

IoT における位置情報に基づくセンシングデータ補間手法

日大生産工(学部) ○吉澤 柊

日大生産工 新井 雅之

1. はじめに

IoT (Internet of Things) とは様々なモノに通信機能を搭載してインターネットに接続し、相互通信によって自動認識や自動制御、遠隔計測などを行う技術である。IoT ではセンシングデータの送受信が行われるが、ノード故障により、一部のデータが送信されない可能性がある。文献[1]では、IoT の普及による重要インフラのセキュリティの要件として、可用性確保の為の問題検知から暫定対策までのアプローチを報告している。IoT における無線センサノード間通信の耐故障設計手法として、神山は m-out-of-n システムを設計している[2]。

本研究では位置情報に基づくノード故障におけるデータ補間手法について検討する。シンクノードは複数のセンサノードから位置情報と温湿度データを取得する。一部のノードが故障した場合、他のノードのデータを用いて故障ノードデータの補間を行う。Arduino, XBee, Pozyxを用いてシンクノード、センサノードを実装し、動作確認を行った。

2. 準備

2-1. IoT及びIoT機器

従来のインターネットでは、ユーザである人間がデバイスに明示的に入力や動作指示を行っていた。IoT ではモノにセンサを搭載させ、センサからインターネットを通じて情報が自動的に集約されるようになり、人間が能動的に操作する必要がない。IoT を利用することで、生活や仕事の快適化に繋がる[1]。

IoT 機器の例をいくつか以下に挙げる[3]。

“JINS MEME” はオフィスワーカーの目の動きや瞬きをデータ化し、集中の度合いを可視化することを目的としたセンシング・アイウェアとして IoT 技術を利用している。また、ペットの鳴き声や動きを感知し、飼い主の携帯への通知や、スマホアプリを通じてリアルタイムでペッ

トの行動を見ることが可能なコミュニケーションデバイス “Furbo” などが開発されている。

2-2. シングルボードコンピュータと無線通信規格及びハードウェアソリューション

以下で、本研究で用いた技術について概説する。

(a) Arduino

ArduinoはAVRマイコンと入出力ポートを備えた基盤を持つ、ワンボードマイコンである[4]。C言語に似たArduino言語による総合開発環境から構成されている。安価で様々なモジュールシールドを簡単に追加できるため、評価用IoTデバイスの開発基盤として適している。

(b) ZigBeeとXBee

ZigBee は無線センサネットワーク用プロトコルのひとつである[5]。転送可能距離が短く、転送速度も低速であるが、安価で消費電力が少ないというメリットを持つ。XBee は ZigBee 規格に対応した無線モジュールのひとつである。

(c) Pozyx

Pozyx システムは、正確な測位及び動作状況を提供するハードウェアソリューションである[6]。室内の4箇所に設置されたアンカーデバイスから送信される電波の到来時間の差に基づいて、タグは自身の位置を特定する。信号方式として Ultra Wide Band (UWB)を採用しており、近距離に限られるものの、高速でGPSよりも正確な位置測定が可能である。

3. 評価システム的设计

位置情報に基づく IoT センシングデータ補間システムの設計を行った。

図1にシステム構成と各モジュールの配置例

Method for IoT Sensing Data Interpolation Based on Position Information
Shu YOSHIZAWA and Masayuki ARAI

を示す。アンカーデバイスを部屋の4箇所に設置する。アンカーデバイスの内側に複数のセンサノードと1個のシンクノードを配置する。

図2にセンサノード及びシンクノードの詳細を示す。センサノードはArduinoにタグ、XBee, 温湿度センサを接続したものである。シンクノードはArduinoにXBeeを接続したものである。センサノードはタグの位置情報と温湿度データを取得し、データをシンクノードへ1分毎にXBeeで通信する。シンクノードは各センサノードから受信したデータを表示し、またセンサノードの位置情報を記憶しておく。

あるセンサノードから一定時間受信しなかった場合、以前に受け取っていた位置情報と他のセンサノードのデータに基づいてセンシングデータの補間をする。

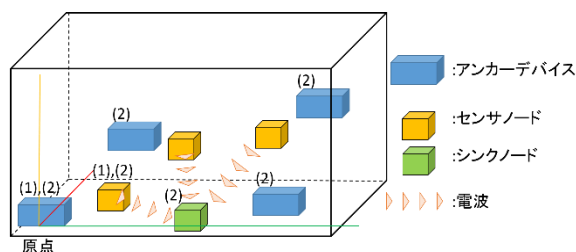


図1 Pozyxの実装図

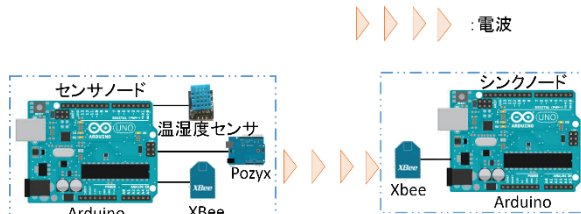


図2 センサノードとシンクノードの接続図

4. 提案システムの実装と動作検証

4-1. Pozyxの動作確認

まず、図1で(1)として示したアンカーデバイス1機とセンサノード1機からなる実験環境を構築し、動作確認を行った。

センサノードとしてArduinoとタグを接続した。センサノードのプログラムでは、使用したアンカーデバイスとタグのIDを設定した。Arduinoとアンカーデバイスの距離をArduino IDEで表示し、動作確認を行った。

4-2. 位置情報の送受信

次に、図1で(2)として示したアンカーデバイスを4機設置してセンサノードの位置情報を取得し、表示させるプログラムを作成した。

アンカーデバイスを部屋の4箇所に置き、1箇所を原点(0,0,0)として、各アンカーデバイスのIDとx, y, z座標をセンサノードのプログラムに設定した。タグの位置情報、温湿度データをシンクノードに送信するセンサノードのプログラムと、受信したデータを表示させるシンクノードのプログラムを作成した。各ノードのプログラムを実行し、シンクノードの表示を確認した。実行結果を図3に示す。シンクノードにおいてx, y, z座標が正しく表示されることを確認した。

COM9 (Arduino/Genui	COM5 (Arduino/Genui Uni
Date! Position: X:1291, Y:1343, Z:2702	POS ID 0x0, x(mm): 1240, y(mm): 1334, z(mm): 2739
Date! Position: X:1225, Y:1317, Z:2758	<PX:1240, Y:1334, Z:2739>
Date! Position: X:1246, Y:1322, Z:2724	POS ID 0x0, x(mm): 1239, y(mm): 1328, z(mm): 2762
Date! Position: X:1240, Y:1334, Z:2739	<PX:1239, Y:1328, Z:2762>
Date! Position: X:1239, Y:1328, Z:2762	POS ID 0x0, x(mm): 1256, y(mm): 1348, z(mm): 2749
Date! Position: X:1245, Y:1315, Z:2800	<PX:1256, Y:1348, Z:2749>
Date! Position: X:1272, Y:1323, Z:2749	POS ID 0x0, x(mm): 1254, y(mm): 1330, z(mm): 2771
Date! Position: X:1262, Y:1333, Z:2724	<PX:1254, Y:1330, Z:2771>
Date! Position: X:1261, Y:1336, Z:2710	POS ID 0x0, x(mm): 1245, y(mm): 1315, z(mm): 2800
	<PX:1245, Y:1315, Z:2800>
	POS ID 0x0, x(mm): 1272, y(mm): 1323, z(mm): 2749
	<PX:1272, Y:1323, Z:2749>
	POS ID 0x0, x(mm): 1262, y(mm): 1333, z(mm): 2724
	<PX:1262, Y:1333, Z:2724>
	POS ID 0x0, x(mm): 1261, y(mm): 1336, z(mm): 2710
	<PX:1261, Y:1336, Z:2710>

図3 実行後のシリアルモニタ(左:シンクノード, 右:センサノード)

5. まとめと今後の課題

本研究では、IoTにおける位置情報に基づく無線センサノード間通信の耐故障設計手法の検討として、データ補間手法を提案した。また、評価システムを実装し、位置情報を表示させることによって動作確認を行った。今後は、複数のセンサノードからのデータに基づく補間システムを作成する予定である。

参考文献

- [1] 田中晋輔 他, “IoTシステムのセキュリティ課題と解決アプローチ,” 日立評論, Vol. 98, No. 6, pp. 437-440, 2016 年.
- [2] 神山晴彦, “IoT環境における位置情報を考慮した耐センサノード故障システム設計,” 日本大学生産工学部数理情報工学科提出卒業論文, 2018 年 1 月 26 日.
- [3] IoT デバイスマップ 2018, <https://robotstart.co.jp/press46.pdf>
- [4] Arduino HP, <https://www.arduino.cc/>
- [5] ZigBee HP, <http://www.zigbee.org/>
- [6] Pozyx HP, <https://www.pozyx.io/>