

地盤沈下観測における IoT の適用  
Application of IoT for Monitoring of Land Subsidence  
河原和好<sup>1</sup>

## 要旨

IoT (Internet Of Things : モノのインターネット) は、様々な「モノ」がセンサと無線通信を介してインターネットと接続される。自動車、家電、ロボット、施設など、あらゆる「モノ」がインターネットにつながり、情報のやり取りをすることによる、「モノ」のデータ化や、それに基づく自動化等を進展させ、新たな付加価値等を生み出し、今後の発展も見込まれている。

この IoT を地盤沈下監視分野への適用可能性を検証するための、新潟県保健環境科学研究所情報調査科と共同研究を 2017 年度から行っており、これまで研究ノート及び論文として新潟国際情報大学経営情報学部紀要にて報告した。

本研究では現時点での共同研究の成果報告をするともに、今後の課題について述べる。

キーワード : IoT、地盤沈下、Arduino、3GIM モジュール、省電力化、RTC

## 1. はじめに

IoT (Internet Of Things : モノのインターネット) という言葉を目にする機会が多くなっている。総務省情報通信白書[1]によると、2018 年時点で IoT デバイス数は 300 億を超え、今後も拡大していくと予想されている (図 1)。

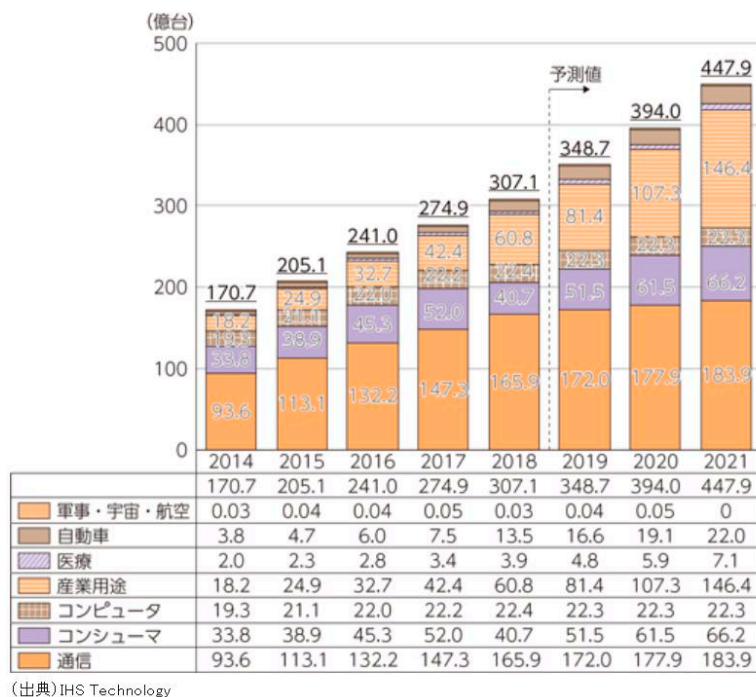


図 1 : 世界の IoT デバイス数の推移及び予測[1]

<sup>1</sup> Kazuyoshi Kawahara 情報システム学科

本学においては、情報処理演習 H において、マイコンボード Raspberry Pi を用いた演習を行うなど IoT を用いた学習に対応している。また、私の研究室においても、Raspberry Pi や後述する Arduino を用いた卒業研究を行っており、2016 年度には新潟県 IT&ITS 推進協議会が主催する「にいがた暮らし IoT アイデアコンテスト 2016」[2]に応募し、書類審査を通過しデモンストレーションを行った[3]。この際、新潟県保健環境科学研究所[4]（以下、保環研）の方と接点を持つことができ、提案を頂いたことで、地盤沈下の監視に IoT を使うための実証実験の共同研究を行うこととなった。本学紀要における研究ノート[5]としての報告及び 2017 年度全環研北海道・東北支部 IoT の環境計測への適用セミナーにおいて発表[6]を行い、2018 年度は本学紀要[7]で論文として成果報告を行った。

本稿では、2019 年度において引き続き得られた共同研究の成果について報告し、さらに今後の課題と展開について報告する。

## 2. IoT

IoT とは、Internet of Things の略語であり「モノのインターネット」と訳される[8,9]。基本的に、あらゆる「モノ」が対象であるが、「何をどのようにネットワークに接続するか」を考える必要がある。そして、「モノの接続」によって、そこに何らかの価値が生み出される。応用分野としては、工場などの製造分野や流通分野、自動車産業、ヘルスケア部門、家庭内など、多岐にわたっている。

概念はインターネット黎明期から存在していたが、技術の進歩に伴い、「マイコン」および「マイコンボード」「センサ」の高性能化・低価格化、通信環境などのインフラの整備及び収集データを解析するためのクラウドの整備により、IoT を実現できるようになった。

要求される機能に応じて様々な API(Application Interface)が必要となり、メール送受信、地図へのマッピング、グラフ作成など、様々な API が提供されている。

## 3. 地盤沈下とその観測

地下水をくみ上げることにより地盤沈下は引き起こされるが、工業用、上水道用とともに、近年では消雪用の地下水くみ上げが原因となっている[10]。新潟県においては、上越地域において顕著となっており、昭和 59 年から 61 年において 3 年連続の豪雪に見舞われたため、消雪用地下水のくみ上げが増加し、全国ワースト 1 位から 3 位の地盤沈下が生じた。このため、従来の工業用や上水道の規制から消雪用を重点とした地盤沈下対策が新潟県において展開されることとなった。同様に、消雪用くみ上げが多い地域である長岡地域、柏崎地域、南魚沼地域でも地盤沈下が見られたため観測が行われている。新潟・新発田地域では、昭和 30 年代には、年間最大沈下量が 54cm にも及ぶ著しい地盤沈下が起こったが、水溶性天然ガスの採取規制の実施などによって、昭和 50 年代以降は一部の地域を除いて全体的に沈静化してきている。しかしながら新潟地域は、信濃川や阿賀野川などの大河川や内陸部の広大な農地を有し、しかも広範囲のゼロメートル地帯を有する地域であることから、地盤沈下対策を継続していく必要がある。

保環研においても、県内において地盤沈下が生じた地域に設置してある地盤沈下観測井を監視し、地下水位及び地層収縮量のデータ集計を行い、その原因調査や地盤沈下防止のための調査を行っている。現在の地盤沈下観測の体制は図 3-1、地盤沈下観測性の仕組みは図 3-2 となっている。地盤沈下観測井で観測されたデータは SD カード等に記録され、当該データを保環研に送付

しており、回収されたデータを解析・集計し、年度ごとに公表している。

## 保健環境科学研究所

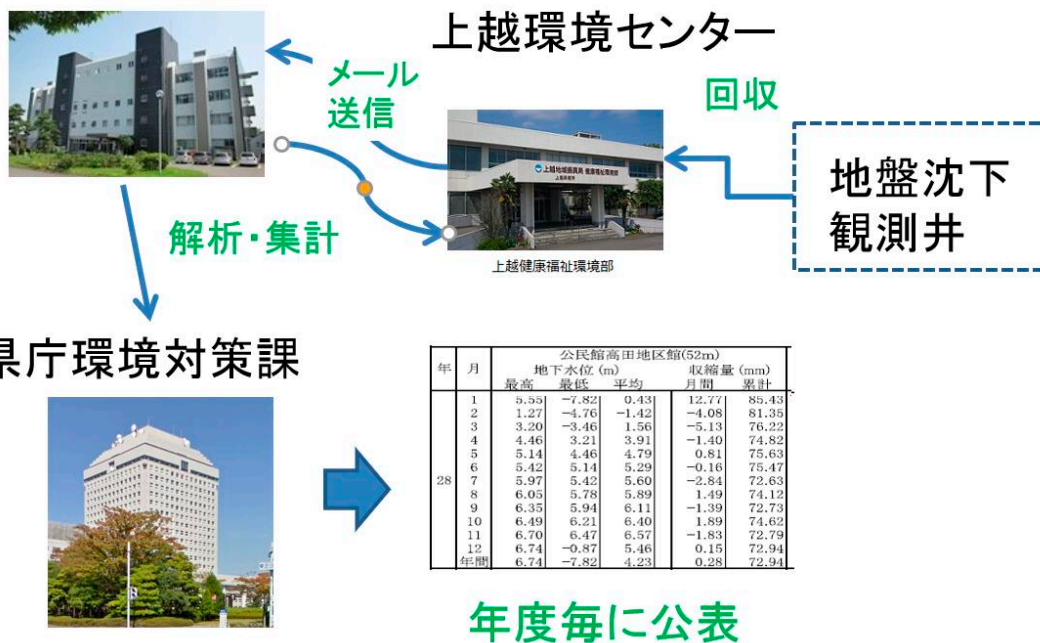


図 3-1 地盤沈下観測の現状

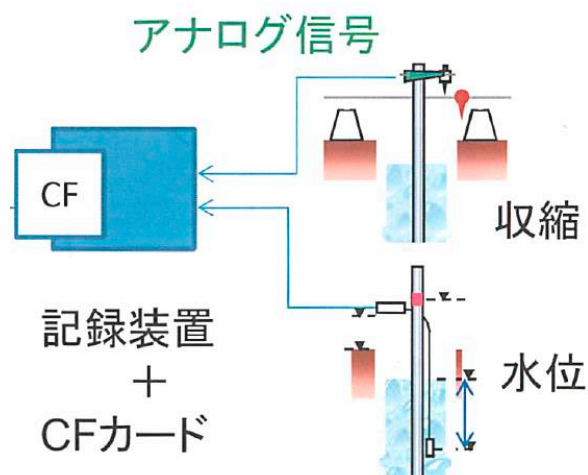


図 3-2：地盤沈下観測井の仕組み

地盤沈下観測井では、水位センサと収縮センサを用いて地下水位を計測している。それぞれのセンサは以下のとおりである（図 3-3）。

水位センサ[11]

- ・ 形式 ノースワン(株) KDC-S10-STM/N20-30
- ・ 測定推進スパン 20m
- ・ 出力電圧 50mV

収縮センサ[12]

- ・ 形式 (株)共和電業 DTH-A-50
- ・ 測定収縮スパン 50mm
- ・ 出力電圧 20mV 以下



図 3-3：水位センサ(左)[11]および収縮センサ(右)[12]

#### 4 本研究の内容

現在の地盤沈下観測における課題として、観測井で観測されたデータを回収する際に人手がかかる点と、それにより月次処理やリアルタイム処理ができていない点が挙げられる。この点に対し、本研究では IoT を用いた解決方法を提案し実施を試みる。

これまでの方式を改良し、観測データを IoT により Web サーバ等へ送信する方法を提案し実証する（図 4-1）。センサにより地盤沈下を観測し、そのデータをマイコンボードに入力して処理し、3G 回線を通してサーバに転送する。これにより、保環研においてリアルタイム監視を行うことが可能となる。

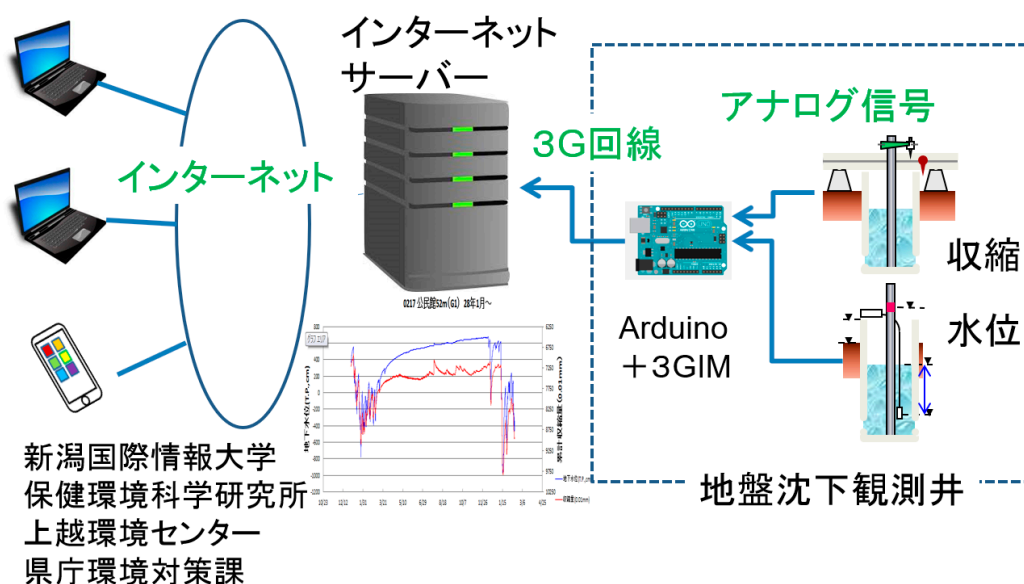


図 4-1：実証実験の仕組み

## 4.1 ハードウェア

IoT における要素であるマイコン、センサ、通信技術について、本研究で使用したものを説明する。

### 4.1.1 マイコン

Atmel「AVR シリーズ」であり、普及している「Arduino」(アルドゥイーノ)を用いた[13]。オープンソースで開発される小型マイコンボードおよびその開発環境を含めたシステムであり、シンプルな開発環境である「Arduino IDE」や様々なライブラリを用いることが出来る。「シールド」とよばれる機能拡張ボードを用いることで、簡単に機能の拡張が可能である。実験用環境としては、もっとも基本である「Arduino Uno」の互換ボードを使用した。

入出力端子があり、デジタル出力、アナログ出力、デジタル入力、アナログ入力、シリアル通信が可能である。Raspberry Pi も検討したが、アナログ信号を取り扱えることと、後述する商品電力の関係から Arduino を用いることとした。

### 4.1.2 センサ

以下の仕様の収縮（沈下）計を用いて実験を行った。マイコンボードとの接続は図 4-2 の通りである。

(株)日さく NLS2[14]

- ・ 測定範囲 0～100mm
- ・ 印加電圧 5V(DC)以内
- ・ 出力電圧 0～5V 以内
- ・ 抵抗値 5k $\Omega$
- ・ 測定精度  $\pm 0.1\text{mm}$  以内

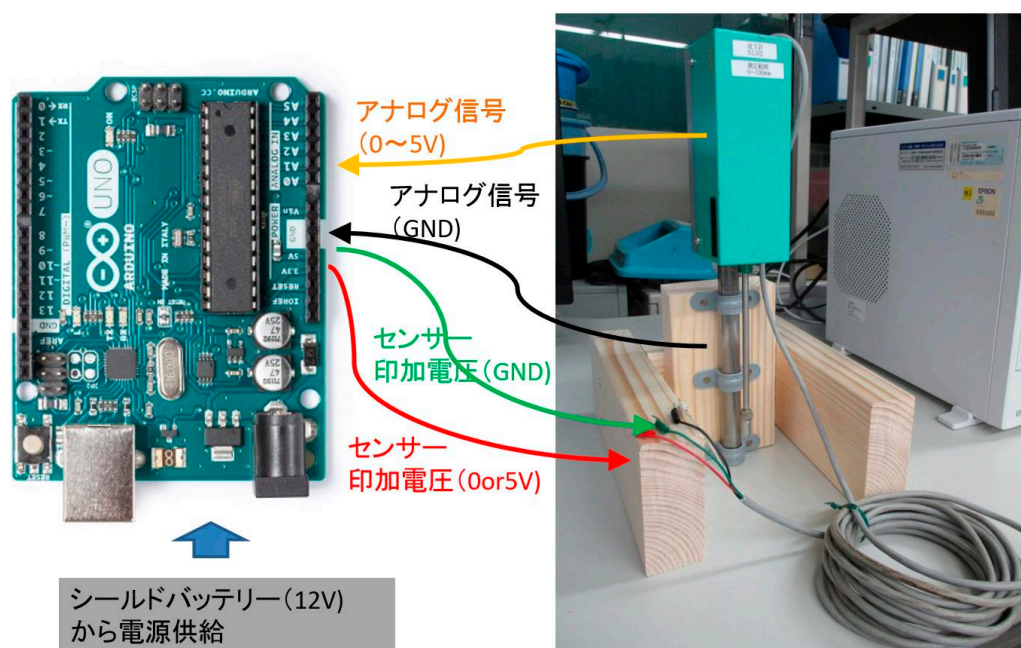


図 4-2 : 収縮計（右）[14]とマイコンボードの接続



### 4.1.3 通信技術

3GIM V2.1 モジュール[15]および 3GIM シールド[16]を Arduino に接続することにより、3G 通信できるようになる (図 4-3)。

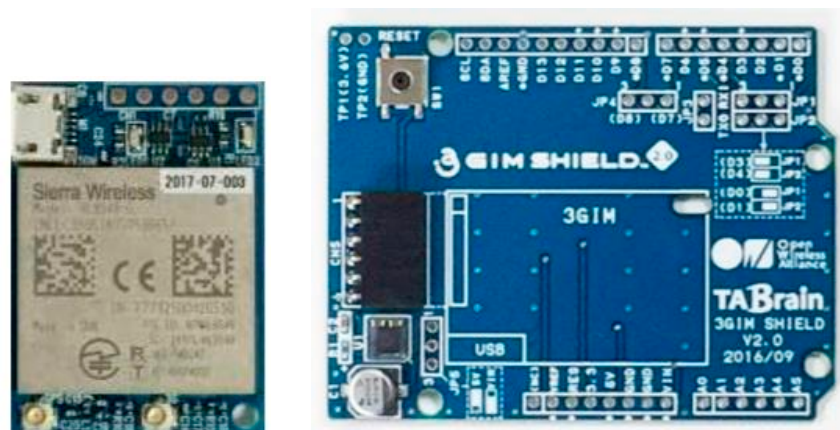


図 4-3 : 3GIM モジュール[15]及び 3GIM シールド[16]

実際の通信には SIM カードが必要となるが、NifMo (ニフモ) [17]を採用した。これはインターネットサービスプロバイダ大手の Nifty が提供する、MVNO のサービスであり、NTT ドコモの通信回線を借り受けて提供されている。LTE に対応しており、上り最大 50Mbps、下り最大 150Mbps の高速通信が可能となっている。本研究では高速及び大容量の通信が必要ではないため、スタートプラン：480 円／月（通信速度：常時 200kbps、月間データ通信容量：2.1GB）を契約した。

実際の観測データは測定日時と観測データであり、テキストデータとすると約 30 バイトである (図 4-9 で後述)。送信間隔は 10 分程度としているため、1 日の送信データは  $30[\text{バイト/回}] \times 6[\text{回/時間}] \times 24[\text{時間}] = 4,320$  バイトとなり、1 ヶ月では最大  $288 \times 3 = 133,920$  バイト  $\approx 130$  メガバイトとなる。

## 4.2 ソフトウェア

センサから入力を読み込み、3GIM モジュールでサーバ等に送信するプログラム作成には、前述の Arduino IDE を用いている。シンプルで扱いやすい IDE (統合開発環境) で、プログラミング言語は C 言語をベースにした独自の専用言語である。

データの閲覧については IoT データ可視化サービスである Ambient[18]を用いた (図 4-4)。仕様は以下の通りであり、本研究で必要な仕様を満たしている。無料で使用可能である。

- ・ 1 ユーザーあたり 8 個までチャンネルを生成可能
- ・ 1 チャンネルあたり 8 種類のデータを送信可能
- ・ 送信から次の送信まではチャンネルごとに最低 5 秒空ける必要がある (それより短い間隔で送信したものは無視される)
- ・ 1 チャンネルあたり 1 日 3,000 件までデータを登録可能 (平均 28.8 秒に 1 回)
- ・ 件数のカウントは 0 時に 0 クリアされる
- ・ 1 チャンネルあたり 8 個までチャートを生成できる

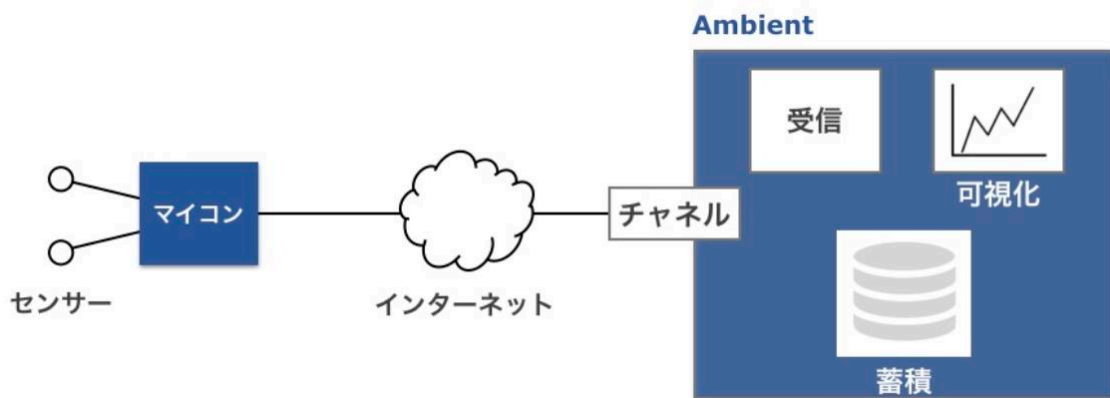
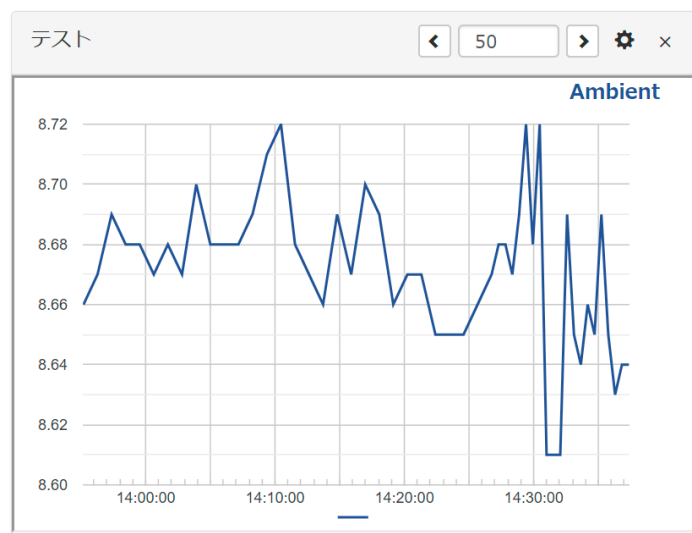


図 4-4 : Ambient のしくみ[18]

### 4.3 これまでの成果

最初の成果として、Arduino と 3GIM モジュールを接続し、アナログデータを観測して送信し、Ambient を用いて観測する一連の動作を確認した。図 4-5 が観測した Ambient のグラフである。



AmbientData Inc.

図 4-5 : Ambient のグラフ

#### 4.3.1 スリープによる消費電力の軽減

前述の通りデータを計測して送信し観測する仕組みは完成しているため、実際の観測井に設置すればシステムを稼働させることができる。しかし、観測井には電源が無く、バッテリー駆動となり、消費電力の関係からこのままでは連続した稼働が難しい。

バッテリーとしては 12V で 7.2Ah のシールド（密閉型）バッテリー [19]を用いており、一般的な Arduino の消費電力は 25mA であるため、このままでは連続稼働は  $7200[\text{mAh}] \div 25[\text{mA}] = 288 \text{ 時間} = 12 \text{ 日}$  となる。実際には 3GIM モジュールを用いて通信をしているため、さらに消費電力は高くなることが予想され、これでは交換の手順が半月ごとにかかってしまう。前述した人手がかかるという問題点の解消にはならず、月次処理やリアルタイム処理にも支障をきたしてしま

う。

Arduino にはスリープ機能があり（図 4-6）[20]、これを用いることで消費電力を抑えることができるため、その機能をプログラムに組み込むことが必要となる。

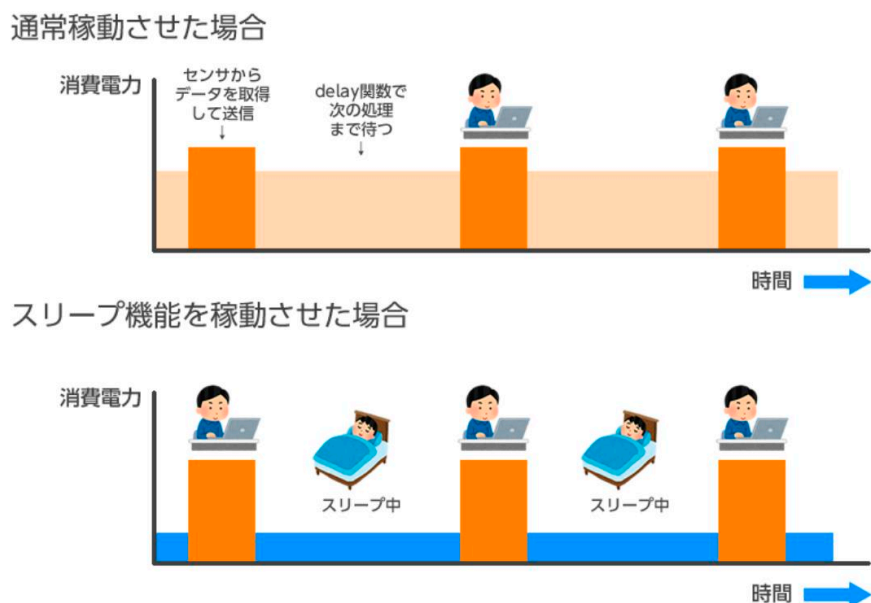


図 4-6：Arduino におけるスリープ機能のイメージ[20]

Arduino には 5 種類のスリープ機能があり（図 4-7）、種類によって省エネ効率や復帰方法が異なる。本研究ではもっとも効率が高い「SLEEP\_MODE\_PWR\_DOWN」を採用した。

モード	説明	省エネ効率
SLEEP_MODE_PWR_IDLE	アイドル状態、システムクロックは停止するが、内臓タイマーや外部割込み、シリアルポートなどの機能は動作します。復帰方法は外部割込み、ウォッチドッグタイマー、ADCの入力変化、リセットピンによる復帰が可能	普通
SLEEP_MODE_PWR_ADC	AD変換ノイズ低減用に利用されます(ADC対応のAVRでないと利用不可)	—
SLEEP_MODE_PWR_DOWN	パワーダウンモード、最低限の動作のみのもっとも消費電力が少なくなるモード。外部割込み、ウォッチドッグタイマー、リセットピンによる復帰が可能。	もっとも高い
SLEEP_MODE_PWR_SAVE	パワーセーブモード、タイマー用の外部発信器は動作しているので、外部割込み、ウォッチドッグタイマー、リセットピンによる復帰が可能。	高い
SLEEP_MODE_PWR_STANDBY	スタンバイモード、パワーダウンと比べて、メインクロックが動作しているので復帰が少しだけ早いモードです。	かなり高い

図 4-7：Arduino におけるスリープ機能の種類[20]



また、スリープから復帰する際には Arduino のタイマー機能である「ウォッチドッグタイマー」を使用した。これは文字通り「番犬」を意味するタイマーで、コンピュータの動作を監視する役割のタイマーである。これを複数回繰り返すことで、最大 8 秒間の省電力なタイマーを実現した例を参考[21]にし、これを複数回繰り返すことで本来の計測タイミングである 10 分間おきに計測とスリープを繰り返すプログラムを作成した。

#### 4.3.2 RTC モジュールの追加

また、4.3.1 項の動作では、起動した時刻から 10 分おきの観測及びデータ送信となるため、観測のタイミングが正式に 10 分毎とはならない。これを解決するには、SIM の通信機能を用いて時間を取得し、その時刻を保持してタイミングを合わせて観測する必要がある。そのため、Arduino に時計機能を持たせる部品として RTC (Real Time Clock) モジュール (図 4-8) [22]を追加し、プログラムを作成した。



図 4-8 : DS1307 リアルタイムクロック(RTC)モジュール[22]

起動した際に SIM で通信を行って現在時刻を取得し、それを RTC に設定する。「00 分」「10 分」「20 分」のように、分の 1 桁台が 0 になるまで待ち、計測及びデータ送信を行う。これを前述のスリープを用いて 10 分おきに繰り返す。これにより、省電力なデータ計測及び通信を行うことが可能となった。

2020-01-09T00:52:51.154Z	2.14
2020-01-09T01:00:02.622Z	2.14
2020-01-09T01:10:02.619Z	2.14
2020-01-09T01:20:02.621Z	2.14
2020-01-09T01:30:02.616Z	2.14
2020-01-09T01:40:02.848Z	2.14
2020-01-09T01:50:02.340Z	2.14
2020-01-09T02:00:02.627Z	2.14
2020-01-09T02:10:02.660Z	2.14
2020-01-09T02:20:02.690Z	2.14
2020-01-09T02:30:02.619Z	2.14
2020-01-09T02:40:02.690Z	2.14
2020-01-09T02:50:02.440Z	2.13
2020-01-09T03:00:02.454Z	2.14
2020-01-09T03:10:02.470Z	2.14

図 4-9 : 観測データの一部

図 4-9 がその際の送信データである。初回起動で時刻を取得して RTC にセットする。これにより、次回以降は通信を行わずに時刻を取得することが可能となる。通信が不要のため省電力につながる。

前述の通り、0 分台の待機後は 10 分おきにデータが送信されているのが分かる（JST：日本標準時でないため 9 時間の誤差がある）。

### 4.3.3 実際の観測井における計測

保環研の協力により、室内における実験と、実際の観測井（日越小学校観測井）における実験を行った。室内では 2019 年 12 月 5 日から 2019 年 12 月 20 日にかけて、実際の観測井では 2020 年 1 月 8 日から 2020 年 1 月 20 日にかけてデを行った。図 4-10 が室内、図 4-11 が観測井における結果のグラフである。

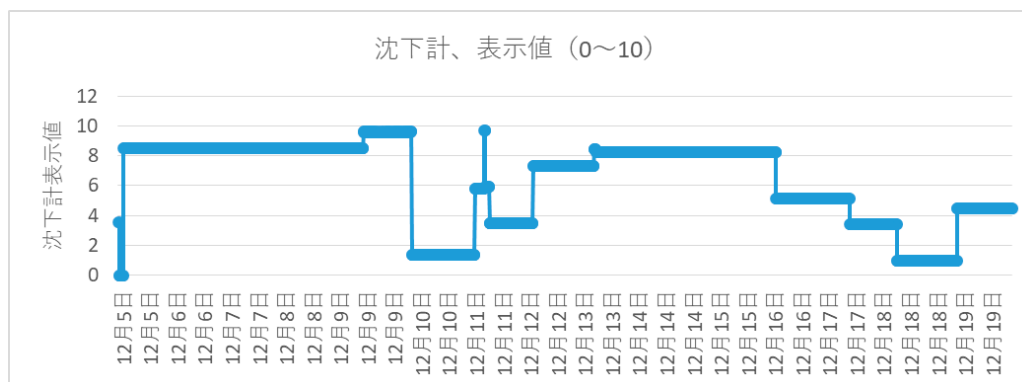


図 4-10：室内における観測データ

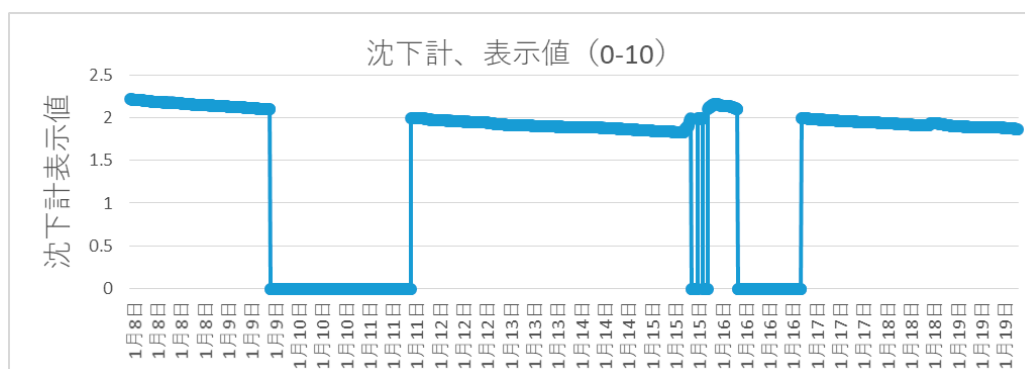


図 4-11：観測井における観測データ

図 4-10 より、室内ではデータ計測及び送信ができており、およそ 10 日程度の稼働が可能であった。しかしながら図 4-11 から分かるように、実際の観測井においては計測できる期間とできない期間が生じてしまった。これに関する原因は、現場の環境や、通信環境の確認により解決する可能性があると思われる。

#### 4.4 今後の課題

前述のように実際の観測井における計測不能な状態が起こる原因の究明が急務である。

また、10 分間隔を 30 分間隔にすることで、より長期の稼働が可能であると考え、それに対する実験を開始する予定である。

さらに、RTC については、もともと 3GIM モジュールに機能として内蔵されている[15,16]ため、実際には RTC モジュールは不要となる。これに対するプログラムの改良により、システムの簡素化とともに、さらなる消費電力の軽減が見込まれる。

#### 5. まとめ

IoT を用いて地盤沈下を観測するためのシステムについて、人手がかかり月次処理が難しいシステムを改良する手法を提案しシステムを作成した。長期の稼働に耐えるように、省電力化の処理を作成し一定の成果は得られた。しかしながら、通信できない状態が生じてしまったため解決する必要がある。また、10 分間隔を 30 分間隔にすること、3GIM モジュール内蔵の RTC 機能を使用することにより、さらに長期間の稼働につながるため、実験を継続していく予定である。

課題点の解決方法によって、地盤沈下を人手などのコストをかけず、月次処理さらにはリアルタイム処理を行うことも可能になり、さらなる利便性が期待される。

#### 謝辞

本研究は新潟県保健環境科学研究所情報調査科との共同研究であり、機材の提供をしていただいている。ここに感謝の意を表します。

#### 参考文献

1. 総務省, 令和元年版情報通信白書, 第 1 部 特集 進化するデジタル経済とその先にある Society 5.0, 第 2 節 デジタル経済を支える ICT の動向, (2) IoT デバイスの急速な普及, <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r01/html/nd112120.html>
2. 新潟県 IT&ITS 推進協議会, にいがた暮らし IoT コンテスト 2016, <http://www.n-it-its.jp/act/2016/07/>
3. 新潟県 IT&ITS 推進協議会, にいがた暮らし IoT コンテスト入選, 猿から畑を守る「サルサーレ!」, <http://www.n-it-its.jp/act/2017/03/post-12.html>
4. 新潟県保健環境科学研究所, <https://www.pref.niigata.lg.jp/site/hokanken/>
5. IoT による地盤沈下監視への適用, 河原和好, 新潟国際情報大学経営情報学部紀要第 1 号, 2018
6. 新潟県保健環境科学研究所の IoT の取組み, 植田信夫、種岡裕, 平成 29 年度全環研北海道・東北支部 IoT の環境計測への適用セミナー, 2018
7. IoT による地盤沈下の観測, 河原和好, 新潟国際情報大学経営情報学部紀要第 2 号, 2019
8. 工学社, 瀧本往人, 『基礎からわかる「IoT」と「M2M」』, 2016
9. 秀和システム, 蔵下まさゆき, 『センサーでなんでもできる おもしろまじめ電子工作』, 2016
10. 新潟県, 県民生活・環境部 環境対策課, 上越地域における地盤沈下, <https://www.pref.niigata.lg.jp/sec/kankyotaisaku/1331240497754.html>
11. ノースワン(株), <http://www.north-one.net/>

12. (株)共和電業, <http://www.kyowa-ei.com/jpn/>
13. Arduino プロジェクト公式サイト, <https://www.arduino.cc/>
14. (株)日さく, <http://www.nissaku.co.jp/>
15. (株)タブレイン, 3GIM モジュール, <http://tabrain.jp/new/product/3GIM.html>
16. (株)タブレイン, 3 GIM シールド, [http://tabrain.jp/new/product/3GIM\\_SHIELD.html](http://tabrain.jp/new/product/3GIM_SHIELD.html)
17. ニフティ(株), NifMO by NIFTY, <https://nifmo.nifty.com/>
18. アンビエントデータ(株), Ambient, <https://ambidata.io/>
19. (株)パナソニック, C-P127R2J(J1) : バックアップ用長寿命シリーズ (LC-P), <https://industrial.panasonic.com/jp/products/batteries/secondary-batteries/vrla/trickle-long-life-type-lc-p/LC-P127R2J%2528J1%2529>
20. Device Plus, 第56回 SORACOM Air×3GIMによるArduinoの3G通信 その4～Arduinoのsleepモードで省エネ実装, [https://deviceplus.jp/hobby/entry\\_056/](https://deviceplus.jp/hobby/entry_056/)
21. ラジオペンチ, Arduino の CPU をスリープさせて消費電流を減らす関数 delayWDT2, <http://radiopench.blog96.fc2.com/blog-entry-830.html>
22. (株)秋月電子通商, DS1307 リアルタイムクロックモジュール (RTC), <http://akizukidenshi.com/catalog/g/gM-13475/>