

省電力なアプリ開発を容易にする センシングミドルウェア

Sensing Middleware for Easy Development of Energy-efficient Apps

● 上和田 徹 ● 中尾 学 ● 長谷川英司

あらまし

モバイル機器のセンシング機能で人の周りの様々な情報が収集できると、情報サービスがより便利に活用できるようになると期待されている。一方、モバイル機器がバッテリー駆動であるため、センシングや通信などの動作に要する消費電力は大きな課題であり、それを解決すべく様々な取り組みが行われている。一般にモバイル製品は、特定の用途や機器に特化するほど高い省電力効果が得られる傾向にある。しかし、特化した省電力化の手法は、ほかの用途や機器に転用できない汎用性の低いものとなりがちで、用途や機器ごとにアプリやファームウェアを作り直す必要が生じてしまう。このような背景から、富士通研究所は個別のセンサーごとの実装を不要にし、各機器に搭載したセンサーを自動的に使い分けて最も少ない消費電力になるようにセンシングを実行する「センシングミドルウェア」の技術を開発した。

本稿では、モバイル機器でセンシングするアプリの可搬性を高め、開発コストを大幅に削減できる本ミドルウェアについて述べる。

Abstract

Information services can be more useful if mobile devices are adapted to gather information in our surroundings using built-in sensing functions. Mobile devices, however, work with batteries, and various approaches are taken to tackle the major challenge the sensing and telecommunications mechanism poses in terms of their power consumption. Generally, mobile products tend to benefit from energy-saving effects more if their purposes or devices to use are more specific. However, saving energy by specifying the usage is likely to lower the products' versatility when it comes to reapplying them to other uses or devices, making it necessary to write apps and firmware every time the usage or applied devices change. Against this background, Fujitsu Laboratories has developed sensing middleware technology that makes it possible to choose and continually execute the least energy-consuming sensing method simply by using the app to specify what kind of information is required, while the middleware collects and manages the power composition of available sensors. This paper presents this middleware that significantly reduces development cost while increasing the app's mobility for mobile sensing.

まえがき

近年、スマートフォンの普及や様々なクラウドサービスの登場に伴い、様々な情報サービスを利用できる場面が急速に拡大している。特に、モバイル環境での活用が著しく増加しており、スマートフォンやタブレットといったスマート端末に搭載された多様なセンサーを用いて利用者の様々な状況を捉え、その状況に応じた便利な情報サービスが提供されている。例えば、GPSや無線LANを用いて利用者の位置を捉え、近隣の施設・店舗の紹介や移動経路の案内といった情報を提供するサービス、モーションセンサーで利用者の動きを捉えて健康促進や危険防止のための情報を提供するサービスなどが続々と登場している。

また、最近ではリストバンド型、腕時計型、眼鏡型などのウェアラブル機器が多数登場しており、モバイル環境における情報サービスのより豊かな利用体験が提案されている。ウェアラブル機器が備えるセンシングや通知の機能、簡便なユーザーインターフェースなどを通じて、より多様な場面でよりきめ細かい情報サービスを、自然な利用形態で提供できるようになりつつある。

このような情報サービスの提供におけるセンシング機能やウェアラブル機器の活用は、ビジネスの現場でより効力を発揮すると期待されている。例えば、物流、医療、建設、小売などの業務の現場において、作業で手が塞がって情報機器を扱づらい、あるいは扱えない場面であっても、センシング機能やウェアラブル機器を用いると、必要な情報の閲覧や入力がハンズフリーで作業に支障をきたすことなく実現できるようになる。

しかし、ウェアラブル機器もバッテリーで駆動するため、センシングし続けると作業の途中でバッテリーが切れてしまうといった問題が生じる。これに対して様々な省電力化の技術が存在するが、大きな省電力効果を得ようとする、ハードウェア構成やアプリに特化した方法になる。このため、ハードウェアやアプリごとにその都度省電力化を作り込む必要があった。

そこで富士通研究所は、ハードウェア構成に依存せずに継続的なセンシングを行うアプリを簡単に開発できる、センシングミドルウェアを軸とし

たソフトウェアアーキテクチャーを開発した。本稿では、そのセンシングミドルウェアに基づいたソフトウェアアーキテクチャーについて説明する。

モバイル端末におけるセンシングの課題

モバイル端末で人の状況を常にセンシングして捉え続けることによって、情報サービスを活用できる場面は大きく広がるのが期待できる。一方で、これらを支えるスマート端末やウェアラブル機器のほとんどがバッテリー駆動であり、連続稼働時間には制約がある。業務中にたびたび充電しなければならないようでは、作業の中断や新たな手間の増加といった不便を招く。したがって、センシングの省電力化は解決すべき重要な課題である。

● 従来技術：専用プロセッサへのオフロード

省電力化の手段として、スマートフォンで用いられる動作クロックがGHz級のメインCPUとは別に、低消費電力な専用のプロセッサを用いてセンサーからのデータを処理することで、継続的なセンシングの省電力化を行う方法がある（図-1）。

その代表例の一つに、富士通製スマートフォンに搭載の「ヒューマンセントリックエンジン」⁽¹⁾がある。この技術では、モーションセンサーが検知する連続した値からユーザーの動作などを捉えて、歩行状態や端末の保持状態に応じた操作支援といったサービスを低消費電力で可能にしている。その後、Apple社のiPhone⁽²⁾やMotorola社のスマートフォン⁽³⁾でも、専用のプロセッサを用いた類似の構成でセンシング処理の低消費電力化が行われている。このような専用プロセッサに多様なセンサーを接続する取組みは増えており、「センサーハブ」として広く用いられるようになってきている^{(4),(5)}。

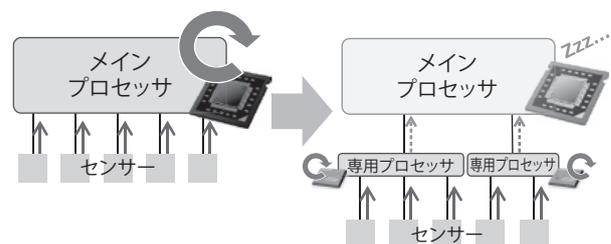


図-1 専用プロセッサを用いた省電力化

● 課題：複数センサーの組合せのファームウェア化

前節で述べた専用プロセッサを活用することで、様々なセンサーを使ったサービスを低消費電力で実現できるようになる。このとき、どのセンサーをいつ使うかをきめ細かく制御すると、更に消費電力を抑えることができる。情報を常時取得しなければならないセンサーだけ電源をONとし、ほかは基本的にOFF、必要になったときだけON、使い終わったら直ちにOFFにするという使い方となる。例えば、スマートフォンの位置情報を常時取得するために、GPS受信や無線LAN基地局の探索を常に続けていると大きく電力を消費してしまう。そこで、より消費電力の少ない加速度センサーを用い、その測定値に変化がない（スマートフォンの位置が変わっていない）と分かる間は、GPSや無線LANの受信を止めることで消費電力を削減できる。こういった省電力化を専用プロセッサのファームウェアとして実装し、メインCPUではなく専用プロセッサだけで制御できれば、様々な場面で人の情報をセンシングし続けられる。

ところが、複数のセンサーをどのように組み合わせさせて判定するかのロジックは、アプリによって異なる。例えば、上述のようにスマートフォンの位置が変わっていないときに無線LANを停止させる方法は、移動している人に対して特定の場所に近づいたら通知するジオフェンシングのようなアプリでは有効である。一方、無線LANのアクセスポイント（AP）の情報を収集する無線ログガーのようなアプリに適用してしまうと、スマートフォンが静止していても変化するモバイルルータなどのAPの情報を取りこぼすようになってしまう。このようなアプリによって異なる処理は、専用プロセッサの固定的なファームウェアとして実装できない。ファームウェア書換えは手間を要しリスクも伴う作業であるため、頻繁に行うものではない。このため、このようなアプリに応じて異なる処理は、メインCPUで行うことになる。このときメインCPUは、他センサーをON/OFFする判断材料となるセンサーの値をモニターし続けなければならない。継続的なモニタリングの省電力化に専用プロセッサを活用できても、他センサーのON/OFFを判断する際には、メインCPUが動作する必

要が生じてしまう。

このように、複数のセンサーの組合せ判定とそれらの駆動制御まで専用プロセッサに任せられると、メインCPUは組合せ判定の条件が成立したときだけ動作すれば良いことになる。一方、前述のとおり、その判定ロジックはアプリ依存となるため、固定的なファームウェアとして実装できない。したがって、この処理に要するメインCPUの消費電力を削減できなかった。

開発技術

前章で述べた課題に対し、富士通研究所ではセンシングミドルウェアを開発した。アプリから指定された情報の継続的なセンシング処理や、個々のセンシング結果に応じたセンサーの駆動制御を低消費電力のプロセッサで実行できるものである。このセンシングミドルウェアは、スマートフォン上でバックグラウンド動作し、センシングを要求するアプリとセンサーをつなぐ役割を担う。センサーを備えたウェアラブル機器側は、ミドルウェアからの指示に従った処理を行うセンシングノードをファームウェアの一部として用意する。これにより、ミドルウェアを介したアプリからの様々な要求に対して、同じファームウェアで対応できるようになる。このセンシングノードが実行できる処理は、例えば次のようなものである。

- センサーの駆動制御（ON/OFF）
- センサーからのデータの取得
- センサーデータに対する演算（マッチングなど）
- 演算結果の記録・保持
- 複数の演算結果に対する組合せ判定

このような処理を実行可能な一つ以上のセンシングノードとそれらを統括するセンシングミドルウェアが、アプリの要求に応じて長時間のセンシング処理を低消費電力で行う端末周辺の分散処理型アーキテクチャーとなっている。その構成を図-2に示す。

このアーキテクチャーでは、スマートフォンに接続されているウェアラブル機器について、それぞれがどのようなセンシング機能や組合せ判定機能を持っているか、それらをどのくらいの消費電力で実行できるかなどのセンシング処理に関わる情報を事前に集めておく。この準備ができたミド

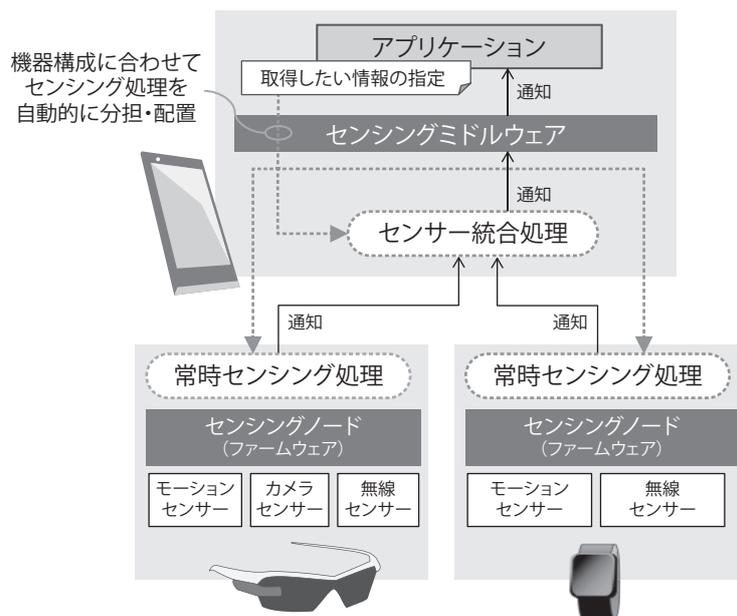


図-2 開発技術の構成

ルウェアに対して、アプリはセンシングの結果として必要な情報を要求する。例えば「歩行中にSSID (Service Set Identifier) の値がXYZである無線LANのAPを見つけたら通知」という内容で要求を伝える。このとき、どのようなセンサーを使ってどのように判定するかをアプリから指定する必要はなく、あくまで欲しい情報が何であるかを伝えるだけで構わない。

この要求を受けたミドルウェアは、受けた内容から「歩行状態」と「無線LAN APのSSID」のデータや値を得る個々のセンシング処理と、センシング結果が「歩行中」や「XYZ」に一致するかどうかの判定処理、および両判定処理結果が正確であるかどうかの組合せ判定処理を行うものと解釈する。ここで解釈した内容を条件判定式のリストとして管理する⁶⁾このリストと、事前に集めておいたウェアラブル機器の情報を用いて、どの条件判定処理をどの機器で実行すれば最も消費電力を抑えられるかを判断し、各処理の機器への割当てを決定する。その決定に従い、個々の条件判定式に対応する機器に送り込むことによって、割り当てた処理の実行指示を出す。

ひとたびウェアラブル機器に実行指示を出したら、ミドルウェアはスマートフォン本体をスリープ状態に移行させて、電力を消費させないように

する。その間、ウェアラブル機器は、受けた指示に沿ってセンシング処理を行い続ける。ウェアラブル機器でセンシングしたデータを条件判定式で評価し、その結果によりスマートフォン本体へ信号を送ってスリープ状態から復帰させ、センシング要求が満たされた旨を通知する。センサーを通じて歩行状態の変化を検知したり、いろいろな無線LAN APを見つけたら通知する。このアーキテクチャーでは、そのたびにスマートフォンがスリープ状態から復帰することはなく、アプリが要求した「歩行中にSSIDの値がXYZである無線LAN APを見つけた」ときだけ復帰してメインCPUを動作させる。そのため、このセンシングに要するスマートフォン本体の稼働時間を大幅に削減できる。

しかも、この省電力効果を得るに当たって、アプリはどのセンサーをどのように使うかを指定する必要はない。そのため、アプリ開発者は端末のハードウェア構成を意識することはなく、本ミドルウェアが動作している端末であれば、同じアプリが同じロジックで動作する。ハードウェア構成に関わらずアプリを共通化できるため、機器に応じて開発の工数が膨れ上がるようなことはない。逆に、端末やウェアラブル機器側の観点からすると、専用のプロセッサに継続的なセンシングを任

せて省電力化するに当たって、アプリのセンシング要求内容が異なっても、ウェアラブル機器のファームウェアを更新することなく、最も消費電力を抑えた方法で実行できるようになる。つまり、センシング要求内容を境界面として、アプリとハードウェア構成の自由自在な組合せと、各組合せに応じたセンシングの省電力化とを、少ない開発工数で両立できるようになっている。

動作例

このミドルウェアを活用する例として、同じ構成のハードウェアをセンシング処理の異なる業務で利用する例を示す(図-3)。ここでは、(A) 機器設備の点検と、(B) 倉庫での商品ピックアップの2種類の業務を想定する。どちらの場合でも、スマートフォンとウェアラブル機器を使用し、このスマートフォンにはセンシングミドルウェアが搭載されている。ウェアラブル機器には、センシングノードとして動作するファームウェアが搭載されており、両方の業務で同じものを使用する。スマートフォン上で動作させるアプリは、それぞれの業務ごとに用意することになる。

機器点検では、作業者が立ち止まった場所の周囲に点検対象の機器があれば、その機器の点検内容を作業者に提示する。機器の有無は、機器に貼り付けられた無線ビーコンを用い、それをウェアラブル機器の無線センサーで検知する。立ち止まったかどうかの判定には、ウェアラブル機器に内蔵された歩行センサーを用いる。機器点検用のアプ

リは「機器の前で立ち止まったら通知」とだけミドルウェアに要求を伝えておくと、それに合わせた指示がセンシングノードであるウェアラブル機器に伝えられ、立ち止まったときだけ無線センサーを動作させるセンシング処理が行われる。

ピックアップ業務では、倉庫内を移動中の作業者が無線ビーコンの貼り付けられた商品棚に近づくと作業者に通知して呼び止め、該当する商品のピックアップを促す。この場合のアプリは「特定IDのビーコンに歩いて近づいたら通知」と要求を伝え、ミドルウェア経由でそれを受けたウェアラブル機器は、作業者が歩行している間だけ無線センサーを動作させる。

以上のように、アプリからのセンシング要求が変わった場合でも、本ミドルウェアとウェアラブル機器の連携によって多様なセンシングを省電力化された方法で実行でき、アプリに合わせてウェアラブル機器のファームウェアを作り直す必要はない。

従来の方法では、アプリに応じて省電力効果を引き出すために、アプリ内での作り込みに加えてドライバやファームウェアまで開発する必要があった。これに対し、本ミドルウェアを用いる方式では、アプリが必要とするセンシング情報を指定するだけで済むため、大幅に開発工数を抑えられる。実際に、前述の点検業務用のアプリとピックアップ業務のアプリの開発に要する工数は、本技術によって10分の1程度に抑えることができた(図-4)。

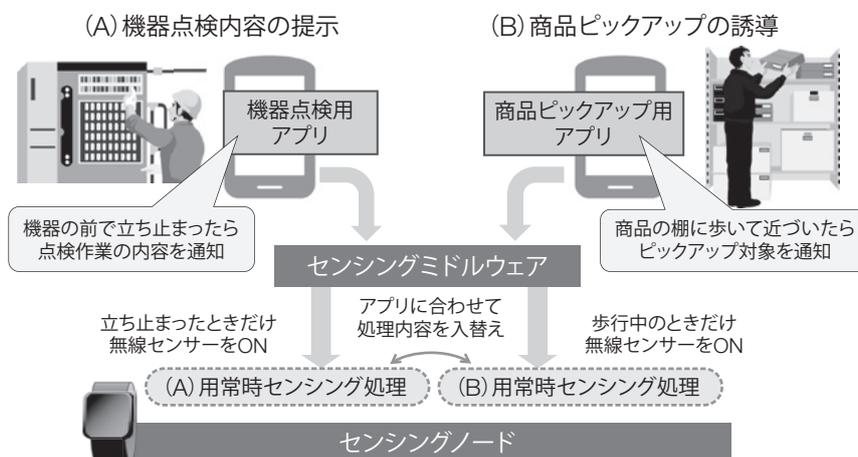


図-3 センシングミドルウェアの適用例

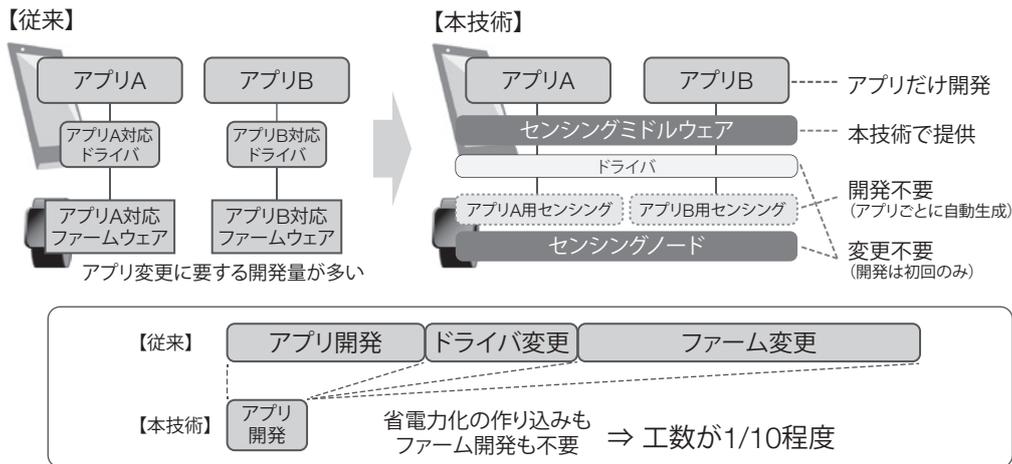


図-4 開発工数の削減

む す び

モバイル機器のセンシング機能によって、情報サービスをより便利に活用できるようになると期待される。一方、そのセンシングに要する消費電力は大きな課題でありながら、センシングの用途や機器によらず省電力化技術を共通に適用できていなかった。そのため、アプリやサービスに応じて個別に省電力化の作り込みが必要となり、開発工数を増加させる要因の一つとなっていた。

この課題を解決するために、富士通研究所はセンシングミドルウェア技術を開発した。この技術では、センサーを備えたウェアラブル機器の情報をミドルウェアが収集・管理し、様々なアプリの要求に合わせてそれらを適切に使い分ける。これにより、アプリが変わってもウェアラブル機器のファームウェアを作り直すことなく利用できたり、同じアプリを様々なハードウェア構成で利用したりできるようになる。そのため、用意したウェアラブル機器で様々なアプリが使える、一度作ったアプリもまた、様々な機器で利用できる。この方式により、あるセンシングのケースでは、開発工数を10分の1程度に抑えられることも確認できた。これにより、モバイルでのセンシング機能を活用したアプリの生産性を高め、情報サービスの可能性を更に押し広げることができる。

参考文献

- (1) 中条 薫ほか：ヒューマンセントリックエンジンとサービス展開. FUJITSU, Vol.63, No.5, p.505-511 (2012).
<http://img.jp.fujitsu.com/downloads/jp/jmag/vol63-5/paper02.pdf>
- (2) Apple Inc. : Apple - iPhone 6 - テクノロジー.
<https://www.apple.com/jp/iphone-6/technology/>
- (3) Motorola Mobility LLC. : X8 mobile computing system for Motorola.
<https://www.motorola.com/us/X8-Mobile-Computing-System/x8-mobile-computing-system.html>
- (4) Atmel Corporation : Sensor Hub.
http://www.atmel.com/products/touchsolutions/sensor_hub/
- (5) Qualcomm Technologies, Inc. : Sensor processing done right : Smart integration of a specialized sensor engine.
<https://www.qualcomm.com/news/onq/2014/06/05/sensor-processing-done-right-smart-integration-specialized-sensor-engine>
- (6) 佐藤卓也ほか：複合的なセンシング処理を端末ハードウェアの構成に合わせて最小の電力で行うセンシングミドルウェア. マルチメディア, 分散, 協調とモバイル (DICOMO2014) シンポジウム, p.1575-1580 (2014).

著者紹介



上和田 徹 (かみわだ とおる)

ユビキタスシステム研究所
ユビキタスデバイスプロジェクト 所属
現在、モバイル端末の省電力化、セン
シング、機器連携技術の研究開発に
従事。



長谷川英司 (はせがわ えいじ)

ユビキタスシステム研究所
ユビキタスデバイスプロジェクト 所属
現在、モバイル端末のセンシング技術
の研究開発に従事。



中尾 学 (なかお まなぶ)

ユビキタスシステム研究所
ユビキタスデバイスプロジェクト 所属
現在、センシング技術の研究開発に
従事。