

令和4年度 特別研究報告書

# センサーネットワークを使用した 水質調査方法の検討

龍谷大学 理工学部 情報メディア学科

T190551 矢埜祥太

三好 力 教授

## 内容梗概

近年、ICT(Information and Communication Technology)や IoT (Internet of Things)が普及してきている。ICT 技術や IoT 技術を利用した製品やサービスは人々の生活に欠かせないものになっている。特に、IoT 技術を利用したサービスでは家電製品、自動車、医療に次いで産業分野での使用が高成長してきている。しかし、地方自治体や行政が行っている河川、湖、水道などの水質調査は人が出向いて水質調査を行う従来の方法で行っている。

本研究では従来の水質調査の方法を自動で効率的に行うことができるように IoT の基盤となっているセンサーネットワークを使用した水質調査の方法を検討することを目的にした。本研究では提案手法として遠距離通信に適しているモバイルアドホックネットワークと自動でデータを取得できるセンサーを使用した手法を提案する。本研究の実験では提案手法の自動でデータを取得する機能を重要視して実験を行った。

# 目次

第1章	はじめに.....	1
1.1	研究の背景と目的.....	1
1.2	本論文の流れ.....	2
第2章	既存技術.....	3
2.1	モノのインターネット(Internet of Things : IoT).....	3
2.2	センサー.....	3
2.3	センサーネットワーク.....	4
2.3.1	センサーネットワークの概要.....	4
2.3.2	センサーネットワークの既存技術.....	5
2.4	本研究で使用する機器.....	5
2.4.1	Raspberry Pi.....	5
2.4.2	ブレッドボード.....	6
2.4.3	Raspberry pi カメラモジュール.....	6
2.4.4	水温センサー(DS18B20).....	6
2.4.5	ジャンパーワイヤー.....	7
2.4.6	LINE Notify.....	7
2.4.7	Google スプレッドシート.....	7
第3章	水質調査.....	8
3.1	水質調査の現状.....	8
3.1.1	水質調査の概要.....	8
3.1.2	水質調査の項目.....	8
3.1.3	水質調査の方法.....	8
3.2	問題点.....	10
3.3	提案手法.....	10
第4章	実験.....	11
4.1	実験目的.....	11
4.2	実験環境.....	11
4.2.1	Raspberry Pi 配線.....	11
4.2.2	実験装置.....	12
4.2.3	プログラムのアルゴリズム.....	13
4.3	実験方法.....	15

4.4	実験結果.....	15
4.5	考察.....	17
<b>第5章</b>	<b>まとめ</b> .....	<b>18</b>
	謝辞	19
	参考文献.....	20
	付録	21

# 第1章 はじめに

## 1.1 研究の背景と目的

近年、情報通信技術(Information and Communication Technology : ICT)やモノのインターネット(Internet of Things : IoT)を利用して、様々な分野の仕事を効率的に取り組もうとしている企業や私たちの生活を便利にするIoTの商品が多くなってきている。総務省によると世界のIoTデバイス数が年々増加していることが分かる(図 1.1.1)[1]。

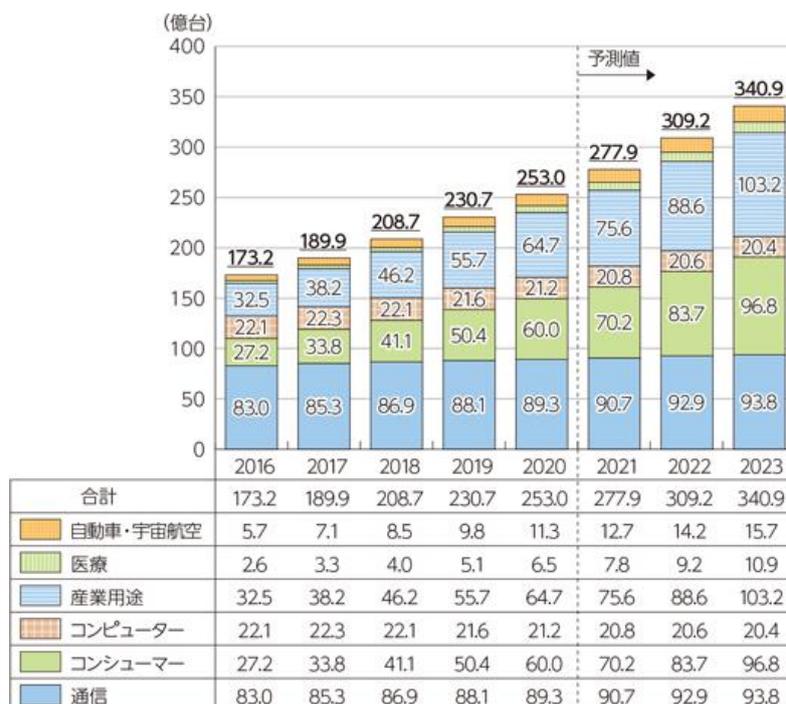


図 1.1.1 世界のIoTデバイス数の推移及び予測

本格的にIoT技術を利用している分野について調べてみると家電製品や自動車、医療だけでなく農業、漁業、環境調査などの産業にも使用されていることが分かった。また、分野・産業別の世界のIoTデバイス数及び成長率予測を調べると次のようなことが分かった。医療や電化製品などのコンシューマーの次に産業用途もIoT化を取り組んで高成長していくと予想されている(図 1.1.2)[1]。

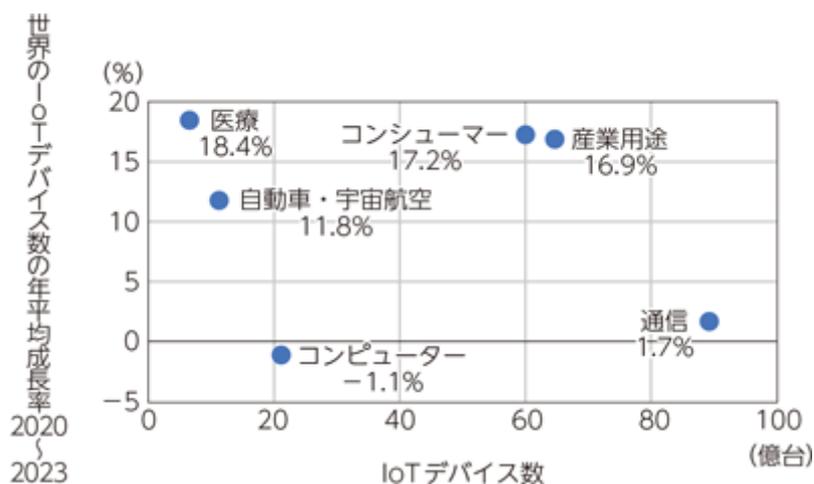


図 1.1.2 分野・産業別の世界の IoT デバイス数及び成長率予測

IoT と産業の関係についても詳しく調べていくと次のようなことが分かった。産業の中でも農業、漁業といった分野はスマート農業やスマート漁業があり、IoT 技術を積極的に利用している。農業や漁業といった産業には水質管理や水質調査という仕事が必要となっている。また、水質調査は農業や漁業だけで行われている仕事ではなく地方自治体や行政が河川や湖、海、水道といった場所でも行っている。しかし、地方自治体や行政が行っている水質調査では IoT 技術をまだ利用せずに従来の方法で行っている。このようなことから、IoT 技術を用いて水質調査を自動で効率的に行えるようなシステムを作成することができたら水質調査の仕事が効率的になっていくと考えた。

本論文ではセンサーネットワークを使用して水質調査を自動で簡単に効率的に行えるようなシステムを作成し水質調査の方法を検討することを目的とする。

## 1.2 本論文の流れ

本論文は以下の章により構成されている。

第 1 章、研究内容の概要について述べている。

第 2 章、既存技術について述べている。

第 3 章、水質調査、問題点、提案手法について述べている。

第 4 章、実験目的、実験環境、実験方法、実験結果、考察について述べている。

第 5 章、本論文に関するまとめ、今後の課題について述べている。



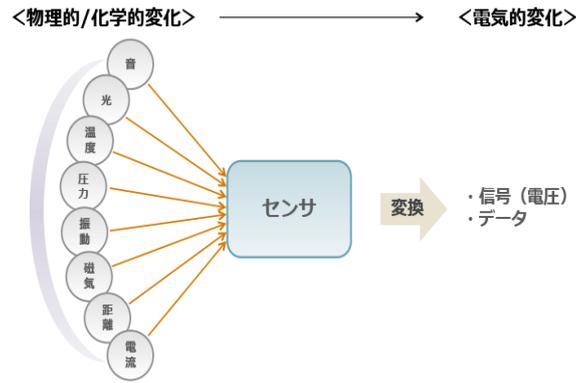


図 2.2.1 センサーの仕組み

## 2.3 センサーネットワーク

2.3.1 にセンサーネットワークの概要、2.3.2 にセンサーネットワークの既存技術について述べる。

### 2.3.1 センサーネットワークの概要

2.1 節で述べたネットワークとはセンサーネットワークのことを表している。IoT 環境を構築するためにはセンサーネットワークという技術が基盤となっている。センサーネットワークとは無線通信機能を持つ多数の小型センサー端末を特定空間内の各所に設置し、それらの機器を協調して働かせることで施設の制御、環境の観測などを可能とする無線ネットワークのことである(図 2.3.1.1)。このように特定空間内に設置した小型センサーがさまざまなデータを検知して収集し情報を数値化する技術をセンシング技術という。センサーネットワークはセンシング技術と無線通信技術を組み合わせて設計されている。小型センサー端末はセンサーノードと呼ばれていてセンサー、無線通信チップ、マイクロプロセッサ、電源ユニットから構成されている。また、生産状況や経過を可視化する際に欠かすことのできない、基礎的なネットワークで、IoT の普及によって需要が急増しこれからの活躍に期待されている。センサーネットワークは無線であるため配線の手間が省けるという長所がある[4]。

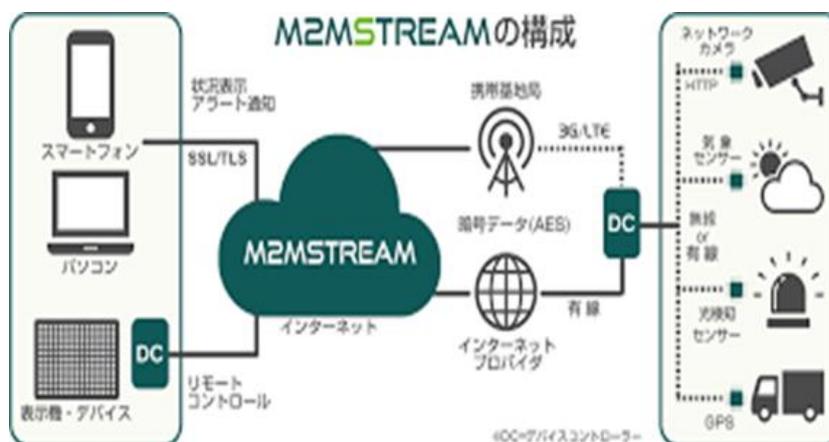


図 2.3.1.1 センサーネットワークの仕組み

### 2.3.2 センサーネットワークの既存技術

センサーノードに搭載されている無線通信チップの通信規格には、微弱無線、Bluetooth、ZigBee などがある。微弱無線、Bluetooth、ZigBee などの通信規格は短距離通信のときに使用される。そのため長距離通信の時には別の技術が使用される。長距離通信に使用される技術としてマルチホップ通信やアドホックネットワークが使用される。マルチホップ通信とはそれぞれの端末同士が直接接続してデータを数珠つなぎに伝送する方式のことだ。このマルチホップ通信を利用しているのがアドホックネットワークだ。アドホックネットワークとはマルチホップ通信のように端末のみで構成され、基地局やアクセスポイントを必要としない自律分散型のネットワークをアドホックネットワークのことである。アドホックネットワークを利用することによってセンサノードは消費電力を抑えながら近距離通信を行い、それを繰り返すことで数 km に及ぶ距離のデータ伝送を実現している。アドホックネットワークにはいくつかの形態がある。その一つの形態にモバイルアドホックネットワーク(MANET)というのがある。MANET とはイーサネットや無線 LAN などのインフラ設備を必要とせず、無線で接続できる端末のみで端末相互の通信を実現する技術である。この通信規格を適材適所で使い分け IoT システムの基盤となっている。図 2.3.2.1 にアドホックネットワークとアドホックではない無線ネットワークの比較を示す[4]。

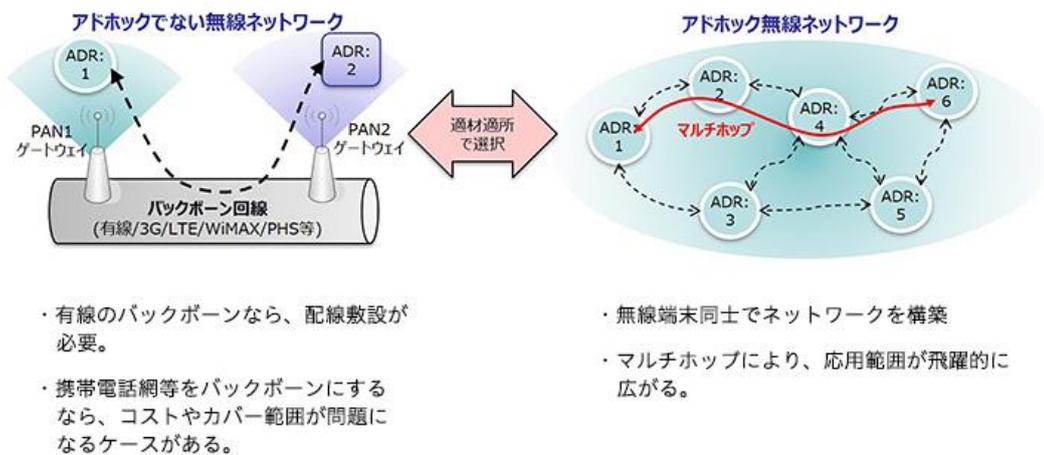


図 2.3.2.1 アドホックネットワークとアドホックではない無線ネットワークとの比較

## 2.4 本研究で使用する機器

本研究で使用する実験機器の概要について述べる。

### 2.4.1 Raspberry Pi

Raspberry Pi はイギリスにある Raspberry Pi Foundation が提供している小型サイズのコンピュータのことである。Raspberry Pi には電源ポートや USB ポート、HDMI 端子が搭載されているためモニターやキーボード、マウスを用意することによって使用できる。また、GPIO(General Purpose Input/Output)を使用することで情報を習得、制御といった電子工作を行うことができる(図 2.4.1.1)。Raspberry Pi で本論文の実験を行った。



図 2.4.1.1 Raspberry Pi

#### 2.4.2 ブレッドボード

ブレッドボードとは、電子部品やジャンパ線をボードの穴に差し込むだけで、はんだ付けをしなくても、手軽に電子回路を組むことができる基盤である(図 2.4.2.1)。本研究では水温センサーを電子工作するときに使用した。



図 2.4.2.1 ブレッドボード

#### 2.4.3 Raspberry pi カメラモジュール

水の濁りや水の色の状態をモニタリングするために小型カメラを使用した(図 2.4.3.1)。

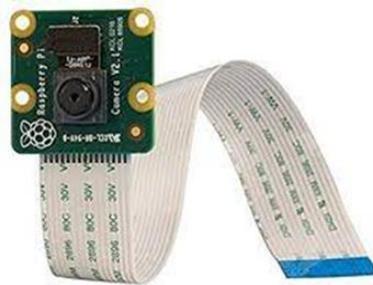


図 2.4.3.1 Raspberry pi カメラモジュール

#### 2.4.4 水温センサー(DS18B20)

1-Wire 通信を利用して水の温度を測定するために水温センサー(DS18B20)を使用した(図 2.4.4.1)。



図 2.4.4.1 水温センサー

#### 2.4.5 ジャンパーワイヤー

Raspberry Pi と水温センサー(DS18B20)をブレッドボードに結線するときに使用した(図 2.4.5.1)。



図 2.4.5.1 ジャンパーワイヤー

#### 2.4.6 LINE Notify

LINE Notify とは LINE と外部の Web サービスやアプリを連携し、ユーザーがカスタマイズされた好みの情報を受け取ることができる機能のことである。今回の研究では LINE Notify を使用して水温のデータを送信した。

#### 2.4.7 Google スプレッドシート

Google スプレッドシートの拡張機能の Apps Script を利用して Raspberry Pi から Google スプレッドシートにデータを送信した。言語は Google Apps Script (GAS) というプログラミング言語を用いた。

## 第3章 水質調査

### 3.1 水質調査の現状

3.11 で水質調査の概要、3.12 で水質調査の項目、3.13 で水質調査の方法について述べる。

#### 3.1.1 水質調査の概要

1.1 の本論文の背景で少し述べた水質調査について詳しく述べる。水質調査とは川の水が量・質ともに正常であるかどうか調べ、知ることにより問題点や解決方法を見つけ出す調査のことをいう。水質調査が必要とされている背景、理由には次のようなことが関係している。また、水質調査が必要とされていなかった時代では現代ほど人間は河川を汚していなかった。河川は自浄能力といって自分の力で水をきれいに保つことができていた。しかし、時代が進むにつれて河川の自浄能力が追い付かなくなり、水質の様々なよごれや濁りの問題が起こってしまった。その結果、水質調査をして水を管理することになった。現代の河川では水質調査だけでなく水質管理も重要とされている。水質調査では河川だけでなく海、湖、水道水などの私たちの生活に必要な場所も対象となって実施されている[5]。

#### 3.1.2 水質調査の項目

水質調査は調査項目というものがある。水質調査項目には一般項目や生活環境項目、健康項目などがある。一般項目とは水質というより水の状態を表す指標で、人間の感覚によって判定するものである。一般項目は水温、外観、臭気、色度、透視度、透明度の 6 項目を対象としている。外観とは水の濁りや水の色の状態のことを指す。生活環境項目とは水質汚濁に係る環境基準のうち、生活環境の保全に関する環境基準で指定されている項目で、最も基本的な水質項目のことである。生活環境項目は水素イオン濃度指数(pH)、溶存酸素量(DO)、浮遊物質質量(SS)などの 10 項目を対象としている。健康項目とは人の健康の保護に関する環境基準で指定されている項目のことである。水質汚濁物質の中でも特に有害性の強いもので、規制値も非常に厳しく定められている。重金属や有機塩素系化合物などの 27 項目を対象としている[6]。

#### 3.1.3 水質調査の方法

現在の河川での水質調査の方法は次のような方法で行われている。橋の上から採水容器を投入するか、直接川に入って、流れの真ん中から採水を行う(図 3.1.3.1、図 3.1.3.2)。現場では溶存酸素濃度と水温、透視度などを測定する(図 3.1.3.3、図 3.1.3.4)。残りの項目はガラスビンやポリビンなどの適切な容器に採取し、実験室で必要に応じて前処理し、機器分析などで定量を行う。測定は分析する人、機関による違いを避けるため分析項目ごとに定められた方法で行っている[7][8]。



図 3.1.3.1 橋上から採取



図 3.1.3.2 直接川に入って採取



図 3.1.3.3 溶存酸素濃度の測定



図 3.1.3.4 透明度の測定

## 3.2 問題点

3.1 節で述べた水質調査の既存技術にはいくつかの問題点がある。日本には河川や湖といった水質調査をする場所は数え切れないほどたくさんある。今までの水質調査の方法では 3.1 節で述べたように調査したい河川や湖に実際に行き水質調査を行っていた。現在の水質調査方法では定期的、臨時的に行わなければならないと定められている。従って、現在の水質調査の方法では膨大な時間と現地に調査に行く人間が必要という問題が起こる。また、現在、現場で測定できる項目は 3.1 節で述べた 4 項目を含めた 6 項目となっている。ここで、現場で測定できる項目が少ないという問題がある。

## 3.3 提案手法

3.2 節の問題点から既存の水質調査の方法では河川、湖に人が出むいて行っていたため膨大な時間と人間が必要で効率的に行うことができていない。そこで、本研究では 3.2 節の問題点を解決するためにモバイルアドホックネットワークとセンサーを用いて自動で水質調査ができる提案手法を提供する。

モバイルアドホックネットワークとセンサーを利用してこれまで河川、湖に人が出向いて行っていた水質調査を自動的に行えるようにすると 3.2 節の問題点は解決することができる。提案手法の手順は調査したい項目のセンサーを用意して河川、湖に設置する。そして、センサーを利用して自動でデータを取得し、モバイルアドホックネットワークでデータを送信する。図 3.1.1 に提案手法のフローチャートを示す。モバイルアドホックネットワークを利用する理由はいつでも、どこでも、だれとでも通信することができ WI-FI、Bluetooth、ZigBee などの短距離通信機器よりも長距離通信をすることができ水質調査では適しているからである。

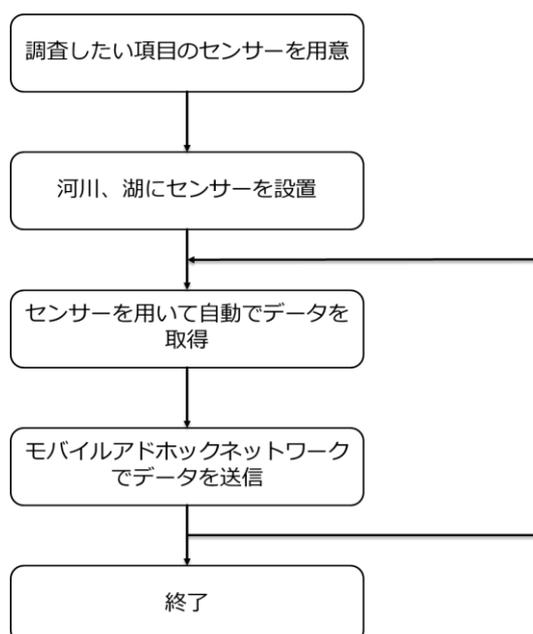


図 3.3.1 提案手法のフローチャート

## 第4章 実験

本章では提案手法で述べた自動でデータを取得する機能の実験を行い考察する。4.1 に実験目的、4.2 に実験環境、4.3 に実験方法、4.4 に実験結果、4.5 に考察を述べる。

### 4.1 実験目的

本研究では Raspberry Pi 3model B を使用し、水温センサー「DS18B20」とカメラセンサー「Raspberry pi カメラモジュール」を用いて作成した水質調査センサーが Wi-Fi 環境の中で自動でデータを取得できるのかを目的として実験を行う。また実験で使用したプログラムを付録に記載する。

### 4.2 実験環境

#### 4.2.1 Raspberry Pi 配線

水温センサー(DS18B20)の GPIO 端子配線を以下に示す。

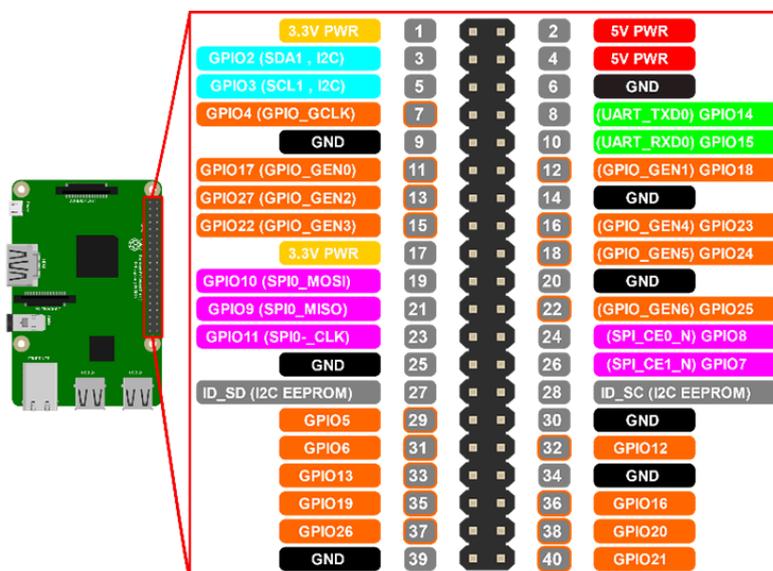


図 4.2.1.1 Raspberry Pi ピン番号

DS18B20	ピン番号
VCC(赤)	1 (3.3V)
GND(黒)	9 (GND)
Data(黄)	7 (1-Wire)

図 4.2.1.2 GPIO 端子とセンサーの配線表

#### 4.2.2 実験装置

本実験では図 4.2.2.1 の回路図のように Raspberry Pi、ブレッドボード、水温センサー (DS18B20)、Raspberry Pi カメラモジュール、抵抗 4.7k $\Omega$ を配線した[9]。そして、透明な器に常温の水を入れ水温センサーとカメラのセンサーを図 4.2.2.2 のように配置して実験を行った。

Raspberry Pi カメラモジュールは Raspberry Pi のカメラポートに接続する。水温センサーは購入時のままではブレッドボードに結線することができなかつたため、水温センサーにジャンパーワイヤーを半田付けしてブレッドボードに結線する。Raspberry Pi の GPIO 端子もブレッドボードに結線するにはジャンパーワイヤーを使用する。Raspberry Pi と水温センサーは図 4.2.1.2 の GPIO 端子とセンサーの配線表のように水温センサーの VCC(赤線)に 1 ピン(3.3V)、GND(黒線)に 9 ピン(GND)、Data(黄線)に 7 ピン(1-Wire)、VCC と Data の間に 4.7k の抵抗を入れ回路を組む。ブレッドボード上では水温センサーの VCC(赤線)が F9、GND(黒線)が F3、Data(黄線)が F5、Raspberry Pi の 1 ピン(3.3V)が J9、9 ピン(GND)が J3、7 ピン(1-Wire)が J5、4.7k の抵抗を H3 から H9 の間の位置に接続をする。

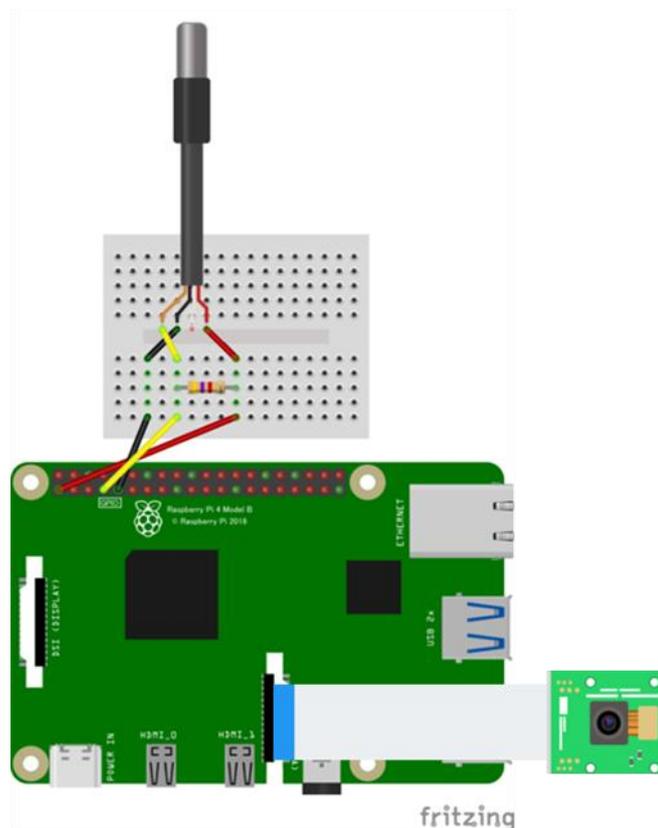


図 4.2.2.1 回路図[8]

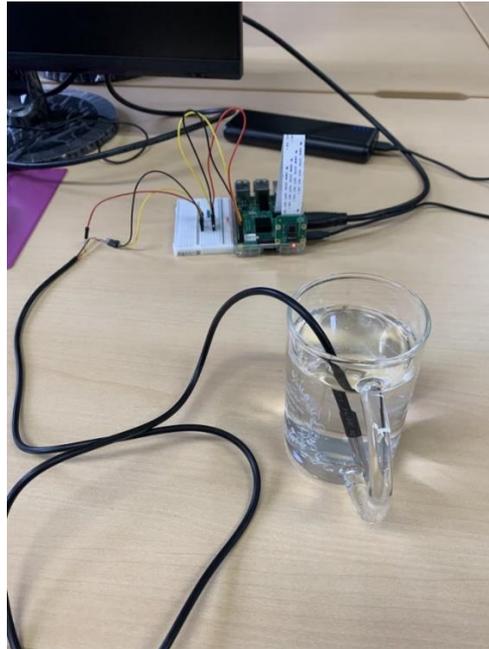


図 4.2.2.2 実験時の配置

#### 4.2.3 プログラムのアルゴリズム

本研究では 2 つのプログラムを作成して実験を行った。今回作成したプログラムは水温センサー(DS18B20)で水温を測定するプログラム、カメラで撮影した写真を LINE に送信するプログラム、ラズパイから Google スプレッドシートへ送信するプログラムを参考に本実験で使用するプログラムを作成した。1 つ目のプログラムは python で作成したセンサーを動かす sensor.py である。sensor.py のアルゴリズムは処理開始、10 分間水温センサーとカメラを同時に起動させ 1 分間隔でデータを取得、撮影した画像をファイルに保存、水温のデータを LINE Notify と Google スプレッドシートに送信する、処理終了という内容のプログラムである。2 つ目のプログラムは GAS で作成したデータを自動で表にする test.gs である。test.gs のアルゴリズムは処理開始、送信されてきたデータを取り出す、現状 Active になっている sheet を取得、Google スプレッドシートの 1 列目に日時、2 行目に送信されたデータ値を追記、処理終了という内容のプログラムである。sensor.py のフローチャートを図 4.2.3.1、test.gs のフローチャートを図 4.2.3.2 に示す[9][10][11]。

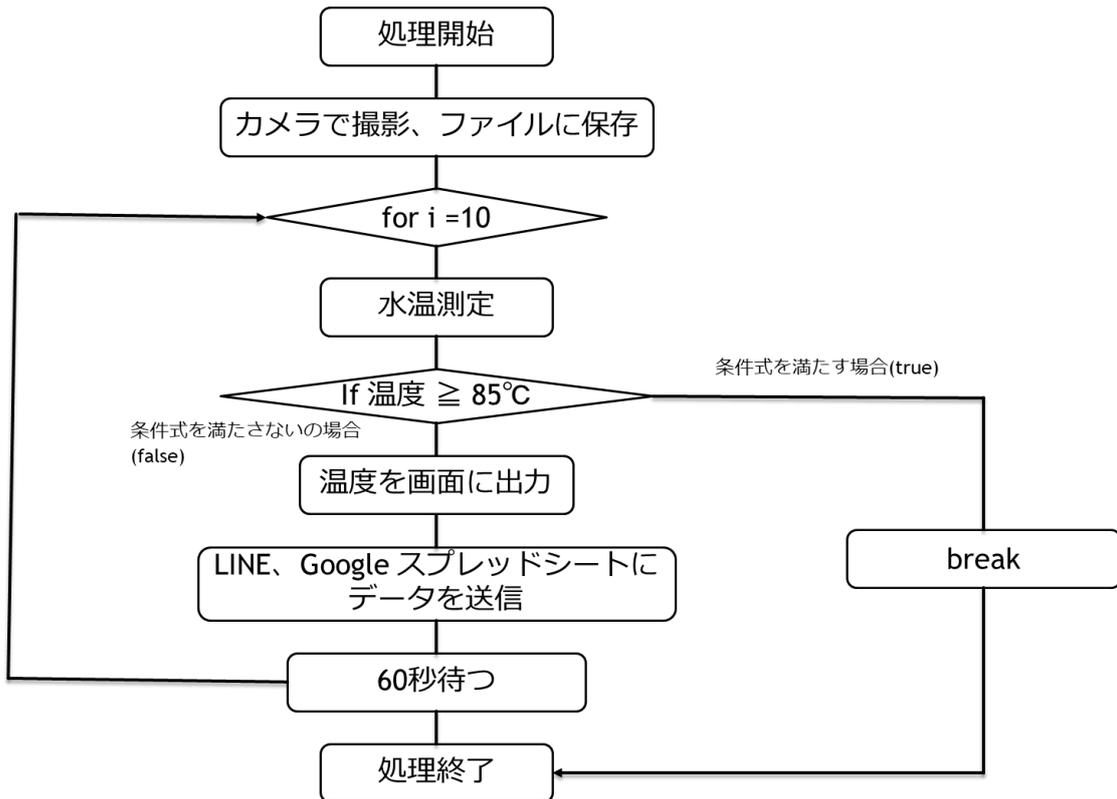


図 4.2.3.1 sensor.py のフローチャート

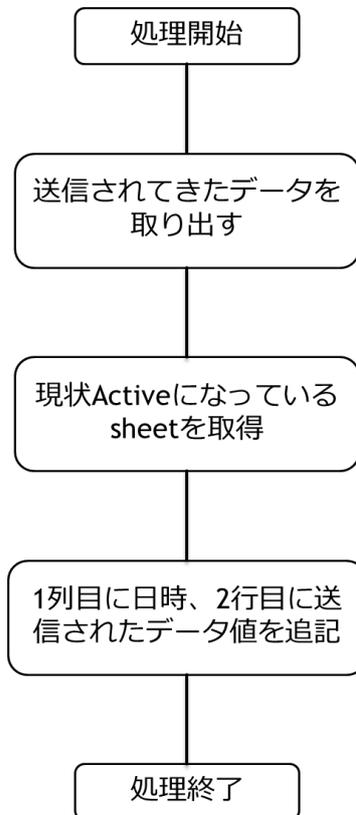


図 4.2.3.2 test.gs のフローチャート

### 4.3 実験方法

本実験の実験場所は Wi-Fi 環境が整っている室内で行う。実験回数は 10 回行いデータを取得する。そして、10 回のデータから典型的なものを実験結果として使用する。今回測定した水は常温の水を使い透明な容器に入れ実験を行う。Raspberry Pi、水温センサー(DS18B20)、Raspberry Pi カメラモジュールを使用して 10 分間水温と写真のデータを取得できるプログラムを動かし、自動で水温の値、撮影した水の外観の写真を取得する。

### 4.4 実験結果

実験結果は次のようになった。水温センサー(DS18B20)で測定した水温は  $21.94 \leq t \leq 22.12$  の範囲となり、測定した値を LINE、Google スプレッドシートに送信することに成功することができた。カメラセンサーを使用して水の外観(水の濁りや水の色の状態)を撮影することに成功した。今回の場合は水の濁りもなく水の色の状態も正常だった。図 4.4.1 に LINE に送信されたメッセージ、図 4.4.2 にラズパイで撮影した水の外観(水の濁りや水の色の状態)、表 4.4.3 に取得したデータ、図 4.4.4 に水温と時間のグラフを示す。

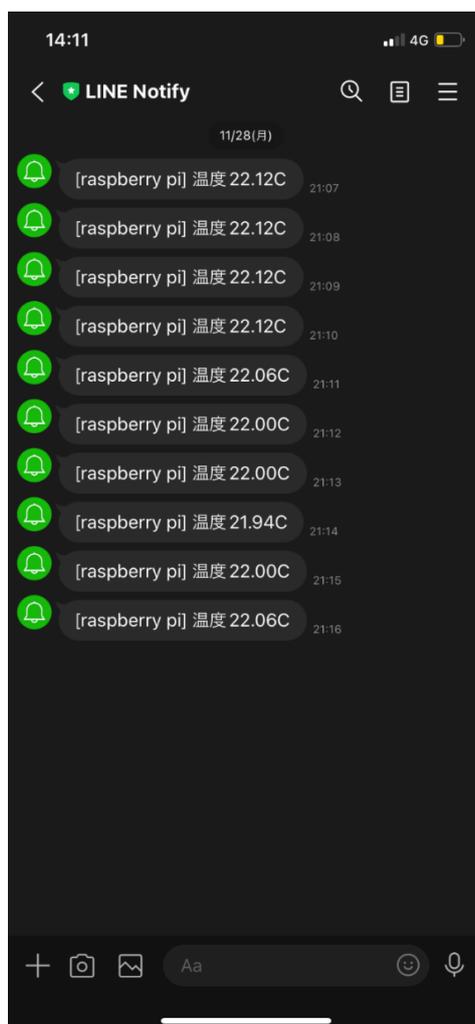


図 4.4.1 LINE に送信されたメッセージ



図 4.4.2 水の外観(水の濁りや水の色の状態)

表 4.4.3 時間と水温

2022/11/28 21:06:59	22.12
2022/11/28 21:08:03	22.12
2022/11/28 21:09:07	22.12
2022/11/28 21:10:11	22.12
2022/11/28 21:11:19	22.06
2022/11/28 21:12:24	22
2022/11/28 21:13:29	22
2022/11/28 21:14:33	21.94
2022/11/28 21:15:38	22
2022/11/28 21:16:44	22.06

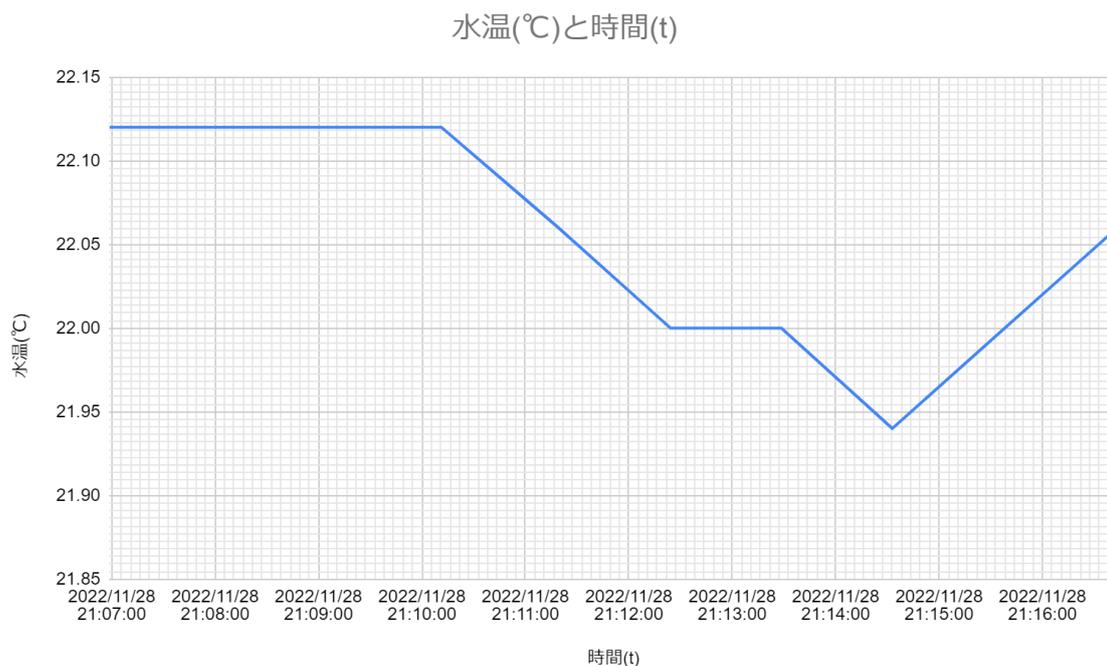


図 4.4.4 水温と時間のグラフ

## 4.5 考察

4.4の実験結果より本実験では実験目的で述べた Wi-Fi と水質調査センサーを用いることによって水温の値と水の外観(水の濁り具合、水の色の状態)の写真を自動で取得できることが分かった。この実験結果を受け、本研究の提案手法で提案している MANET を利用することによって今回の実験と同様に自動で値を取得することが可能となり、本実験よりも長距離通信で実験を行うことができると考えた。しかし、本実験では 2 種類のセンサーでしか実験を行っていないため実用化するためには他の水質調査項目でも自動で測定できるかなどの実験を行う必要があると考えた。

## 第5章 まとめ

本研究では、従来の水質調査の方法の問題点を解消するために水質調査センサーと MANET を利用して自動でデータを取得する手法を提案した。本実験では提案手法で挙げた自動でデータを取得するというのを目的として実験を行った。その結果、Wi-Fi 環境が整っている室内で水温センサーと Raspberry Pi カメラモジュールを使用して水温と水の外観(水の濁りや水の色の状態)のデータを自動で取得することに成功した。

今回は、WI-FI 環境がある室内、2 種類のセンサーを使用するというのを前提にプログラムを作成して実験を行ったため、実際にセンサーの種類を増やして湖、河川に行って実験を行うことはできなかった。今後の研究では、センサーの種類を増やして、MANET や Raspberry Pi に付けられる外付け通信規格を利用して実際に湖、河川に行き通信方式の検証、データを取得できるのかという実験を行っていこうと考えている。今後、センサーネットワークを利用した事業や製品などが増加していくことを期待する。

## 謝辞

本研究を進めるにあたり、龍谷大学工学部情報メディア学科三好力教授には、指導教員として終始熱心なご指導を頂きました。心から感謝いたします。最後に、三好研究室の皆様には、本論文の執筆にあたり多くのご助言、激励を頂きました。本当にありがとうございました。

## 参考文献

- [1]令和3年版 情報通信白書 | IoT デバイスの急速な普及 - 総務省  
<https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r03/html/nd105220.html>
- [2]IoT とは？活用のメリットや懸念点、身近な例をわかりやすく解説！  
<https://www.pasonatech.co.jp/workstyle/column/detail.html?p=7369>
- [3]IoT で活用されるセンサーとは？ 種類や特長を徹底解説  
<https://www.hitachi-solutions-create.co.jp/column/iot/iot-sensor.html>
- [4]センサネットワークとは何か？ 使用されている技術や活用例など  
<https://mvno.freebit.com/column/iot-m2m/sensor-network.html>
- [5]水質調査の目的  
[http://www.skr.mlit.go.jp/yongi/duties/quality/mizupatrol/moktek\\_f.html](http://www.skr.mlit.go.jp/yongi/duties/quality/mizupatrol/moktek_f.html)
- [6] 水質調査  
<https://www.hrr.mlit.go.jp/hokugi/file/mijika/glossary1.pdf>
- [7] 「水質調査」では何を調べているの？ - 埼玉県  
<https://www.pref.saitama.lg.jp/cess/cess-kokosiri/cess-koko8.html>
- [8] 水質調査方法 | 法令・告示・通達 - 環境省  
<https://www.env.go.jp/hourei/05/000140.html>
- [9] ラズパイで水温センサー (DS18B20) から python3 で水温を取得する。  
<https://growplants.jp/ds18b20/>
- [10] ラズパイのカメラ写真を python で LINE に送信  
<https://blog.memo-labo.com/?p=1739>
- [11]ラズパイから Google スプレッドシートへ送信  
[https://monomonotech.jp/kurage/raspberrypi/google\\_spreadsheet.html](https://monomonotech.jp/kurage/raspberrypi/google_spreadsheet.html)
-

## 付録

[9][10][11]

```
sensor.py

import picamera

import time

import datetime

import requests

from w1thermsensor import W1ThermSensor, Unit

sensor = W1ThermSensor()

ERR_VAL = 85000

with picamera.PiCamera() as camera:

    #カメラ設定

    camera.resolution = (640, 480)

    camera.rotation = 180

    camera.start_preview()

    time.sleep(5)

    dirname = '/home/pi/Pictures/'

    #撮影

    nowTime = datetime.datetime.now()

    fileName = nowTime.strftime('%Y%m%d-%H%M')+'.jpg'

    camera.capture(dirname + fileName)

    #30 分間測定

    for i in range(10):

        #温度測定

        temperature_in_celsius = sensor.get_temperature()

        print("celsius:    {:.2f}".format(temperature_in_celsius))

        #エラー処理

        if temperature_in_celsius == ERR_VAL:

            print("ERROR")

            break

        #google spreadsheets に温度を送信

        gurl = "https://script.google.com/macros/s/AKfycbwRZpjauFLlcWK1UcBpFf8OPvuCQDfnSUcz0VkwP4rLH4yt24i71nNQPE7PliJPqCXAp

g/exec?

        data1 =[str(format(temperature_in_celsius, '.2f'))]

        payload = {"data1" : data1}

        requests.get(gurl,params=payload)

        #LINE に送信
```

```

lurl = "https://notify-api.line.me/api/notify"
token = "LINE トークン"
headers = {"Authorization" : "Bearer " + token}
message = ["温度" + str(format(temperature_in_celsius, '.2f')) + "C"]
payload = {"message" : message}
r = requests.post(lurl, headers = headers, params=payload)
time.sleep(60)

test.gs

function doGet(e) {
  //データ data1 を取り出す
  var value = e.parameter.data1;
  //現状 Active になっている sheet を取得
  var sheet = SpreadsheetApp.getActiveSheet();
  //1 列目に日時、2 行目に送信されたデータ値 を追記
  sheet.appendRow([new Date(), value]);
}

```