

## センサノード用組み込み OS のハードウェア化

小林昌太<sup>†</sup> 三井浩康<sup>†</sup>

東京電機大学理工学部

## 1. はじめに

近年, さまざまな分野へのセンサネットワークの応用が研究されている. センサと無線通信端末からなる無線センサノードを使用することで周囲の情報を収集し, インターネット環境に取り込むことが可能になる.

無線センサノードはバッテリーで駆動することが多く, 森林, 農場, 工場などに多数設置した場合, 点検や回収が困難になるため電池交換の作業も困難になる. できるだけ長期間バッテリー駆動させるためにセンサノードは低消費電力であることが望ましい.

低消費電力で動作させるために, センサノードの CPU クロック性能は低く抑えなければならない. そのためソフトウェア (S/W) を開発する時は限られた資源をうまく活用し, 高い性能を実現する必要がある. しかし, 最適化しすぎると S/W の開発効率や再利用性を下げることになる.

これらの問題を解決するためにセンサノードに組み込み OS を使用することで S/W の開発効率を上げ, 限られた資源を有効に使い, 高い性能を実現することができる.

本研究ではセンサノード用 OS として広く用いられている TinyOS の機能の一部をハードウェア (H/W) 化してオーバーヘッドを減らし更なる消費電力の削減をすることを目的とする.

## 2. 研究の背景

組み込みシステム用 OS の H/W 化の研究として,  $\mu$ ITORN の LSI 化の研究がある. この研究は  $\mu$ ITORN の機能の一部を LSI 化することによってシステムのリアルタイム応答性を向上させるというものである<sup>[2]</sup>.  $\mu$ ITORN はリアルタイム OS として組み込みシステムで広く使用されているが, センサノードに適用するには, 規模が大きすぎるため, 本研究では TinyOS を選択した.

TinyOS はワイヤレス・センサ・ネットワークで使用されている OS である. オープンソース

プロジェクトとして開発されており, 簡単に機能の拡張を行なうことができる. また, 軽量であることを目標に開発されたため, OS に要求される資源やオーバーヘッドの最小化が実現されている. TinyOS の省資源性や低オーバーヘッドを超える OS は存在していない<sup>[1][3]</sup>.

## 3. 研究内容

## 3.1 研究概要

本研究の目標は TinyOS のオーバーヘッドを減らすことで, センサノードの消費電力を削減することである.

TinyOS はオーバーヘッドを最小限に抑えるよう設計されているため S/W 的にこれ以上オーバーヘッドの削減をすることが困難である. そこで TinyOS の機能の一部を H/W で高速に処理を行いオーバーヘッドの削減を行なう.

また, H/W 化の手段として再構成可能な H/W である FPGA (Field Programmable Gate Array) を使用することで, OS の仕様の変更やバグにも柔軟に対応できるようにする.

図 1 に提案システムの概要図を示す.

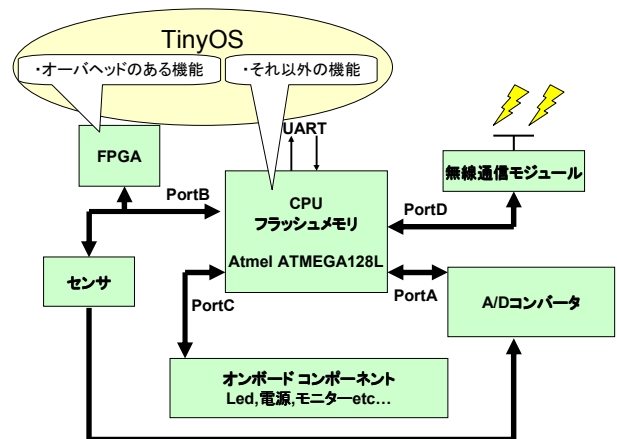


図 1. システム概要図

本研究では H/W 化した機能を実装するために FPGA を採用した. TinyOS に仕様変更やバグが発生した場合に LSI などの H/W では回路の修正が困難であり, 開発期間や開発コストも増加する. しかし, FPGA を使用することで機能の追加

Implementing embedded OS for sensor node using Hardware

<sup>†</sup>Shota Kobayashi, Hiroyasu Mitsui

School of Science and Technology, Tokyo Denki University

や改良が容易になり, これらの問題に柔軟に対応することができる。

また図1に示すように FPGA と既存の CPU を独立して動作させることが可能となる。

### 3.2 TinyOS のハードウェア化

図2に TinyOS スケジューラ機能の H/W 化の図を示す。

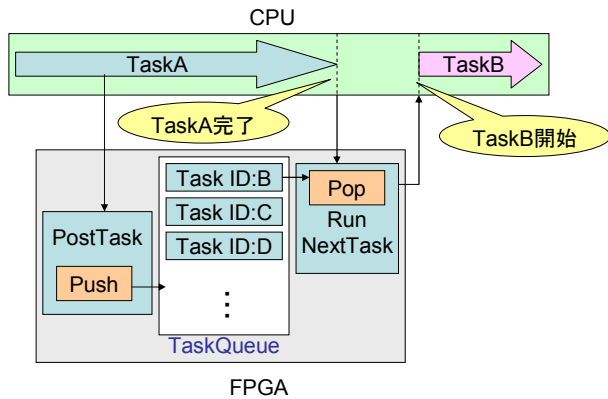


図2. TinyOS スケジューラ機能の H/W 化

本提案ではスケジューラ機能を H/W 化の対象とする。TinyOS のスケジューリング機能は FIFO (First In First Out) で構成され, 低オーバーヘッドのスケジューリングを可能にしている。しかし, CPU で処理されるデータ収集や通信のタスクは TinyOS のスケジューリングのオーバーヘッドと同程度に処理時間が短いため, TinyOS のオーバーヘッドを無視することができない。また, FIFO で CPU 資源を割り当てているため, タスクの実行中にタスクをタスクキューに挿入するとタスクの実行に遅延が起ってしまう。

従来のスケジューラではソフトウェアによる順次処理を行なっているが, ハードウェアに置き換えることによって最小限のクロック数で処理を行なえることになり, 処理速度が向上する。また, CPU と独立に処理させているのでタスク実行中にタスクをタスクキューに挿入した場合でも CPU 資源を使わずにすむため発生する遅延が減少する。その結果オーバーヘッドの削減が可能となり, 従来手法よりも消費電力を削減することができる。

## 4. 実装・評価

### 4.1 実装

今回は提案手法を実装するためにプロトタイプの構築を行なった。

図3にプロトタイプの構成図を示す。

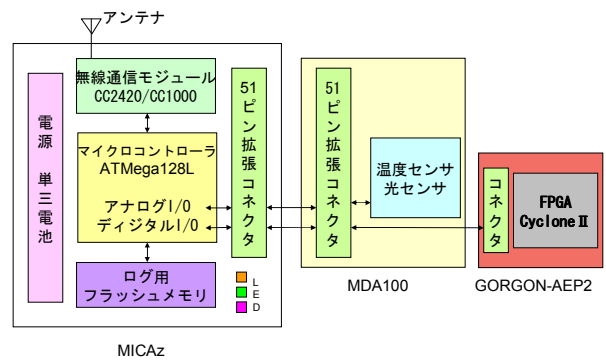


図3. プロトタイプの構成図

センサノードとしては Crossbow 社の MICAz を使用した。MICAz の CPU は Atmel 社の ATMega128L である。FPGA には Altera 社の Cyclone II を搭載している GORGON-AEP2 評価ボードを使用した。MICAz と GORGON-AEP2 を接続するために, MICAz の拡張基板である MDA100 を用いた。MDA100 は温度センサと光センサを搭載している。MICAz と GORGON-AEP2 は 51 ピンコネクタを使用して接続する。

TinyOS のスケジューラ機能の H/W 化は, H/W 記述言語である verilog-HDL を使用して開発し, FPGA に実装した。

TinyOS と FPGA の通信機能 S/W は C の拡張言語である nesC を用いて実装した。

### 4.2 評価

評価方法として, 従来の TinyOS を MICAz で動作させた場合と, 提案手法である MICAz + FPGA で動作させた場合の性能の比較を行なう。

評価項目としてオーバーヘッドの時間, 動作時と待機時の消費電力を取り上げ, 提案手法の評価を行った。

## 5. まとめと今後の課題

センサノード用 OS のスケジューラ部をハードウェア化することで, 性能向上と消費電力の削減を行なう方法を提案した。

センサノードの CPU がスリープ状態の時に FPGA をどのように制御するかが今後の課題である。

### 参考文献

- [1] 坂田史郎:「ユビキタス技術 センサネットワーク」, オーム社, 2006
- [2] 中野巧:「ITRON を LSI 化する〜リアルタイム OS をより高速化するために〜」, Design Wave Magazine, No27, p74-89, 2000
- [3] 猿渡俊介・鈴木誠・大原壮太郎・南正輝・森川博之:「無線センサネットワークにおけるオペレーティングシステムの研究動向」東京大学 先端化学技術研究センター, 技術研究報告書 No.2008003, 2009