

# *INADUMA RULER3*

香川高等専門学校 宇宙開発研究部  
気象観測機器コンテスト2次審査報告書

開発者

機械工学科 3年 大西克也  
電気情報工学科 3年 和気佑弥

担当教員

村上幸一

# 目次

要旨.....	1
1 制作背景と目的.....	2
2 ミッション内容.....	2
3 使用機器.....	3
3.1 AS3935 使用雷センサーモジュール.....	3
3.2 Raspberry Pi.....	3
3.3 Raspberry Pi Zero.....	4
4 稲妻ルーラー3 の説明.....	4
4.1 稲妻ルーラー3 で使用する観測機器（稲妻ルーラー）の説明.....	5
4.2 稲妻ルーラー3 を使用した落雷の観測方法.....	5
4.3 雷センサーのシステム.....	7
4.3.1 雷センサーの動作システム.....	7
5 観測機器を制御するプログラム.....	8
5.1 センサーの設定を行うプログラム.....	9
5.1.1 アンテナのチューニングを行うプログラム.....	9
5.2 観測を行うプログラム.....	10
6 落雷の地点を特定する方法.....	11
6.1 計算方法.....	12
6.2 測位システム.....	14
6.2.1 データ入力の流れ.....	14
6.2.2 測位計算の流れ.....	15
6.2.3 データ出力の流れ.....	15
6.3 設置地点の座標とプログラムの検証.....	15
6.3.1 設置地点の座標について.....	15
6.3.2 検証.....	16
7 落雷の観測と結果.....	17
7.1 室内からの落雷の観測.....	17
7.2 Raspberry Pi Zero を使用した観測.....	18
7.3 落雷の発生地点の観測.....	18
9 考察.....	19
10 感想.....	20
参考文献.....	20

## 要旨

本機器の前身である「稲妻ルーラー」は Raspberry Pi を用いて雷センサーモジュールを制御し落雷地点との距離を観測する機器として開発してきた。今回製作した「INADUMA RULER3（稲妻ルーラー3）」は、「稲妻ルーラー」を3台用いて3地点から落雷を観測することで得られるデータを利用し、落雷地点を特定する機器である。

また、特徴として機器が小さく、安価に製作することができる。また設置と稼働が簡単で誰でも扱えるような機器である。

# 1 制作背景と目的

これまでの稲妻ルーラーはオフラインで持ち運びができ、どこでも雷を探知できることを目的としていたが、新しく制作する「稲妻ルーラー3」は安価で扱いやすく広く正確な観測ができることを目的とした。

今回は稲妻ルーラー3を用いて瀬戸内海で発生する落雷について観測を行う。これによりフェリーや瀬戸大橋を利用する人や、瀬戸内海の島々を訪れる観光客を雷の脅威から遠ざけることができる。また稲妻ルーラー3は機械について詳しくない人たちでも簡単に設置し利用できることを目的としており、これにより様々な観光地などで活用することができると思われる。

# 2 ミッション内容

## ① 香川県と岡山県にそれぞれ Raspberry Pi で制御された雷センサーを設置する。

これまで1か所にしか設置しなかったセンサーを3地点に設置する。設置場所は香川県で高松市と三豊市、岡山県では玉野市に設置する。これにより瀬戸内海付近の落雷を観測することができるようになる。センサーの制御とデータの保存はこれまでと同じように Raspberry Pi を用いて行う。

## ② 設置したセンサーで発生した雷との距離、発生時間を観測する。これを1つの雷に対して3つのセンサーで同時に行う。

落雷との距離と発生時間の観測及び記録は雷センサーが自動で行う。センサーは Raspberry Pi が起動すると同時に観測を開始する。観測では特定の落雷に対して3つの距離データが得られるようにする。

## ③ 取得した3つのデータをもとに発生した雷の位置を特定する。

3つのラズベリーパイから回収した落雷データは、一度、落雷時刻が一致する同じ雷のものと思われるデータごとにまとめる。このデータから、GPSの測位原理を用いた測位プログラムを利用することで、落雷地点の具体的な特定を行う。

### 3 使用機器

- ・ AS3935 使用雷センサーモジュール
- ・ Raspberry Pi
- ・ Raspberry Pi Zero (試験運用段階で使用)

#### 3.1 AS3935 使用雷センサーモジュール



図1 AS3935 使用雷センサーモジュール

この雷センサーに搭載されているアンテナは雷が発生した時に起きる電磁波を受信して、そのパターンを解析し、自然現象である雷と人工的なノイズを判別する。受信した電磁波が雷と判別されるとそのエネルギー量を計算して距離を推測する。測定は設置地点から半径 40Km までの範囲で可能である。

#### 3.2 Raspberry Pi



図2 Raspberry Pi

「Raspberry Pi」はラズベリーパイ財団が開発したシングルボードコンピューターである。Raspberry Piにはさまざまな種類が存在するが今回は「Raspberry Pi 2 Model B」と「Raspberry Pi 3 Model B」を使用した。Raspberry PiはCPUにARMプロセッサが用いられている。またハードディスクなどの記憶装置は搭載されていないためSDカードを取り付けて使用する必要がある。今回はRaspberry Piをセンサーの制御とデータの保存、解析のために用いる。

### 3.3 Raspberry Pi Zero

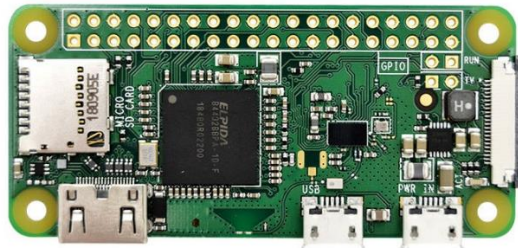


図3 Raspberry Pi Zero

「Raspberry Pi Zero」は上記で説明したRaspberry Piを小型化し、省電力で取り扱えるコンピューターである。今回はWi-Fi/BLE機能が付いた「Raspberry Pi Zero W」を使用した。Raspberry Pi Zeroを利用しようとした目的は稲妻ルーラーをより低価格で扱いやすくするためである。試作機段階では利用をしたが後述する理由によって本番機には利用しなかった。

## 4 稲妻ルーラー3の説明

稲妻ルーラー3は一昨年に制作した初期の稲妻ルーラー（発生した落雷との距離を観測できる機器）の数を増やし、特定の雷に対して得られる複数の距離データを用いて落雷の発生地点を観測する一連のシステムである。

#### 4.1 稲妻ルーラー3 で使用する観測機器（稲妻ルーラー）の説明

稲妻ルーラー3 の開発において制作する観測機器（稲妻ルーラー）は上記で説明した「AS3935 使用雷センサーモジュール」を「Raspberry Pi」の GPIO に接続して制御し、落雷の発生時間とその距離を測定する装置である。

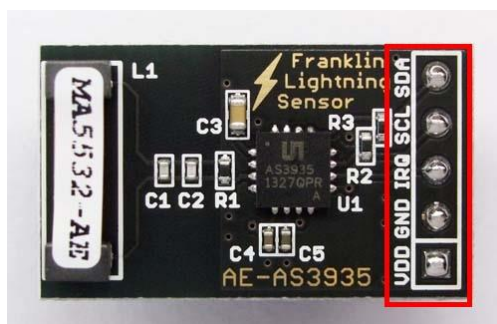


図 4 雷センサーの接続部

雷センサーは図 4 の赤枠部分にピンをはんだ付けすることで Raspberry Pi の GPIO に接続できるようになる。雷センサーの「VDD」は Raspberry Pi の「3.3v (pin 1)」に、「GND」は「Ground」、「IRQ」は「GPIO 4 (pin 7)」、「SCL」は「SCL (pin 5)」、「SDA」は「SDA (pin 3)」にそれぞれ接続する。

センサーの制御については「Python」で作成したプログラムで行う。センサーのレジスタについての設定などは Phil Fenstermacher 氏が開発したプログラムをダウンロードし、バグを修正して使用する。この他にアンテナのチューニングを行うプログラムと雷センサーを制御して観測を行い、データを保存するプログラムを作る必要がある。これらのプログラムの詳細については後述する。

#### 4.2 稲妻ルーラー3 を使用した落雷の観測方法

稲妻ルーラー3 は過去に制作してきた機器とは違い、落雷の距離データを取得できるとそれを一度収集して、データを基に落雷地点の算出を別のプログラムを用いて行う必要がある。下記の表は機器の設置から落雷の発生地点を調べるまでの主な流れである。

##### 落雷を観測する手順

(雷の発生前)

- ・観測機器を 40km 程離れた場所にそれぞれ設置
- ・機器が周囲から発生する電磁波を常に受信し、判別を行う

(雷の発生)

- ・センサーが落雷によって発生した電磁波を受信

- ・受信した電磁波が雷か人工的なノイズかを判別
- ・雷と判別した場合、独自のアルゴリズムでその距離を計算
- ・距離と発生時間をデータとして出力、Raspberry Pi が保存  
(雷の発生後)
- ・各地点に設置された機器を回収
- ・保存されたデータを取り出し、解析
- ・専用のプログラムを使い 3 個の距離データから雷の発生地点を算出

上記に記載された雷の発生から発生後までの手順については既に過去の稲妻ルーラーで行ってきたことである。発生後の手順、特に「専用のプログラムを使い 3 個の距離データから雷の発生地点を算出」については今回が初めての試みであり、稲妻ルーラー3の開発において最も重要な作業である。機器を設置する 3 地点はミッション内容でも記載したが、香川県で高松市と三豊市、岡山県では玉野市に設置する。

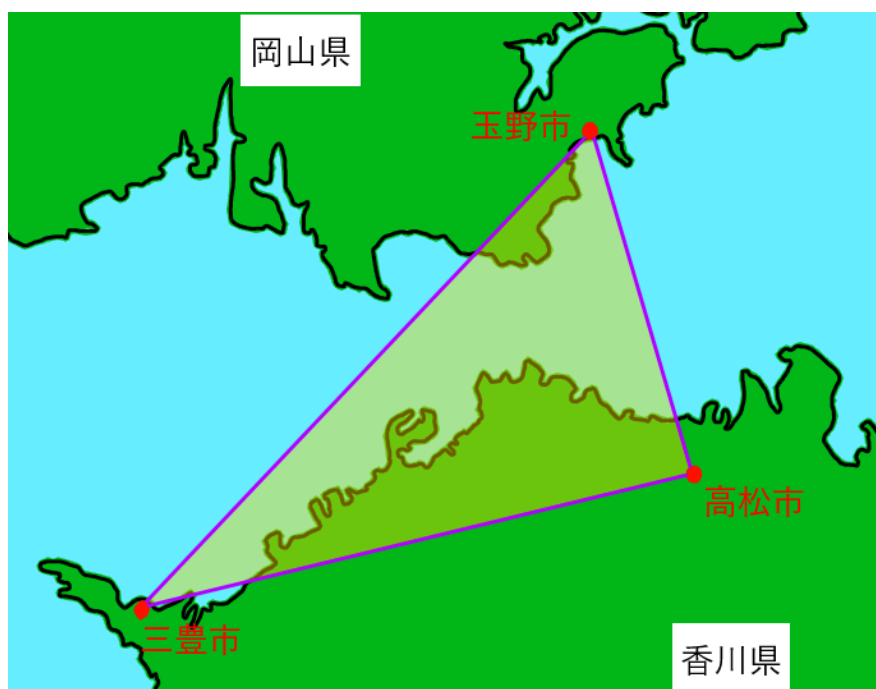


図 5 機器の設置場所

図 5 は各観測機器の設置場所を示している。赤点が設置地点を示し、紫線で結ばれた黄色の三角形内で落雷の測定を行う。高松市の設置地点から三豊市の設置地点までの距離が 40km 程で、玉野市の設置地点までは 40km より短い。また、三豊市の設置地点から玉野市の設置地点までの距離は 40km より長い。図からも読み取れるように今回の測定では香川県の北部と瀬戸内海の一部を観測することができると考えられる。



それぞれの観測機器で雷のデータが取得できた場合それらを回収しに行く必要がある。観測機器はインターネットに接続されていないためデータの収集などは全て手動で行う。今後の発展として、この作業はインターネットを介して行えるようになればより実用性が向上すると思われる。

### 4.3 雷センサーのシステム

使用機器の説明でも紹介したが、稲妻ルーラー3 において雷の測定は「AS3935 使用雷センサーモジュール」を用いて行う。このセンサーは電磁波を受信すると独自のアルゴリズムでそれを自然発生した雷か人工的なノイズかを自動で判別し、距離を算出することができる。

#### 4.3.1 雷センサーの動作システム

「AS3935 使用雷センサーモジュール」には「パワーダウン・モード」、「聴取（リスニング）モード」、「信号検証モード」の3つの動作モードがある。センサーはこれらのモードを状況に応じて切り替えながら長期間の観測を行う。

パワーダウン・モード	雷の観測を行っていない状態。この状態ではセンサーの消費電流を最低限の 1 $\mu$ A まで下げるため、すべての回路の電源が切られる。
聴取（リスニング）モード	雷の観測を行っている状態。この状態に切り替わるとセンサーは、AFE、監視（ウォッチドッグ）、ノイズ下限水準を生成し、観測を行う。この状態でのセンサーの腫脚電流は 60 $\mu$ A 程度である。
信号検証モード	受信した信号を解析する状態。センサーは聴取（リスニング）モードで設定された値を超える信号を受信するとこのモードに切り替わる。信号検証モードで解析した信号が人工的なノイズであった場合センサーは自動的に聴取（リスニング）モードに切り替わる。

聴取（リスニング）モードでは常に周囲から発生する電磁波を取得する。そしてその電磁波が監視（ウォッチドッグ）の閾値を超えると信号検証モードに切り替わる。この閾値は必要に応じて変更をすることができ、閾値を高くすることで雑音などに強くなるが、遠くで発生した雷などの観測が難しくなる。

次にこの雷センサーがどのような回路で動作をしているかを説明する。

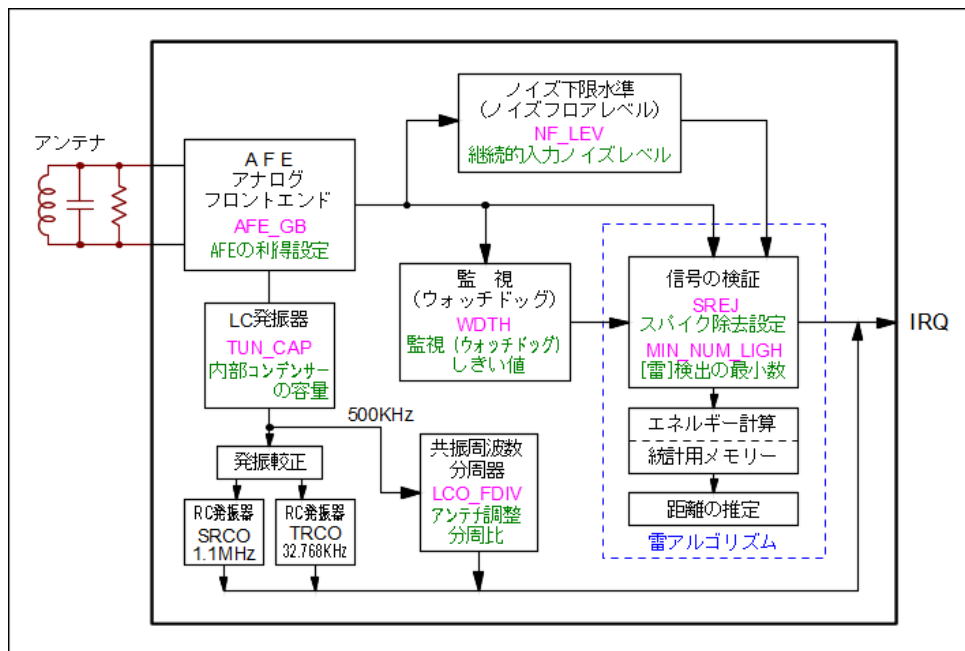


図 6 回路のブロック図

雷センサーは外部のアンテナで信号を受信する。アンテナは内部の AFE（アナログ・フロントエンド）に直接接続されており、受信した信号を増幅し復調している。AFE によって増幅された信号は監視（ウォッチドッグ）に出力されて信号の検証が行われる。またこの信号は同時にノイズ下限水準を生成するためにも利用される。ノイズ下限水準は常に監視（ウォッチドッグ）の閾値と比較されている。AFE で増幅された信号で生成されたノイズ下限水準が監視（ウォッチドッグ）の閾値を超えた場合、センサーは自動的に外部装置へ正常に動作していないことを知らせる。

AFE によって増幅された信号が監視（ウォッチドッグ）の閾値を超えると信号の解析が始まる。解析した結果が雷であった場合センサーは独自のアルゴリズムでその雷のエネルギーを計算し、それを基に距離の推定を始める。

## 5 観測機器を制御するプログラム

稲妻ルーラー3 のプログラムは大きく分けると 2 つある。1 つは Python で作成した観測機器の制御に用いられるプログラムで、2 つ目は C 言語を使用した 3 つの距離データを入力することで特定の雷の発生地点を算出するプログラムである。ここでは 1 つ目の観測機器を制御するプログラムについて説明する。

観測機器を制御するプログラムは「センサーの設定」と「観測を行うプログラム」の 2 つ

の構成で作られている。

## 5.1 センサーの設定を行うプログラム

センサーの設定を行うより先に、まず Raspberry Pi が I2C インターフェースを使えるように、I2C のパッケージとユーティリティをインストールする必要がある。また、割り込み信号 IRQ を読み取るために GPIO も使うため、GPIO ライブラリもインストールする。

I2C インターフェースと GPIO が使用できるようになると、次は雷センサーの設定を行う。雷センサーの設定を行うプログラムは、センサーのレジスタについて設定するプログラムとアンテナのチューニングを行うプログラムがある。センサーのレジスタについての設定は「4.1 稲妻ルーラー3 で使用する観測機器（稲妻ルーラー）の説明」でも記載したように Phil Fenstermacher 氏が開発したプログラムをダウンロードし、バグを修正して使用することで完了する。

### 5.1.1 アンテナのチューニングを行うプログラム

雷センサーには周囲から発生する電磁波を受信するアンテナが搭載されている。このアンテナは使用する前にチューニングを行う必要があり、これを行わないと正確な観測ができなくなる。

このアンテナは並列 LC 共振器に基づく、ループアンテナを使用している。アンテナは 500 [kHz] の周波数で共振し、約 15 の Q 値を持つように設定しなければならない。アンテナのチューニングには一定の周期で `tun_cap` 値切り替わるプログラムを作成し、オシロスコープで各 `tun_cap` 値の IRQ の周波数を測定する。また IRQ ピンには 1/16 周期で矩形波が出力されるため、オシロスコープに表示された値を 16 倍してその値がアンテナの共振する周波数である 500 [kHz] に最も近くなる `tun_cap` 値を使用する必要がある。

アンテナのチューニングを行うためのプログラムは下の表のような構成で作られる。このプログラムを実行している最中は、オシロスコープのプラス端子を IRQ に、マイナス端子を GND に接続する。

- |  |
|--|
| <ul style="list-style-type: none"><li>• 雷センサーの I2C アドレスと Raspberry Pi の種類を指定</li><li>• アンテナの共振周波数を IRQ ピンに出力するように指定</li><li>• 10 秒ごとに <code>tun_cap</code> 値を切り替えてその値を表示（<code>tun_cap</code> 値は 4~15 まで表示される）</li><li>• <code>tun_cap</code> 値を 15 まで表示させたらプログラムを終了しレジスタを元に戻す</li></ul> |
|--|

表 1 アンテナのチューニング結果

tun_cap 値	測定値 [kHz]	16 倍した値[kHz]
4	31.38	502.08
5	31.17	498.72
6	31.08	497.28
7	31.00	496.00
8	30.84	493.44
9	30.75	492.00
10	30.67	490.72
11	30.59	489.44
12	30.46	487.36
13	30.38	486.08
14	30.29	484.64
15	30.17	482.72

表 1 は実際にアンテナのチューニングを行った際の結果である。この雷センサーは tun\_cap 値が 5 の時にオシロスコープでの測定値が 31.17 [kHz]である。この値を 16 倍すると 498.72 [kHz]となり、4 から 15 までの tun\_cap 値の中で最も 500 [kHz]に近い値になる。よってこのセンサーの tun\_cap 値は 5 に設定する。

## 5.2 観測を行うプログラム

観測を行うプログラムは起動されると同時に実行されるように設定する。また、観測されたデータは常時テキストファイルに書き込まれ、設置する直前に Raspberry Pi に時刻指定することで発生時間の記録もできる。

- ・雷センサーの I2C アドレスと Raspberry Pi の種類を指定
- ・屋内と屋外のどちらで観測するかを指定
- ・アンテナのチューニング結果（使用する tun\_cap 値）を指定
- ・各ノイズのレベルで表示する結果を設定  
(ここから 1 秒ごとに無限ループ)
- ・GPIO をセットしてセンサーによる観測を開始
- ・信号を受信するとそのノイズのレベルを確認
- ・信号が落雷と判断された場合、その距離を推定
- ・観測できた落雷の発生時間と距離をテキストファイルに出力

このプログラムは信号を受信するとそれを解析し、その種類に応じて結果をテキストファイルに出力する。まず、大きい電磁波を受信したときテキストファイルに「Noise level too high」と表示する。そのデータを解析し、電子機器などから発生する人工的なノイズと判断されるとテキストファイルに「Disturber detected」と表示し、マスクビットを立てる。一方で解析結果が落雷と判断されると「lightning!」と表示され距離の推定を開始する。表示結果に人工的なノイズの観測が多い場合、センサーの閾値を上げて対応することができる。またセンサーが観測を開始するとテキストファイルに「Waiting for lightning」と出力される。これにより観測の開始時間も知ることができる。

実際に記録されていたテキストファイルの内容

```
2021/08/26 12:37:37 , Waiting for lightning , 63 , 0    (ここから測定を開始)
2021/08/26 12:37:37 , Noise level too high , 63 , 0    (開始と同時に人工的ノイズを感知)
2021/08/26 12:37:37 , Disturber detected , 63 , 0    (人工的なノイズと判断される)
2021/08/26 12:38:12 , Noise level too high , 63 , 0
2021/08/26 12:42:00 , Waiting for lightning , 63 , 289968
2021/08/26 12:42:00 , Disturber detected , 63 , 289968
```

これは 2021 年の 8 月 26 日にセンサーが記録した内容の一部である。「lightning!」の表示はないが「Disturber detected」が表示されていることから人工的なノイズと落雷との区別ができていたことが分かる。

もしこの時間帯(2021/08/26 12:42:00)で距離 27 km の地点に落雷が発生しセンサーが正しくそれを観測できた場合テキストファイルに、  
「2021/08/26 12:42:00 , lightning! , 27 , X」  
と表示される。27 はセンサーが推定した距離であり、X には信号のエネルギーの大きさが具体的な数値で書き込まれる。

## 6 落雷の地点を特定する方法

落雷位置は、3 地点から取得した 3 つのデータを組み合わせて算出する。落雷位置の特定のための計算方法は GPS の測位原理を参考に導出した。

## 6.1 計算方法

図7に示すように、落雷発生地点の座標を  $P(x_p, y_p, z_p)$  とする。また、観測機器の設置地点の座標をそれぞれ  $R_1(x_1, y_1, z_1)$ 、 $R_2(x_2, y_2, z_2)$ 、 $R_3(x_3, y_3, z_3)$  とする。この時、 $R_i$  地点( $i=1,2,3$ )から落雷地点までの距離は次の式(1)で表すことができる。

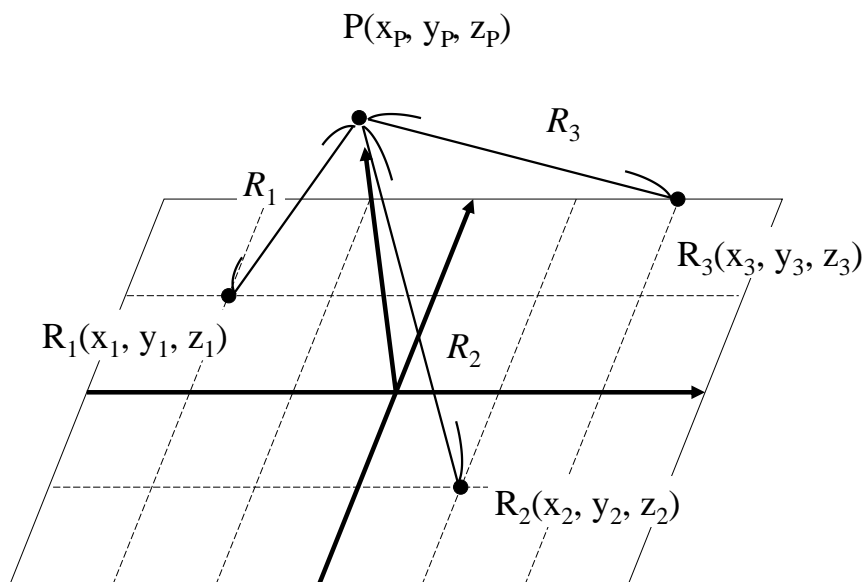


図7 観測地点と発雷地点の模式図

$$R_i = \sqrt{(x_i - x_p)^2 + (y_i - y_p)^2 + (z_i - z_p)^2} \quad (1)$$

式(1)から、1つの落雷に対し観測地点ごとに3つの式を立て連立方程式を解くことで、落雷地点を特定することができる。しかし、このような連立方程式を解くのは容易ではないため、ニュートン法を用い近似値を求める。

ニュートン法を用いるため、落雷地点  $P$  の各要素を次のように初期値( $0$ )と修正値( $\Delta$ )で表す。

$$x_p = x_0 + \Delta x$$

$$y_p = y_0 + \Delta y$$

$$z_p = z_0 + \Delta z$$

また、式(1)の全微分は式(2)となる。

$$\Delta R_i = \frac{\partial R}{\partial x} \Delta x + \frac{\partial R}{\partial y} \Delta y + \frac{\partial R}{\partial z} \Delta z \quad (2)$$

ここで、 $\Delta R_i$ は落雷地点との距離  $R_i$  と、初期値から得られる落雷地点との距離との差分であるから、次の式(3)となる。

$$\Delta R_i = R_i - \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} \quad (3)$$

また、式(2)中の偏微分項はそれぞれ

$$\begin{aligned}\frac{\partial R}{\partial x} &= \frac{-(x_i - x_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}} \\ \frac{\partial R}{\partial y} &= \frac{-(y_i - y_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}} \\ \frac{\partial R}{\partial z} &= \frac{-(z_i - z_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}}\end{aligned}$$

となり、上からそれぞれ $\alpha_i$ ,  $\beta_i$ ,  $\gamma_i$ とおくと、式(2)は次の式(4)のように行列の形に表すことができる。

$$\begin{pmatrix} \Delta R_1 \\ \Delta R_2 \\ \Delta R_3 \end{pmatrix} = \begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ \alpha_2 & \beta_2 & \gamma_2 \\ \alpha_3 & \beta_3 & \gamma_3 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} \quad (4)$$

さらに、式(4)の各行列について、

$$A = \begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ \alpha_2 & \beta_2 & \gamma_2 \\ \alpha_3 & \beta_3 & \gamma_3 \end{pmatrix}, \quad \Delta X = \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix}, \quad \Delta R = \begin{pmatrix} \Delta R_1 \\ \Delta R_2 \\ \Delta R_3 \end{pmatrix}$$

とおくと、式(4)は次の式(5)と表すことができる。

$$\Delta R = A \Delta X \quad (5)$$

これより修正値 $\Delta X$ は次の式(6)から求めることができる。

$$\Delta X = A^{-1} \Delta R \quad (6)$$

ニュートン法では、修正値 $\Delta X$ を求めた後、その各要素が十分小さくなければ $(x_0, y_0, z_0)$ の各要素に修正値 $(\Delta x, \Delta y, \Delta z)$ を加えたものを新たな初期値として再度修正値を求め、十分小さければ現在の初期値 $(x_0, y_0, z_0)$ を解とする。つまり、観測機器の設置地点  $R_i$ 、発雷地点とそれぞれの観測地点からの距離  $R_i$  が分かるとき、発雷地点の適当な初期値を用意すれば、次のような手順によって発雷地点を求めることができる。

- (1) 式(3)により $\Delta R_i$ を求める
- (2)  $\alpha_i$ ,  $\beta_i$ ,  $\gamma_i$ を計算し、行列 $A$ を求める
- (3) 行列 $A$ の逆行列 $A^{-1}$ を求める
- (4) 式(6)より修正値 $\Delta X$ を求める
- (5)  $\Delta X$ の要素が十分小さくなければ初期値に修正値を加算し(1)に戻り、十分小さければ初期値を解として計算を終了する。

## 6.2 測位システム

6.1 節をもとに、落雷地点を特定するプログラムを作成した。データの入力、測位計算、データの出力は一つの C 言語プログラムによって実行し、ファイルからデータの入出力を行うこととした。図 8 にシステムのファイル構造を示す。

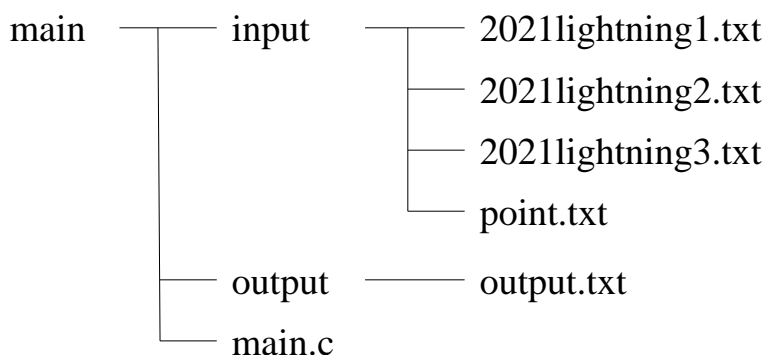


図 8 測位システムのファイル構造

### 6.2.1 データ入力の流れ

データ入力には各地点の観測データが書き込まれた 3 つのテキストファイルと初期値、各観測地点のそれぞれの座標データが書き込まれたテキストファイルの、合わせて 4 つのファイルを使用する。

観測データには雷との距離データ (“lightning”) のほかに、ノイズが高いことを示す (“Noise level too high”) や観測状態に入ったことを示す (“Waiting for lightning”) などのログも含まれている (図 9)。そのため、テキストファイルを 1 行ごとに構造体ヘッダを読み込み、その後に距離データを判別する。構造体には時刻データ、メッセージ、距離データ、信号データを入力するが、“lightning”を判別した後は、時刻データと、距離データ以外のデータは使用しない。また、落雷地点の初期値と観測地点の座標データは、観測データとは別のテキストファイルから読み込み計算に使用する。

```
2021/09/09 08:17:37,Disturber detected,63,0
2021/09/10 18:10:21,Waiting for lightning,63,0
2021/09/10 18:10:22,Noise level too high,63,0
2021/09/19 19:13:06,lightning!,12,46145
```

図 9 観測データの例



## 6.2.2 測位計算の流れ

6.2.1 で読み込んだデータを用い落雷地点の座標を計算する。6.1 節に示した計算手順をもとに、アルゴリズムを作成した。

入力された値から落雷地点の初期値と観測地点の座標のデータをもとに $\alpha_i, \beta_i, \gamma_i$ を求め、行列  $A$  となる二次元配列に格納する。また、各地点の観測データ、落雷地点の初期値、観測地点の座標のデータをもとに式(3)から $\Delta R_i$ を求め一次元配列に格納（式(5)の $\Delta R$ に相当）する。ここで、式(5)の拡大係数行列を示す。

$$(A|\Delta R) = \left( \begin{array}{ccc|c} \alpha_1 & \beta_1 & \gamma_1 & \Delta R_1 \\ \alpha_2 & \beta_2 & \gamma_2 & \Delta R_2 \\ \alpha_3 & \beta_3 & \gamma_3 & \Delta R_3 \end{array} \right)$$

拡大係数行列で表された連立方程式はガウスの消去法によってその解を得ることができる。これより以上で求めた  $A$  に相当する二次元配列と、 $\Delta R$ に相当する一次元配列からガウスの消去法によって $\Delta X$ を求める。この時、修正値 $\Delta X$ は $\Delta R$ が格納された一次元配列に上書きされる。そして、 $\Delta X$ の各要素についてその大きさを判別し、十分小さくなければ初期値に $\Delta X$ を加算して行列  $A$  となる二次元配列を求めるところへ戻る。 $\Delta X$ の各要素が十分小さいときはその時の初期値を解として終了する。

## 6.2.3 データ出力の流れ

6.2.2 で求めた落雷地点の座標をファイルへ出力する。出力するデータは発雷時刻と落雷地点の座標であり、発雷時刻は 6.2.1 で構造体に入力されたデータを用いる。出力データは出力ファイルのデータの末尾に記録する（図 10）。

```
2021/09/18 10:56:49(0.000034, 9.999732, 8.000920)
2021/09/25 14:32:40(0.000269, 19.999217, 8.001013)
2021/09/25 14:34:10(19.999468, 0.000724, 8.000385)
2021/10/08 09:09:27(20.000403, 9.999471, 7.999762)
2021/10/15 02:20:56(29.999730, 30.000399, 7.999952)
```

図 10 出力ファイルの例

## 6.3 設置地点の座標とプログラムの検証

ここでは設置地点の具体的な座標と、それを用いた試験的なプログラムの運用例を示す。

### 6.3.1 設置地点の座標について

観測機器の設置地点の座標は香川県三豊市の設置場所を基準にしてその他の地点の座標

を割り出した。各地点の平面座標は香川県三豊市を(0, 0)として、緯度経度からほかの地点との距離の差を x, y 軸ごとに求めその地点の座標とした。z 軸の値にはその地点の標高を用いる。それぞれの観測地点の座標を次の表 2 に示す。

表 2 観測地点の座標

観測地点	座標 (x [km], y [km], z [km])
香川県三豊市	(0, 0, 0.017)
香川県高松市	(38.452876, 9.376886, 0.0115)
岡山県玉野市	(31.348301, 33.344306, 0.00542)

### 6.3.2 検証

落雷地点の座標を決め、逆算した各観測地点と落雷地点との距離をプログラムに入力し、正しく落雷地点の座標を求められるか検証した。表 3 に結果をまとめる。この時、各観測地点の座標には表 2 の値を、初期値には(10.0, 10.0, 10.0)という値を使用した。

表 3 プログラムの検証

設定値 [km]				計算結果 [km]
落雷地点 P	距離 R <sub>1</sub> -P	距離 R <sub>2</sub> -P	距離 R <sub>3</sub> -P	落雷地点 P'
(0, 0, 8)	7.983	40.378	46.459	(-0.242821, -0.089584, 8.063142)
(0, 10, 8)	12.796	39.279	39.895	(0.000034, 9.999732, 8.000920)
(0, 20, 8)	21.534	40.685	34.996	(0.000269, 19.999217, 8.001013)
(0, 30, 8)	31.044	44.359	32.524	(0.000541, 29.999563, 8.001746)
(10, 0, 8)	12.796	31.005	40.392	(10.000120, -0.000088, 8.000434)
(10, 10, 8)	16.240	29.560	32.629	(9.990676, 9.995768, 8.156968)
(10, 20, 8)	23.743	31.404	26.415	(10.000474, 19.999305, 8.001259)
(10, 30, 8)	32.615	36.037	23.040	(9.993402, 29.995665, 8.180905)
(20, 0, 8)	21.534	22.187	36.118	(19.999468, 0.000724, 8.000385)
(20, 10, 8)	23.743	20.117	27.160	(20.000403, 9.999471, 7.999762)
(20, 20, 8)	29.389	22.741	19.255	(20.000133, 19.999841, 7.999128)
(20, 30, 8)	36.929	28.803	14.279	(20.000733, 29.999529, 8.001357)
(30, 0, 8)	31.044	14.940	34.316	(29.999911, -0.000100, 8.000437)
(30, 10, 8)	32.615	11.647	24.712	(30.014822, 9.968768, 8.186060)
(30, 20, 8)	36.929	15.752	15.614	(30.000123, 20.000296, 8.000219)
(30, 30, 8)	43.171	23.677	8.770	(29.999730, 30.000399, 7.999952)

表 3 より、どの場合においても落雷地点の近似値を正確に求めることができていることが分かる。設定値と計算結果の誤差は大きいところでおおよそ 240 m、小さいところでは 1 m 未満と、ある程度の精度をもっていることが確認できる。

## 7 落雷の観測と結果

今回は「Raspberry Pi」を使用した室内からの落雷の観測と、「Raspberry Pi Zero」を使用した落雷の観測、「稲妻ルーラー」を 3 地点に設置して落雷の発生地点の観測を行った。

室内からの観測と「Raspberry pi Zero」を用いた観測については、今後「稲妻ルーラー」をさらに発展させるために、実用性の向上とより安価な製作を目的に行った。

### 7.1 室内からの落雷の観測

5.2 の「観測を行うプログラム」でも説明したように、稲妻ルーラーは屋内と屋外での設置を指定することでどちらでも観測を行うことができる。しかし、過去に制作してきた稲妻ルーラーでは屋外での観測しか行ったことがないため、実際に屋内での観測が可能であるかが不明であった。そのため今回は 7 月から 8 月中旬までの間、稲妻ルーラーを室内に設置して観測を行った。

2021/07/14 14:13:14 , lightning! , 10 , 60130
2021/07/14 14:15:28 , lightning! , 10 , 8603
2021/07/14 14:19:39 , lightning! , 10 , 11776
2021/07/14 14:28:29 , lightning! , 10 , 26101
2021/07/14 14:29:31 , lightning! , 10 , 3230
2021/07/14 14:29:46 , lightning! , 10 , 11290
2021/07/14 14:31:10 , lightning! , 12 , 38396
2021/07/14 14:32:37 , lightning! , 12 , 33678
2021/07/14 14:34:38 , lightning! , 12 , 4146
2021/07/14 14:36:10 , lightning! , 12 , 29318
2021/07/14 14:36:41 , lightning! , 12 , 20676
2021/07/14 14:42:17 , lightning! , 12 , 12547
2021/07/14 14:44:14 , lightning! , 12 , 6328
2021/07/14 14:45:11 , lightning! , 12 , 7970
2021/07/14 14:46:17 , lightning! , 12 , 2446

これは2021年の7月14日に室内に設置したセンサーが記録したデータの一部である。この日の天候は雨で、落雷情報を閲覧できるWebサイトなどでもこの時間にセンサーを設置した地点の周辺で何度も落雷が発生していることを確認できた。また14時13分から29分までが10 kmで14時31分からその後は12 kmと表示されている。この結果から雷雲が移動して観測地点から離れていることが確認できる。

観測の結果より稲妻ルーラーは室内でも観測が可能であると考えられる。しかし、室内ではセンサーの周辺などにノイズを発生させる電子機器や屋外に比べて落雷による電磁波を受信することが難しいことから、屋外での設置の方がより正確であると考えられる。

## 7.2 Raspberry Pi Zero を使用した観測

「Raspberry Pi Zero」は通常の「Raspberry Pi」よりも小型で安価に入手することができる。そのため「Raspberry Pi Zero」を稲妻ルーラーに搭載し観測を行うことができれば今よりもコストを削減して稲妻ルーラーを製作することができる。

今回は3地点で落雷を観測することを想定して3つのセンサーと「Raspberry Pi Zero」を用意しそれぞれで観測を行った。しかし、記録されていたデータは3つとも違い、全てのセンサーが正常に動作をしていなかった。1つ目の機器は、観測期間中にはほぼ毎秒でノイズを感知して「Noise level too high」とテキストファイルに出力していた。そのため記録しているテキストファイルの情報量がとても大きくなった。2つ目の機器は起動して観測を開始してから一切のデータを取得できなかった。3つ目の機器は観測を開始すると常に「lightning!」のみをテキストファイルに出力していた。またその距離は1 kmで常にセンサーの真上で誤って落雷を検知していた。

稲妻ルーラー3の製作を開始する以前はセンサーの制御に「Raspberry Pi Zero」を使用する予定だったが、この観測実験で一切のデータを取得できなかったため制御機器はこれまで通り「Raspberry Pi」を使用することにした。

## 7.3 落雷の発生地点の観測

稲妻ルーラーで使用されるセンサーは落雷との距離を推定することができても具体的な方位、つまり1つでは正確な発生地点を測定することはできない。そこで今回はセンサーの数を増やし、6.1の「計算方法」で説明した内容を基に落雷の発生地点の観測を行う。

今回は香川県で高松市と三豊市、岡山県で玉野市の3地点に機器を設置して約2週間の観測を行った。しかし、こちらの観測も「Raspberry Pi Zero」を使用した際と同じような結果になった。三豊市と玉野市に設置した機器は一切のデータを取得することができず、また高松市の観測機器は「Raspberry Pi Zero」を使用した3つ目の機器と同じように常に頭上で

雷が検知されていた。しかし高松市の観測機器は2021年10月22日の19時から20時にかけて約5 km離れた地点で落雷の発生を記録していた。この時間帯は雨が激しく降っており落雷も実際に発生していたため、正確に観測できたデータだと思われる。

## 9 考察

今回の観測は3点から落雷の距離データを取得し、それを基に落雷の発生地点を推定することだったが、それぞれの観測機器が正常に動作しなかったため3つの距離データを取得することができなかった。「Raspberry Pi Zero」を使用して落雷の観測を行った際にデータの取得が一切できなかったことから、正確な観測ができない原因は「Raspberry Pi Zero」を使用したことにあると考えたが、その次の観測で「Raspberry Pi」でも同じ誤作動が起こっていた。そのため観測が正確に行えなかった原因はプログラム自体にあると考えられる。

雷センサーを制御するプログラムには使用する「Raspberry Pi」の種類を指定したり、レジスタの値を調整し、ノイズに対しての閾値を変更したりする内容がある。今回は制御するコンピューターが変更された際、作成したプログラムを対応したものに書き換えなかったことが原因だと考えられる。レジスタの値を変えてノイズに対する閾値を正常にすれば、「Noise level too high」や「lightning!」が常に出力されるのを改善することができると考えられる。

また、データが一切取得できないのはセンサーが動作していないからでそれは「Raspberry Pi」がセンサーの制御ができていないことが原因だと思われる。これはプログラムを作成する段階で使用する「Raspberry Pi」の種類を正しく指定することで改善されると考えられる。

## 10 感想

私は、今回の開発で観測を主に担当しました。稲妻ルーラーの観測はとても大変で、センサーの性能やシステムの動作確認をする段階でも、なかなか発生しない雷を待つ必要があり、問題が発生したときに改善がなかなか進まずもどかしい思いをしました。また、当初の予定ではインターネットにつなぎ、逐次オンライン上でデータを更新する予定でしたが、雷の観測がなかなかうまくいかず、そこまでの段階にたどり着くことができなかつたことは残念でした。その一方で今回は、これまでの稲妻ルーラーの機能を大きく拡張することを目指して頑張ってきましたが、観測ができなかつたことを除けばこの点について達成できたと思っています。

機械工学科 3年 大西 克也

今回、私は主に落雷地点の算出プログラムの開発に携わりました。私は、これまで学校の授業などでもプログラムを作成してきましたが、この算出プログラムは私が作ってきたプログラムの中でもとてもオリジナルの部分が多いものとなりました。このプログラムを作成するにあたりいくつか参考にしたものはありますが、多くの部分においてこれまで授業で習った知識を最大限生かし目的としていた機能を実現することができ、とてもよかつたと思っています。今回、稲妻ルーラーを3機用意することになり、その設定や観測がなかなかうまくいかず大変でしたが、算出プログラムをひとまずの完成に持ち込めたことは私にとって大きな成果だと思っています。

電気情報工学科 3年 和気 佑弥

## 参考文献

ナチュラル研究所

[https://www.ishikawa-lab.com/RasPi\\_lightning.html](https://www.ishikawa-lab.com/RasPi_lightning.html)

AS3935使用雷センサマニュアル

[http://www.ne.jp/asahi/shared/o-family/ElecRoom/AVRMCOM/AS3935/AS3935\\_test.html](http://www.ne.jp/asahi/shared/o-family/ElecRoom/AVRMCOM/AS3935/AS3935_test.html)

あなたの生活情報 – 過去の雷観測

<https://weather.kakutyoutakaki.com/thander/index.php>

GPS 測位計算プログラム入門

[https://www.enri.go.jp/~fks442/K\\_MUSEN/](https://www.enri.go.jp/~fks442/K_MUSEN/)