

IoT 環境におけるセンサノード間通信の耐故障設計

日大生産工(学部) ○神山 晴彦
日大生産工 新井 雅之

1. はじめに

IoT (Internet of Things) の重要な要素技術の一つとして、センシングデータの自動的な送受信が挙げられる[1,2]. しかし、ノード故障によるセンサ情報の欠落が問題となる可能性がある. 文献[1]では重要インフラ向け IoT セキュリティの要件として可用性に着目し、問題検知から暫定対策までを報告している.

本研究では、多様な機種を用いた IoT における無線センサノード間通信の耐故障設計手法の検討を行う. 手法として m-out-of-n システムを用い、一部のノードが故障しても他のノードが対応するシステムを構築する. 親機として Raspberry Pi, センサノードとして Arduino, 通信には XBee を用いて、実機による実験環境を実装し、その環境下で動作確認を行う.

2. 関連研究と準備

2-1. IoTに関する論文の報告

IoT (Internet of Things) とは、センサやデバイスといったモノがインターネットを通じてクラウドやサーバに接続され、情報交換することにより相互に制御する仕組みのことである. センシングデータの自動送受信の応用が期待されている. 佐藤らは、パワーエレクトロニクスに向けたワイヤレスセンサのデータ伝送について報告している[2].

重要インフラも IoT の適用領域となりつつある[1]. IoT システムへのセキュリティ対策は重要度が高まっている. 一方、接続されるデバイスの数が飛躍的に増加することで、影響範囲の拡大という技術面の課題に加え、セキュリティ運用管理者の不足といった問題が生じると考えられる. 課題解決へのアプローチとして文献[1]では、センサとクラウドの間に中継装置を設置し、大量のセンサ情報を中継装置で集約する階層化アーキテクチャを導入している.

2-2. シングルボードコンピュータと無線通信規格

以下で、本研究で用いた実機を解説する.

(a) Raspberry Pi

Raspberry Pi とは ARM プロセッサを搭載し、Linux が動作するシングルボードコンピュータの一種である[3]. HDMI 出力, USB ポート, LAN ポートを持つため、直接機器本体に接続し、ネットワーク環境の構築、サーバでの数値集計や制御を行う事が可能である. そのため、IoT, 特にセンサを用いた環境で活躍する.

(b) Arduino

Arduino とは、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の一種である[4]. C 言語に似た、Arduino 言語による統合開発環境を備えている. 単独でスタンドアロン型のインタラクティブデバイスを開発できるが、ホストコンピュータ上のソフトウェアで制御が可能で、柔軟で使いやすい電子プロトタイピング・プラットフォームである.

(c) XBee

ZigBee は、センサネットワークを主目的とする近距離無線通信規格である[5]. この通信規格は、転送可能距離が短く転送速度が低速な反面、安価で消費電力が少ない特徴を持つ.

今回は、ZigBee 規格に対応した無線モジュールのひとつである、XBee を用いる.

3. 評価システムの設計

Arduino に温湿度センサを搭載し、XBee を用いて通信する実験環境を構築する. 図1に、システム構成及び通信例を示す.

Arduino に温湿度センサ DHT11 と XBee ZB S2C モジュールを接続する. これを一つのノードとし、二組用意する. Raspberry Pi にも XBee ZB S2C を接続する. Arduino で計測した温湿度

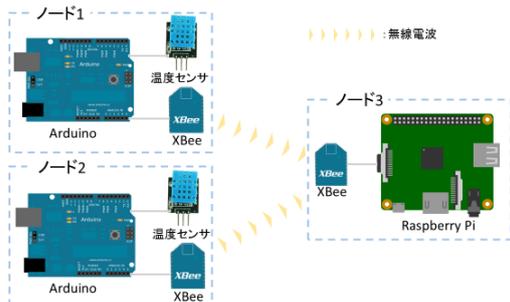


図 1. XBee を用いた無線センサノード間通信
データを, XBee を用いて Raspberry Pi に送り表示する.

各ノードは, 1 分間に 1 回センサ情報を送信する. 一定期間送信がない場合, 生存しているノードの情報を用いてデータの補完を行う. m-out-of-n システムの詳細については, 現在設計中である.

4. 実装, 動作検証

4-1. Arduino側の通信設定

(a) XBeeの設定, 実装

XBee USB インターフェースボードにはんだ付けを行い, XBee を USB 接続出来るようにした. デフォルトの XCTU には XBee ZB S2C のファームウェアが備わっていない為, 公式サイトから対応しているデータを手し, アップデートを行った. 通信の確認は, Arduino を用いて行った.

図 2 に, 動作確認の環境を示す. XBee をそれぞれ Coordinator と Router に設定し, PAN ID を揃え, API モードに統一した. スイッチと XBee を接続した Arduino と, LED と XBee を接続した Arduino を用意した. Arduino IDE を用いて, それぞれのプログラムを作成した. スイッチ側のプログラムは, スイッチが押されると特定の文字を送信するものとした. LED 側のプログラ

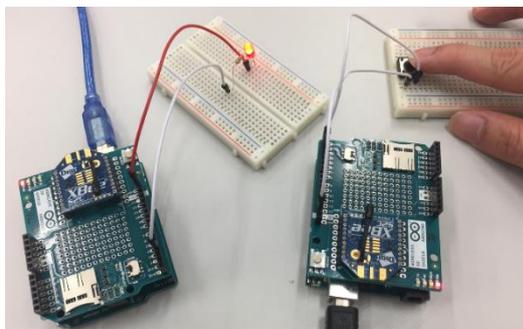


図 2. XBee を用いた Arduino 同士の通信動作確認

ムは, 特定の文字を受け取ると, 1 秒間 LED を点灯させ消灯させるものとした. 記述したプログラムを動作させ, XBee 同士の無線通信を確認した.

(b) 温度データの取得

Arduino に DHT 用のライブラリをインストールし, 計測した温湿度を 3 秒毎に表示するプログラムを作成した. シリアルモニタを用いて, 動作確認を行った.

4-2. Raspberry Piの設定及びネットワーク通信

親機である Raspberry Pi に OS のインストールを行い, OS の初期設定, 環境設定を行った. Python を扱える環境にし, LED を光らせるプログラムを作成した. 実際に LED を光らせ, プログラムの動作確認を行った. 次に, もう一台同設定の Raspberry Pi を用意した. 片方をクライアント, もう片方をサーバに見立てて, TCP/IP 通信を行った. 内容は, クライアント側の Raspberry Pi に文字を入力すると, サーバ側に表示され, その文字がクライアントに返信されるというものである. 実行し, 送受信の成功を確認した.

現在, Raspberry Pi と XBee モジュールを用いた通信環境の構築を作業中である.

5. まとめ, 今後の課題

本研究では, 多様な機種を用いた IoT における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, m-out-of-n システムを使用した. 今後は, Arduino に接続した温度センサから取得したデータを Raspberry Pi に送り, 表示可能にする予定である. また, 評価システムを用いて, 故障が発生した際の可用性を評価する予定である.

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] 佐藤龍太郎, 和田圭二, “パワーエレクトロニクス用ワイヤレスセンサのデータ伝送の検討,” 電気学会全国大会プログラム, Paper 4-076, p. 129, 2017 年.
- [3] Raspberry Pi HP, <https://www.raspberrypi.org/>
- [4] Arduino HP, <https://www.arduino.cc/>
- [5] ZigBee HP, <http://www.zigbee.org/>