

フィジカルコンピューティングデバイス Arduino を用いた センサネットワーク管理機構の開発

竹永 康夫 横田 裕介 大久保 英嗣

立命館大学情報理工学部

1 はじめに

ワイヤレスセンサネットワークシステムは、デバイスの小型化や高性能化、省電力化が進められ、様々な実用的なアプリケーションに適用可能になってきている。例えば、環境モニタリングでは広範囲に数十、数百のセンサノードが配置されることが想定されるが、このような大規模なセンサネットワークを運用するにあたっては、ノード設置現場における機器の故障や、通信障害への対応作業が非常に重要である。しかし、従来のセンサネットワークシステムでは、ホスト PC 側における管理機能に重点が置かれており、現場での管理作業を支援する機能は不十分であった。本稿では、モバイルノードと呼ぶシステム管理作業のためのノードを導入することにより、このような問題を解決する手法について述べる。

2 センサネットワークの管理と課題

センサネットワークを管理する方法としては、ホスト PC 側で各センサノードのバッテリー残量やネットワークトポロジ、ノードの状態を把握し、管理を行う方法が一般的である。代表的なネットワークの管理ツールとして MOTE VIEW [1] が挙げられる。MOTE VIEW とは、米 Crossbow 社で開発されたセンサノードデバイスのためのアプリケーションであり、ネットワークトポロジの視覚化や、センサデータのグラフ化、データベースとの接続によるセンサデータのデータベースへの保存などを実現する。

MOTE VIEW はセンサネットワークシステム管理に必要な主要な機能を提供する、代表的なツールであると言えるが、実際の運用においてはまだいくつかの課題が残されている。

- ノードの位置の管理：あるノード ID を持つノードが実際にどの位置に設置されているかを管理することが難しい。手作業によって設置位置を記録したり、ノード本体に ID を書いたラベルを貼る必要がある。
- 通信障害への対処：長期運用時に、周囲の環境の変化によって通信障害が発生し、一部のノードがネットワークから孤立する場合がある。この場合、ノード自体は故障していないため、センサデータの取得は可能である。このようなノードにおいて取得され

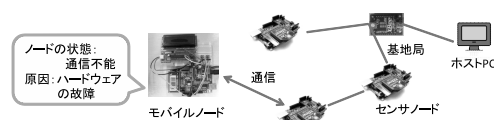


図1 システム構成

たセンサデータもできるだけ活用したい。

- 障害の原因の特定：あるノードからデータが取得できなくなった場合、ノード本体の故障だけでなく、バッテリーの消耗、通信デバイスの故障、センサデバイスの故障、周囲の環境を原因とする通信障害といった様々な原因が考えられるが、その特定が困難である。

3 管理専用ノードを用いたネットワーク管理手法

センサネットワークの管理は、ホスト PC における機能のみでは不十分である。従って、設置現場における管理作業を支援するための新たなデバイスとして、管理専用ノードであるモバイルノードの導入を提案する。

3.1 概要

図1に示すように、提案するシステムにはホスト PC、基地局、センサノード、モバイルノードがある。ホスト PC は、センサデータの集計とネットワークトポロジの管理を行う。基地局は、受信したデータをホスト PC に送信する。センサノードは取得したセンサデータを基地局に送信するとともに、残電力や RSSI などのノードの状態情報を定期的に取得し、センサデータと合わせて自身の二次記憶に記録する。このように二次記憶にセンサデータとノードの状態情報を記録することにより、ネットワークからノードが孤立した場合のデータの取りこぼしに対処することができる。

モバイルノードは、現地に設置されたセンサノードと直接通信を行うことで異常のあるノードの発見、故障原因の特定、異常ノード修理後の正常動作の確認などが可能であり、管理をサポートする役割を担う。

3.2 モバイルノードの機能

モバイルノードの機能としてノードの状態情報の取得、ノードの発見、過去のセンサデータの取得、トポロジ情報の表示の4つがある。ノードの状態情報には、PAN ID、送信先アドレス、自身のアドレス、親ノードのアドレス、残電力、RSSI、ネットワークへの参加状

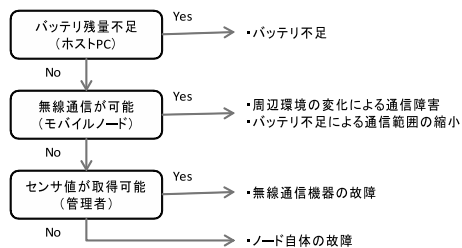


図2 問題の切り分け

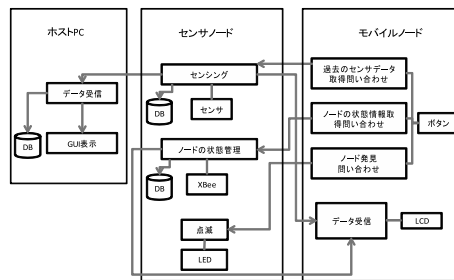


図3 ソフトウェア構成

況が含まれる．これらの機能を利用することで現場におけるノードやネットワークの管理作業が簡便化される．例えば，Host PC上で，あるノードからの通信が途絶えた場合，そのノードの場所と通信が途絶えた原因を探り，メンテナンスや修理を行う必要がある．まず対象となるノードを見つけるためにノードの発見機能を用い，通信可能な範囲にある全ノードのLEDを点滅させることで，点滅しないノードが通信が途絶えたノードであると判断することができる．次に原因を特定するために，図2に示す手順で問題の切り分けを行う．これにより，何が故障の原因であり，どこを修理すればよいかの判断を行うことができる．また，センシングは行っているが，故障などの原因により送信できていないセンサデータに関しては二次記憶上に保存されたセンサデータを取得することで活用できる．

修理したノードが正常に動作していることを確認するためには，現場において，そのノードの状態情報と過去のセンサデータを取得する操作を実施する．

その他のモバイルノードの使い方として，Host PCで管理しているトポロジ情報を持ち出すことで，実際の現場で無駄のないトポロジになっているか確認することができる．

4 モバイルノードとセンサノードの設計

Arduinoを用いてプロトタイプシステムの開発を行った．Arduinoはフィジカルコンピューティングのために開発されたオープンソースハードウェアであり，ハードウェアの拡張性に優れている．Arduinoに無線通信機器であるXBeeを接続することで無線通信を可能にしており，外部バッテリーを接続することで単体で動作させることが可能である．モバイルノードはXBee，バッテリーの他に各種機能を実行するためのタクトスイッチとデータを表示するためのLCDを持つ．センサノードはXBee，バッテリーの他にセンサデータとノードの状態情報を記録するためのSDカードと自身の位置を知らせるためのLEDを持つ．

システムのソフトウェア構成を図3に示す．モバイルノードには，各機能の問合せのモジュールと問合せに対する応答を表示させるモジュールがある．ノードの状態情報の取得と過去のセンサデータの取得は，センサノードに記録されているものすべてを取得する．センサノード

には，センシングモジュール，ノードの状態管理モジュール，点滅モジュールがある．センシングモジュールは定期的にセンシングを行い，そのデータをHost PCに送信し，SDカードに記録する．記録内容はプログラムが起動してからの時間，ノードID，センサデータである．ノードの状態管理モジュールは定期的に状態情報を取得し，ノードが起動してからの時間とともにSDカードに記録する．点滅モジュールはモバイルノードからノード発見問合せがあった場合にLEDを10秒間点滅させる．Host PCでは，各ノードからのセンシングデータのGUI表示とデータベースへの保存を行う．

5 システムの実装と評価

Host PC上にデータのGUI表示，およびデータベースへの保存を行うツールを実装し，システム全体の動作確認を行った．GUI表示では，ノードID，バッテリー残量，センサデータを各ノードごとに表示する．

モバイルノードとセンサノードの機能の評価として次の作業を行った．小規模なセンサネットワークを構築し，ノードの状態情報の取得機能と過去のセンサデータ取得機能が正常に動作していることが確認できたが，データの送信速度に対し通信速度が遅かったため，モバイルノードの受信データの取りこぼしがあった．

6 おわりに

本稿では，センサネットワークの管理における課題について述べ，課題解決の方法としてモバイルノードの導入を提案し，実装と評価を行った．その結果，現場での管理のサポートとしてノードの発見や，ネットワークから孤立したノードへの対処が可能となったことが確認できた．今後の課題として，モバイルノードの受信データの取りこぼしの対処や受信データの表示方法の検討などが挙げられる．

参考文献

- [1] Turon, M., MOTE-VIEW: A Sensor Network Monitoring and Management Tool, 2nd IEEE Workshop on Embedded Networked Sensors (EmNetS-II), pp.11-18, May 2005.
- [2] Arduino: <http://arduino.cc/>