

## 論文内容の要旨

論文題目 ユーザエクスペリエンスを考慮した無線センサネットワークに関する研究

氏名 猿渡 俊介

半導体技術の進歩はパーソナルコンピュータの高性能化に留まらず、さまざまな分野に大きな影響を与えている。半導体技術によって培われたリソグラフィ技術はMEMSを生み、超小型・高感度・低消費電力の特徴を持ったセンサやアクチュエータの実現を可能とした。また、SoC (System-on-Chip) はマイクロコントローラと無線通信機能を1チップ化し、RFID技術や無線センサネットワークなどの誕生に寄与した。さらに半導体技術とMEMSを組み合わせることで、マイクロコントローラ、無線通信機能、センサ・アクチュエータを1つのチップ上に統合することも可能になりつつある。このような超小型のセンサやアクチュエータが、将来的にはさまざまなオブジェクトに組み込まれることが予想される。

センサやアクチュエータが組み込まれたオブジェクトはさまざまな情報の入力や出力が可能となる。本稿ではセンサを「環境からの情報を取得する機能」、アクチュエータを「環境に対してアクションを起こす機能」というように広義に捉える。例えば、センサは椅子に組み込まれた3軸加速度センサといった単なるセンサだけでなく、壁のスイッチなどの入力装置もセンサとして扱う。アクチュエータは、カーテンの開閉を制御するモータだけでなく、部屋の間接照明などの情報を出力可能な装置もアクチュエータとして扱う。このようなわれわれの身の回りに存在する膨大な数のセンサとアクチュエータを連携させることで今までにないような新しいサービスが誕生する可能性がある。

筆者は、このような無線センサネットワークの可能性に着目し、通信プロトコルからオペレーティングシステム、アプリケーションまでの総合的な観点から研究を行った。特に、新しい技術は使いやすさ(ユーザエクスペリエンス)が重要であると考え、プログラムの開発のしやすさを考慮した無線センサノード用のオペレーティングシステムと、ユーザが身の回りに組み込まれたセンサやアクチュエータを使用して簡単にサービスを構築可能とするフレームワークの設計と実装を行った。

現在、無線センサネットワークのオペレーティングシステムとしてTinyOSが標準として扱われている。筆者らは2001年から無線センサネットワークの研究を進めており、そのとき既に標準になりつつあったTinyOSの使用を検討した。しかしながら、TinyOSがevent modelを用いているためにプログラムが書き辛かったこと、ハードリアルタイム性をサポートしていないことの2点の理由から筆者の研究の要求に合致しなかったため、独自にオペレーティングシステムを開発することにした。

このような背景から開発されたオペレーティングシステムが本論文に示すPAVENET OSである。PAVENET OSは、thread modelを用いることでユーザに対してプログラムの書きやすさとハードリアルタイム処理のサポートを実現することを目指している。PAVENET OSではpre-emptiveとco-operativeのハイブリッドのスケジューラを具備している。pre-emptiveのスケジューラではCPUの動的な優先度割り込みの機能とハードウェアによるコンテキストスイッチの機能を利用す

ることで少ないオーバーヘッドでハードリアルタイム処理を実現する。co-operative のスケジューラでは、os\_yield や sleep などユーザが明示的に CPU を開放する関数を実行することで排他制御やコンテキストスイッチの負荷を軽減する。さらに、無線センサノードにおけるタスク間のデータの交換が無線通信における各層のパケットの交換時に多発することに着目し、オペレーティングシステムとして無線通信プロトコルのレイヤリングの機能を提供して排他制御を API に隠蔽することで各層のモジュール性を実現する。

PAVENET OS と TinyOS を比較すると、PAVENET OS のカーネルは TinyOS と同程度の計算資源で実現できる。例えば、TinyOS のサンプルプログラムである Blink と同じ機能を PAVENET OS で実現した場合には TinyOS ではデータメモリが 74byte、プログラムメモリが 1700byte 必要であるのに対し、PAVENET OS ではデータメモリの使用量は 60byte、プログラムメモリが 1600byte で実現できる。また、パフォーマンスも計算資源と同様に TinyOS と同程度の性能を実現できる。一方で、PAVENET OS は TinyOS と異なり、ハードリアルタイム処理を実現する機構を提供する。そのため、TinyOS では実現できないような、たとえば無線通信を行いながら正確な 100Hz を刻むといったハードリアルタイム処理が実現できる。

このような特徴を持つ PAVENET OS を用いて筆者らは、センサとアクチュエータを連携させるためのデバイス連携フレームワーク ANTH の設計と実装を行った。ANTH は、センサやアクチュエータが身の回りのありとあらゆるオブジェクトに組み込まれた空間において、一般のユーザが簡単な操作でこれらのオブジェクト同士を連携させ、サービスを構築するためのフレームワークである。センサとアクチュエータによってもたらされるサービスはセンサがイベントによる環境の変化を検出するという機能を持っているがゆえにイベントドリブンの特性を持つ。また、われわれの身の回りに存在するオブジェクトは膨大な種類が存在するため、オブジェクト同士の接続の柔軟さや新しいオブジェクトの開発の容易さが重要になってくる。

筆者は ANTH をイベントドリブンプログラミングの仕組みを単純化した Bind Control Model を基に構築した。通常のイベントドリブンプログラミングではイベントからコールバック関数に対してデータを渡すため、イベントとコールバック関数との関係が静的に決められる。それに対して Bind Control Model ではすべてのコールバック関数をデータを受け取らない形に統一する。このような仕組みにより、呼び出し側のプログラムから呼び出される側のプログラムへの接続が柔軟になる。また、呼び出し側と呼び出され側のプログラムをそれぞれ独立に設計することができ、開発が容易になる。一方で Bind Control Model では単純さゆえに実現できるサービスが固定的になり、ユーザによる操作の負担が発生する。ANTH ではセンサノード上で動作する VM を用いてセンサやアクチュエータの機能を動的に追加する機構を導入することで実現できるサービスの幅を広げる。また、操作端末を利用したいオブジェクトに近付けて操作する直感的なヒューマンインタフェースを導入することでユーザによる操作の負担を軽減する。さらに、無線通信におけるアプリケーション層から MAC 層までを縦断的に考慮することで低消費電力な無線通信を実現する。

これらの機構を筆者が開発した PAVENET OS 上に実装した。ANTH の最小セットはプログラムメモリ 3 kbyte、データメモリ 237 byte で実現することができる。つまり ANTH の最小セットは Microchip 社の PIC16 といったメモリ 256 byte、プログラムメモリ 1 kbyte の非常に計算資源の少ない CPU 上でも実装することができる。また、VM の命令セットを ANTH で扱うサービスに特化して構築することでネイティブで実行するとき比べて 1.24 倍の実行速度の低下に留めた。さらに、ANTH の無線通信プロトコルをシングルホップで構築し、かつ電源状況に応じて複数の MAC プロトコルを切り替えることでバッテリーで動作するオブジェクトでは劣化を無視した場合に単三電池 2 本で約 3 年動作の低消費電力化を実現した。