

# Sun SPOT におけるワイヤレスセンサネットワークを用いた温度制御方式

寺島 悠貴<sup>†</sup> 山原 亨<sup>†</sup> 中村 謙太<sup>†</sup> 大谷 真<sup>†</sup>

湘南工科大学情報工学科<sup>†</sup>

## 1. はじめに

Sun SPOT (Sun Small Programmable Object Technology) とは Sun Microsystems が開発及び販売している無線センサネットワークデバイスである。今までのセンサデバイスと異なり、Java 言語でのプログラミングが可能である。本研究では Sun SPOT を用いての温度制御方式を検討し妥当性評価のためのシミュレーションプログラムを開発した。またその結果から Java で小型組み込みシステムを開発することの評価を行った。

## 2. センサネットワーク及び Sun SPOT

### 2.1 センサネットワーク

近年センサネットワークはさまざまな分野に使われ始めてきた。例としては、ビル管理オートメーション、環境や自然災害などのモニタリング、児童の登下校時における安否確認、あるいは健康管理、品質管理にも使われている [1]。

センサネットワークはかなり身近で使われてきているが、将来的にはもっと広域的に利用されることが望まれている [2]。そのためには、システムをもっと容易に開発できることが必要である。

### 2.2 Sun SPOT

Sun SPOT は Sun が開発した最新の無線センサネットワークデバイスで、加速度、照度、温度の 3 種のセンサと、3 色 LED、制御用スイッチ、及び、IEEE802.15.4 準拠の無線を持つ。

最大の特徴はこれらを Java でプログラミングし、動作させることが出来るという点である。このため専用の Java VM を搭載している。

## 3. 温度制御方式について

### 3.1 検討内容

小型組み込み系システムを Java で作ることに對しての評価を行うため Sun SPOT の温度センサと無線通信を利用したアドホックな温度制御を検討した。

### 3.2 温度制御モデル図

図 1. に前提とした温度制御方式のモデル図を示す。図はサーバあるいは重要機材が設置されている空間(例:サーバールーム、倉庫)としたモデル図で、Experiment on Wireless Sensor Network Application using Sun SPOT for Temperature Control Yuuki Terashima<sup>†</sup>, Toru Yamahara<sup>†</sup>, Kenta Nakamura<sup>†</sup>, Makoto Oya<sup>†</sup> Department of Information Science, Shonan Institute of Technology<sup>†</sup>

サーバあるいは重要機材を管理する管理者がいるものとする。管理者は管理用マシンと管理用 Sun SPOT を使用する。サーバなどの温度制御が必要な機器は対象機材として、周囲に温度測定用の Sun SPOT を配置。対象機材には実際に温度を下げる温度制御機器、温度制御機器には制御機器を動作させる制御用 Sun SPOT が接続されている。

管理用 Sun SPOT は測定用 Sun SPOT と制御用 Sun SPOT に命令を送り、データを集めるものである。測定用 Sun SPOT は温度を測定し管理用 Sun SPOT にデータを送るものである。制御用 Sun SPOT は温度制御機器を動作させて対象機材の温度を下げるものである。Sun SPOT 間の通信構成は Peer to Peer である。

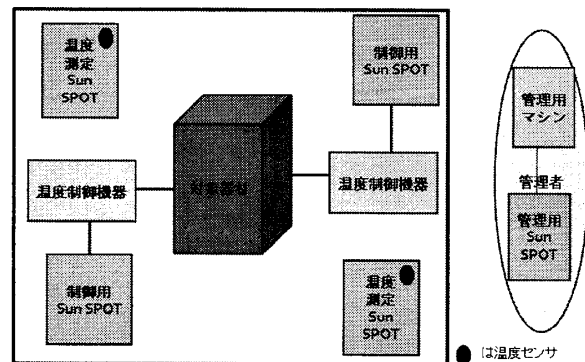


図 1. 温度制御方式のモデル

### 3.3 温度制御方式

Sun SPOT を用い以下のとおり制御する方式とした。動作としては、

1. 管理者が管理用 Sun SPOT から温度測定を要求する。
2. 測定用 Sun SPOT は要求に従って温度を測定及び管理用 Sun SPOT に送信。
3. 温度制御が必要な場合、制御用 Sun SPOT に指令を送信。制御用 Sun SPOT は温度制御機器を動作させ対象機材の温度を下げる。
4. 測定用 Sun SPOT は一定周期で温度を測定、送信し続ける。
5. 温度が十分に下がり、制御の必要性がなくなったら、測定用 Sun SPOT、制御用 Sun SPOT に対して停止命令を送信して温度測定と温度制御機器を停止させる。

### 3.4 プログラム構成

Peer to Peer の通信構成を実現するためにマルチスレッドを使用した。温度及び各命令の送受信の組ごとにスレッドとして動作するようにプログラムを行った。

以下に温度測定と送信を行うソースの一部を示す。

```
range = tempSensor.getRange();
temp = tempSensor.getValue();
tempT = tempSensor.getCelsius();
dg.reset();
dg.writeUTF("temp");
dgT.reset();
dgT.writeDouble(tempT);
dgConnection.send(dg);
dgConnection.send(dgT);
```

図2.プログラムの部分構成

tempSensor は温度センサのオブジェクトであり、getCelsius()は摂氏温度での温度測定を行うメソッドであり、dgConnection.send()は()内のデータを送信するメソッドである。

このように Java 言語により、C 言語に比してはるかに少ない量のコードで開発が出来る。

#### 4. シミュレーションについて

##### 4.1 シミュレーションモデル

今回の開発環境では対象機材、温度制御機器が無い場合、実際に動作する温度制御方式のプログラムは出来ない。そこでシミュレーションを行い温度制御方式の妥当性を検討した。

図 1. のモデル図で対象機材、温度制御機器に相当する部分を管理用 PC とは別の 1 台の対象 PC としたモデルを元にシミュレーションを行う。

##### 4.2 シミュレーション方式

シミュレーションでは、対象 PC の温度が温度制御を必要とするほど上がらないといったことを避けるため実装されている温度センサを使わずに、35.0 以上 60.0 未満の範囲を指定した乱数を生成して、温度とみなした。また、下がった温度を測定するために、乱数と、パラメータとを使い測定用 Sun SPOT で減算することで、温度制御を行った。そして、温度制御を行うためには測定側が温度を下げる必要があると認識しなければならないので、制御用から測定用に対して仮の命令を送信することで、温度を下げる準備を行う。

乱数が 50.0 を超えた場合、20 秒ごとに 1.0 ずつ減算が行われる。減算により値が 50.0 未満になるか、乱数生成時点で 50.0 未満であれば次の乱数を生成する。

##### 4.3 実行結果

実行時間 640 秒で 36 回分の温度測定を行ったシミュレーション結果(図 3.)を見ると 50.0 以上の値は徐々に小さくなっていることから温度制御に関して問題なく行えている。

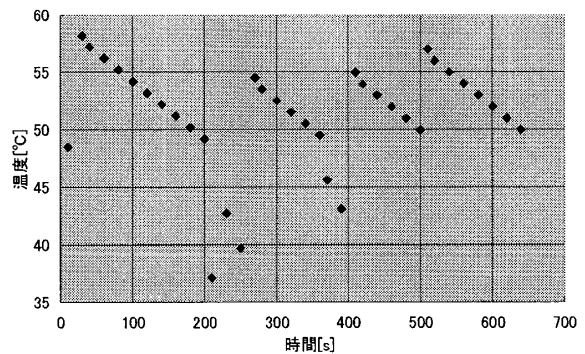


図 3. シミュレーション結果

図 3. で 50.0 以下の値を生成した次の生成で、突然 60.0 近くの高い値を生成したり、温度制御が終了した次の乱数生成時に 40.0 を下回るような値が生成されたりしているのは、最小値は温度制御が必要なく、最大値では温度制御が必要となる間で乱数を生成していたことによるものである。

#### 5. 開発効率について

##### 5.1 温度制御方式の検討

上記のシミュレーション結果より、温度のように単位時間当たりの変化量が少ないものに関してのセンサを使用したモニタリングは Java での開発も有効である事が分かった。

##### 5.2 Java による小型組込みシステムの開発

温度センサのアクセス時間は、500 回の計測で平均 0.556 ミリ秒となった。

ガベージコレクション発生時には最大で 15 ミリ秒の時間がかかり、動作全体では 1.57 秒遅延した。

C 言語による温度センサでの温度測定のソース [3] と Sun SPOT の温度センサでの温度測定のソースを比較してみると、C 言語のライン数は 6、Java のライン数は 2。比較すると C 言語の 33%分の長さになった。

以上から本アプリケーションないし類似のアプリケーション開発には Java の開発も有効である。

#### 6. まとめ

本論文では温度制御方式を検討したが、機材不足のため、シミュレーションによる評価となった。今後は実際の機材を用いた実験を行い評価する予定である。

応答性能はガベージコレクションの遅延を含めても実用には十分であった。今後現実と同等の環境での検証を十分に行うことによってセンサネットワークの広域的な使用に有効なものになるだろう。

参考文献  
[1] [http://www.soumu.go.jp/menu\\_02/ictseisaku/ictRD/051020\\_2\\_3\\_3.html](http://www.soumu.go.jp/menu_02/ictseisaku/ictRD/051020_2_3_3.html)  
[2] <http://labs.ubiteq.co.jp/ja/ips/sn/>  
[3] C 言語の温度センサのソース <http://www.picfun.com/p63005.html>  
[4] Sun SPOT-OwnersManual