あるべき消費電流波形をa.のデータとする．操作開始時はCPU，必要なデバイスが有効になり，操作終了後は操作開始前と同じ状態に戻る．この状態であれば，電池持ちに影響はない.

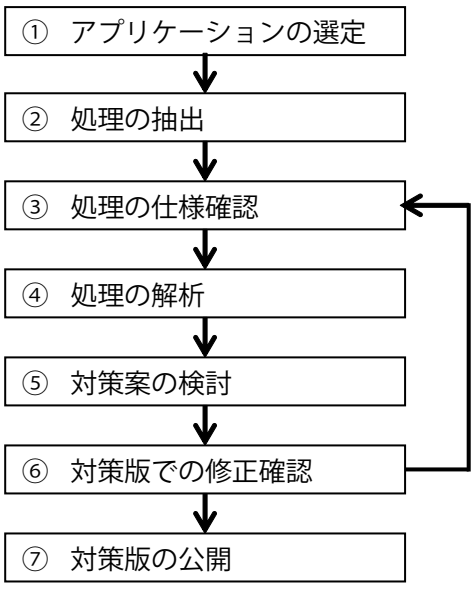


図 4 「電池持ち改善」取り組みフロー

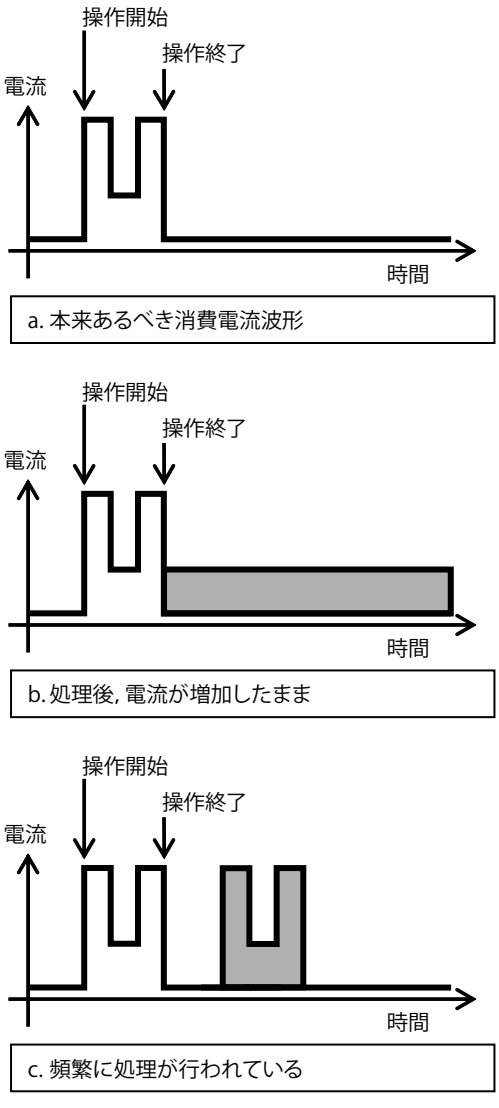


図 5　電池持ちが短くなる例

しかし，b.のデータは処理が終わっても電流が増加した状態になる一例である．灰色の部分が電池持ちが短くなる要因である．ユーザ操作終了後もアプリケーションが動作するときに必要なデバイスが有効になっている可能性があり，有効になっているデバイスの制御を調査する．

また，c.のデータは頻繁に処理が行われている一例である．ユーザ操作終了後もアプリケーションのタイマ処理が頻繁に動作している可能性があり，アプリケーションのタイマ動作の頻度，処理内容を調査する．

4.2.4　消費電流増加につながる処理の仕様確認

ステップ③では，ステップ②で調査した消費電流増加につながる処理について，アプリケーション開発者に仕様確認を行う．

筆者らはアプリケーション開発者に，

- 再現手順を伝えた上で，仕様通りの動作であるか
- どのような条件で動作する処理であるか
- 処理の目的は何か

上記の仕様確認を行う．

4.2.5　消費電流増加につながる処理の解析

ステップ④では，アプリケーション開発者が2次解析を行う．筆者らはアプリケーション開発者に対して，

- 現象が再現する手順，再現しない手順
- 動作プロセス，動作周期に関する情報

上記の連携を行う．

筆者らは対象となるアプリケーションを改修することはできないが，解析に必要な情報をアプリケーション開発者に提供できる．そして，アプリケーション開発者が消費電流を削減することはユーザ満足度を向上させると納得してもらうための協力を行う．

4.2.6　対策案の検討

ステップ⑤では，アプリケーション開発者とともに，動作仕様や消費電流への改善効果を考慮した上で対策案を検討する．

- 実現したいことを損わない改善策の提案
- 消費電流波形データを取得し影響に関する詳細連絡

上記を行い，筆者らはアプリケーション開発者へ対策案をうまく実行できるように調整する．

4.2.7　対策版アプリケーションでの修正確認

ステップ⑥では，アプリケーション開発者より対策版アプリケーションが提供され，現象が改修されているかどうかを確認する．

- ステップ⑤の対策案が実行されているか
- 消費電流の増加が改善されているか

上記について対策版で改善がみられない場合は，処理の仕様確認（4.2.4節）に戻る．

4.2.8　対策版アプリケーションの公開

ステップ⑦では，アプリケーション開発者の協力により，対策版アプリケーションが公開されるまでの進捗確認を行う．具体的には，

- 公開予定日
- 公開後，ステップ⑤の対策案が実行されているか

上記について確認する．

4.2.9　取り組みの成果

図5-cの事例に対して取り組む中で電流測定機器を使用せずに，ソフトウェアの動作状況が分かるツールを開発した．ツールは，カーネルログにタイマ動作とプロセス番号が出力されるように修正したものである．このツールをステップ③においてアプリケーション開発者に公開し，実装の動作を開発者にも確認してもらい，共通認識を持つようにした．結果として，図5-cの事例にあるような，繰り返し処理が行われることがなくなった．筆者らは7つのステップを地道に繰り返したことで，開発者に納得してもらい，アプリケーションのサービス維持と省電力化の両立を実現できた．

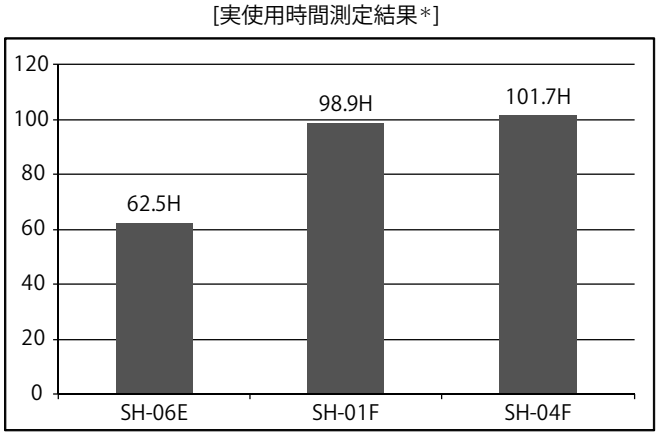
スマートフォンの電池持ちは
世界中のアプリケーション
開発者と協力，対話が必要

5. おわりに

スマートフォンの電池持ちを長くするために，筆者らは，低電力で表示可能なIGZO液晶を搭載し，CPU制御の最適化を行ったスマートフォンを開発した．また，アプリケーションの動作が電池持ちに影響を与えることを調査した．この取り組みは，1つのアプリケーションに対応するだけでも長い時間を要する．

筆者らは電池持ち向上のために，今後もハードウェア，ソフトウェアそれぞれの改善点を見つける取り組みを継続する（図6）．2013年5月に発売したSH-06E，2013年11月に発売したSH-01F，2014年5月に発売したSH-04Fは，取り組みを継続することで電池持ち時間を伸ばすことができた．また，ユーザの電池持ちに対する不満を解消するためには，アプリケーション開発者の協力が不可欠である．図7は，アプリケーション開発者と対話を行い，改善した一例として紹介する．そして何より，この

取り組みを世界中のアプリケーション開発者に知ってもらうことが重要である.



*一般に想定されるスマートフォンの利用があった場合の電池の持ち時間.

図 6 「電池持ち改善」結果

https://twitter.com/unity_japan/status/471957786767552513

> 2014/05/29 - Unity 4.5 には、Android 端末において消費電力を抑えるための、ある小さな修正がひとつ組み込まれています。この修正を行うにあたって、シャープ株式会社さんより測定データやアドバイスのご提供をいただきました

図 7　アプリケーション改善事例（Unity）

参考文献

1）MMD 研究所 : 2013 年度スマートフォン購入者の満足度調査（Android 編），https://mmdlabo.jp/（2014 年 4 月 18 日現在）

丸木 勝也（非会員）maruki.katsuya@sharp.co.jp
シャープ（株）通信システム事業本部所属．2012 年より電池持ちを長くする取り組みを担当.

曽根田 裕郁（非会員）soneda.hirofumi@sharp.co.jp
シャープ（株）通信システム事業本部所属．2007 年よりシステム関連のソフト開発を中心に担当し, 現在はスマートフォン・タブレットの省電力化，発熱対策，パフォーマンスチューニングを担当.

日原 圭祐（非会員）hihara.keisuke@sharp.co.jp
シャープ（株）通信システム事業本部所属． 映像コーデック・グラフィックス関連業務に従事，担当モジュールの省電力化を担当.

陣之内 宏基（非会員）jinnouchi.kohki@sharp.co.jp
シャープ（株）通信システム事業本部所属．移動体ハードウェア開発を担当，これまで PHS 開発，PDC 開発，3G-FOMA 開発，LTE 対応スマートフォンを手掛ける．基地局通信制御，電流削減，回路開発，電源供給ネットワーク基板設計を得意分野とする.

江角 直起（非会員）esumi.naoki@sharp.co.jp
シャープ（株）通信システム事業本部所属．携帯電話のアプリケーション，ミドルウェア開発を担当.

採録決定：2014 年 9 月 8 日
編集担当：中野美由紀（芝浦工業大学）