卒業研究報告書

マイコンを用いたインフラサウンドイベント判断および

自動記録システムの製作

報 告 者

学籍番号:1200180

氏名: 米沢 一輝

指 導 教 員

山本 真行 教授

令和 6 年 2 月 16 日

高知工科大学 システム工学群 電子・光工学専攻

目次

第1章 序論

1.1 背景

1.1.1 インフラサウンド

インフラサウンドとは人間の可聴域外である周波数 20 Hz 以下の音波のことで、可聴域の音波より減衰しにくく遠方でも観測することが可能である。インフラサウンドは地震、火山噴火、雷などで発生し、これらの事象によるインフラサウンドを遠方から観測することが可能である。

1.1.2 従来の記録方法

高知工科大学システム工学群宇宙探査システム研究室（山本研究室）ではこれまで従来はインフラサウンドの記録をインターネット経由で取得する方法や大容量のストレージに現地保存しそれを定期的に回収することで記録を蓄積してきた。これによりメンテナンスの労力や停電などの電源の問題が発生し、これらを原因とする様々な障害がデータ回収効率を下げることとなった。同研究室の井上らはデータ記録用ノート PC を 2021 年にシングルボードコンピュータの Raspberry Pi Zero を用いた収録装置（データロガー）を製作してアップデート（可搬性を高め海外での臨時観測を成功に導いた）したが、大容量バッテリーや商用電源がある地点でしか観測運用が行えなかった[1]。

1.2 目的

ここまで述べてきたように、現状のデータ回収方法には改善の余地がある。ノート PC や Raspberry Pi などを利用したデータ回収のみでは十分に改善されたとは言えない。そこで、本研究では無線通信を用いたデータ回収手法を検討する。現在では無線モジュールの発展に伴い、低用量通信であれば、省電力かつ長距離の通信ができるよ

うになってきた。しかし大容量のデータを送ることはできないため、観測点においてマイコンを用いてインフラサウンドのイベントを自動検出しその前後のデータのみを記録することで、データサイズを小さくすることで問題を解決する。そこで本研究では、マイコンに実装できるインフラサウンドのイベント検出用のアルゴリズムおよびそれを実装した観測機器の試作開発を目的とする。

第2章　イベント検出とデータ記録方法

2.1　イベントの検出

2.1.1　しきい値法

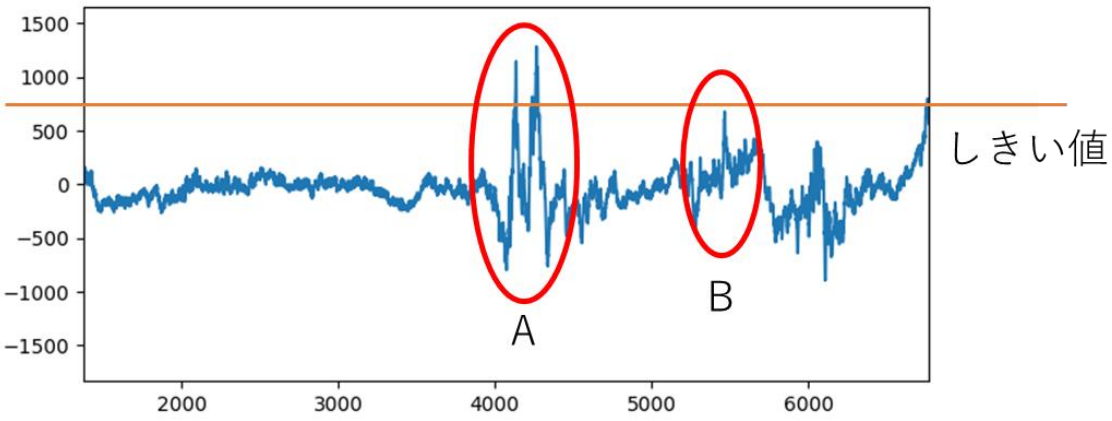

図 2.1　しきい値の設定とインフラサウンドデータ波形の事例

図 2.1 のように、インフラサウンド波形に対して、しきい値を設定しそれを超えたときにイベントがあったと判断する手法である。この方法は必要な計算が少なくマイコンへの実装が容易でマイコンの計算量も少なくて済む。しかし、しきい値を超えないような小さな信号（図 2.1 に示す波形 B）を検出することができない。

2.1.2 STA/LTA 法

地震学で使われている手法[2]であり、ノイズが多い環境でもノイズによる影響を抑えることができる。

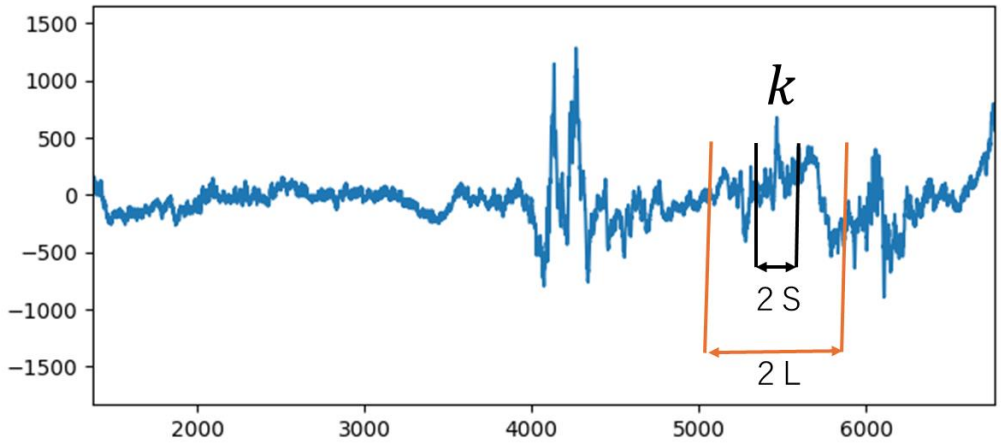

図2.2 STA/LTA 法の説明

本手法では図2.2 のように STA(short time average)と LTA(long time average)という短時間と長時間の 2 つの時間幅での移動平均をとり、その比によりイベントを検出する手法である。

$$\frac{STA}{LTA} = \frac{\sum_{i=k-S}^{k+S} \frac{|X_i|}{2S}}{\sum_{i=K-L}^{k+L} \frac{|X_i|}{2L}} \tag{2.1}$$

$\bar{x}$: 平均　N: データ数(個)

式(2.1)は平均値の計算方法で、移動平均ではこれを新しいデータが来るたびに順次計算する。本手法ではデータ数を変えて 2 つの移動平均をとる。そして STA は LTA と中心が同じ位置に来るようにする。STA/LTA 法はイベント信号の長さにより比が変わり、イベント波形による変動が、STA 区間に収まる長さの信号だと比が大きくなり、STA 区間に収まらなくなると小さくなる。これにより風などによる比較的長時間にわたり継続する振動の影響を抑えることができる。

2.2 データの記録方法とフォーマット

本研究で開発するシステムでは、2.1.2 項の手法により検出したイベントの前後区間のデータを EEPROM 上に保存する仕様とした。

2046バイト					
AD変換回数	データ0	データ1	・・・・	データ1019	データ1020
4バイト	2バイト	2バイト	・・・・	2バイト	2バイト

図 2.3 EEPROM 上への記録のフォーマット

図 2.3 は本研究で策定した EEPROM に記録する際のフォーマットの 1 例である。最初の 4 バイトは時間の代わりに起動時からサンプリング（AD 変換）した回数を 4 バイト整数で記録している。残りは記録したデータを 2 バイトで順次 1021 個保存している。プログラムを変更すればデータ長を変えることができる。

第3章 観測機器の開発

3.1 観測機器の構成

本研究で使用した主な機器および主な部品を表 3.1 にまとめた。

表 3.1 使用部品

型番	メーカー
ATMEGA4809-PF	Microchip Technology
24FC256-I/P	Microchip Technology
NJM2884U2-05	日清紡マイクロデバイス
INF03S	株式会社サヤ

これらの部品(表 3.1)のうち INF03S 以外は秋月電子通商より購入した。本研究では消費電力を少なくすることが主目的ではないが、実際の使用場面では消費電力が少ないほうが有用であり、これを考えて使用するマイコンを選定した(表 3.2)。

表 3.2 マイコンの消費電流比較

	Raspberry Pi Pico	ATMEGA4809-PF
消費電流	9.4 (mA) [3]	11.4 (mA) ～ 21(μA) [4]
動作周波数	133 (MHz)	20 (MHz)～32.768 (kHz)

表 3.2 は 32bit のマイコンである Raspberry Pi Pico と今回用いた ATMEGA4809-PF を比較したものである。動作周波数により消費電力は変わるが、ATMEGA4809-PF の動作周波数を低くすると Raspberry Pi Pico より消費電流を少なくすることができると期待されるためこのマイコンを使った。

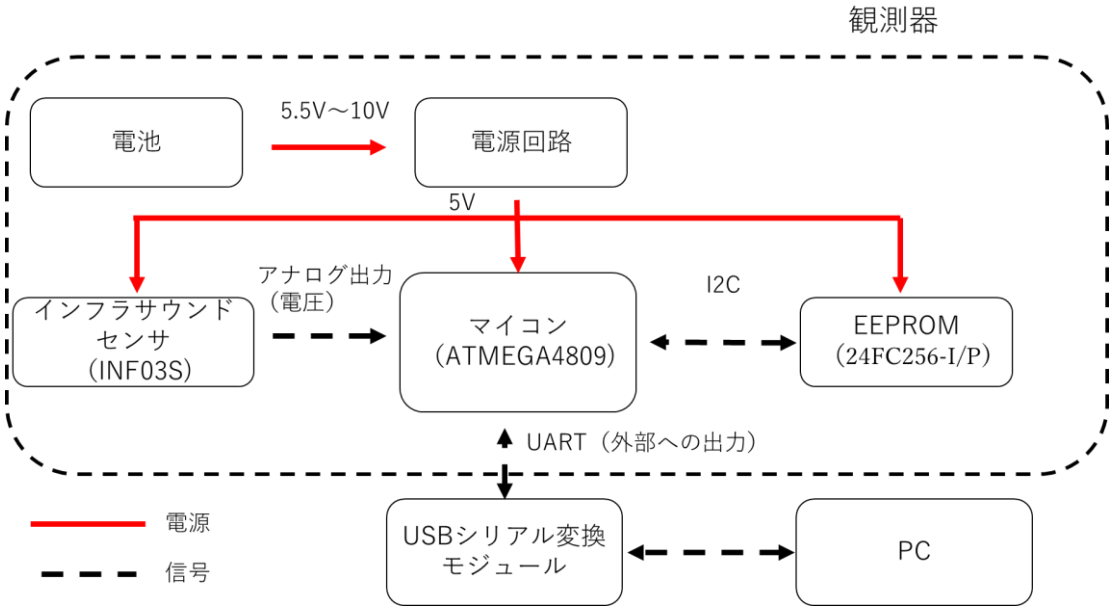

図 3.1　観測器のシステム構成

構成は図 3.1 のようになっている。破線で囲まれている部分が観測器になっており、破線の外側にある USB シリアル変換モジュールと PC は計測データ確認用であり観測器に含まれない。観測器ではイベントの検出、EEPROM への記録、EEPROM からの読み出しを行っている。破線の外側には実運用では通信モジュールが接続されるが、本研究では EEPROM への記録内容を確認するための読み出し時にのみ使用する。

3.2 観測器の製作(ハードウェア)

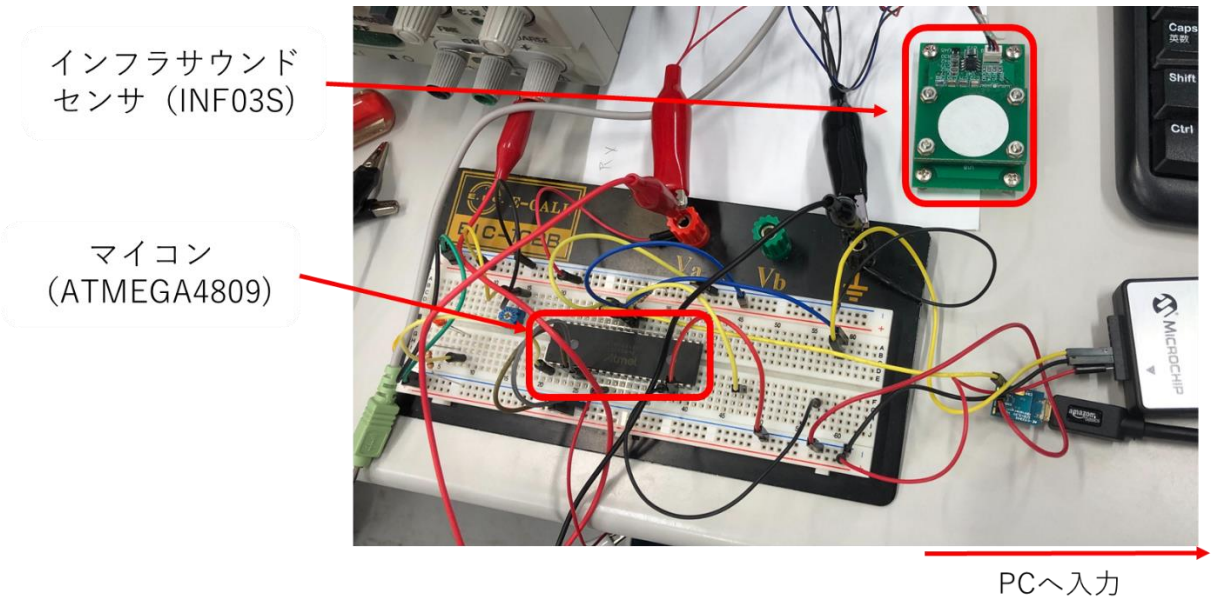

図 3.2 ブレッドボード上のテストの様子

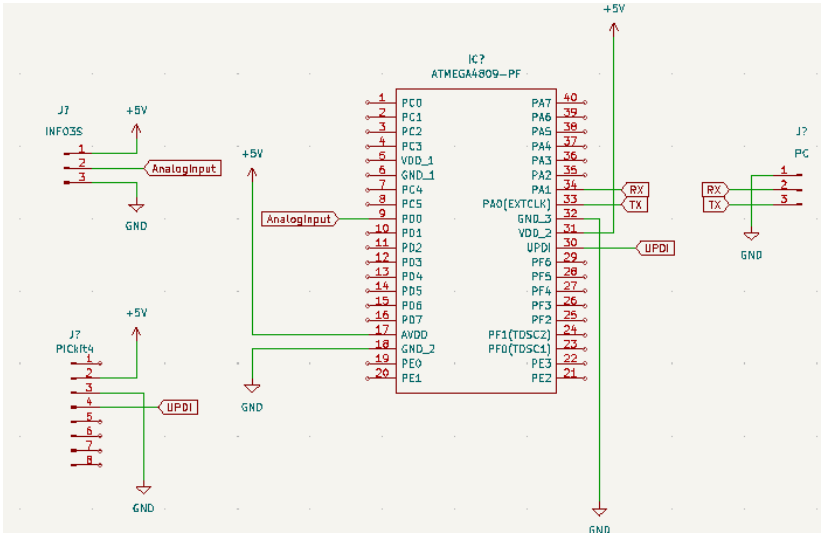

図 3.3 ブレッドボード上の回路図

ブレッドボード上でマイコンへの書き込み、インフラサウンドセンサの出力電圧の読み取り、PC との通信をテストした結果どれも問題なく動作の確認ができた。この動作試験を経て実際に使うことを想定したプリント基板を製作した。

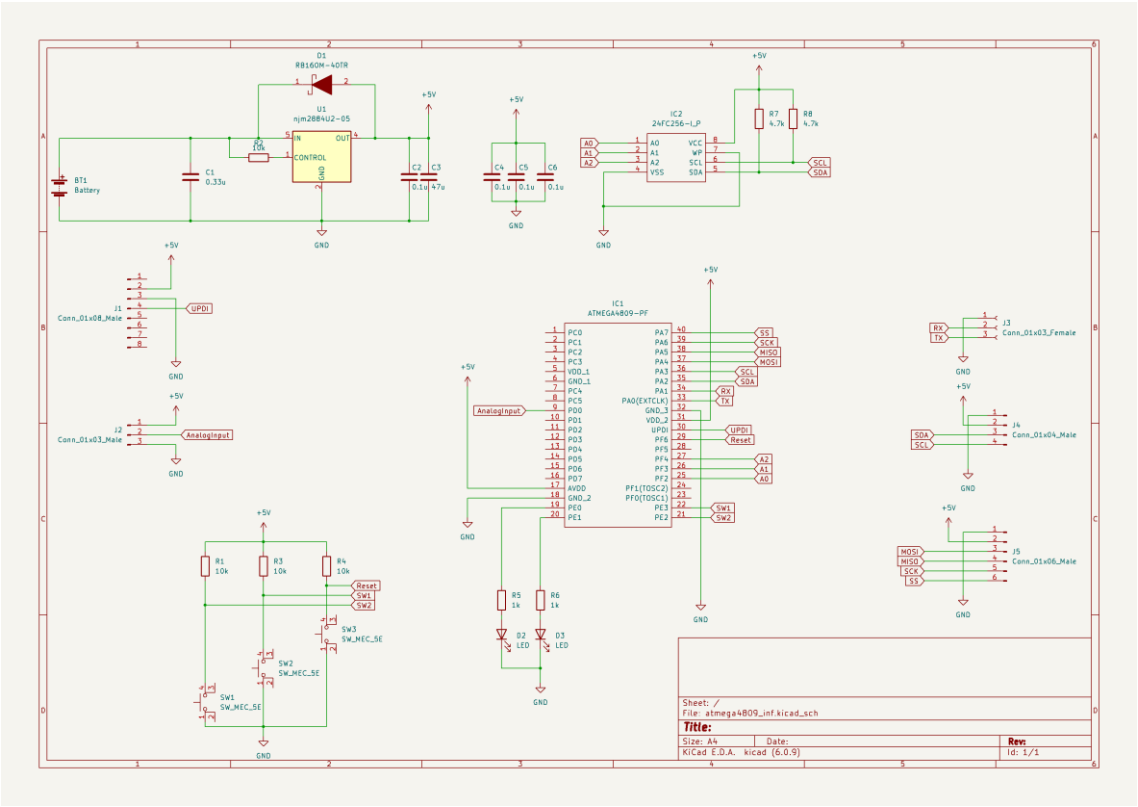

図 3.4　回路図

図 3.4 は回路図で、後のこと考えて別のセンサも使えるように SPI および I2C 接続で通信できるコネクタ、状態の確認のための LED および機能の切り替えのためのスイッチを取り付けた。

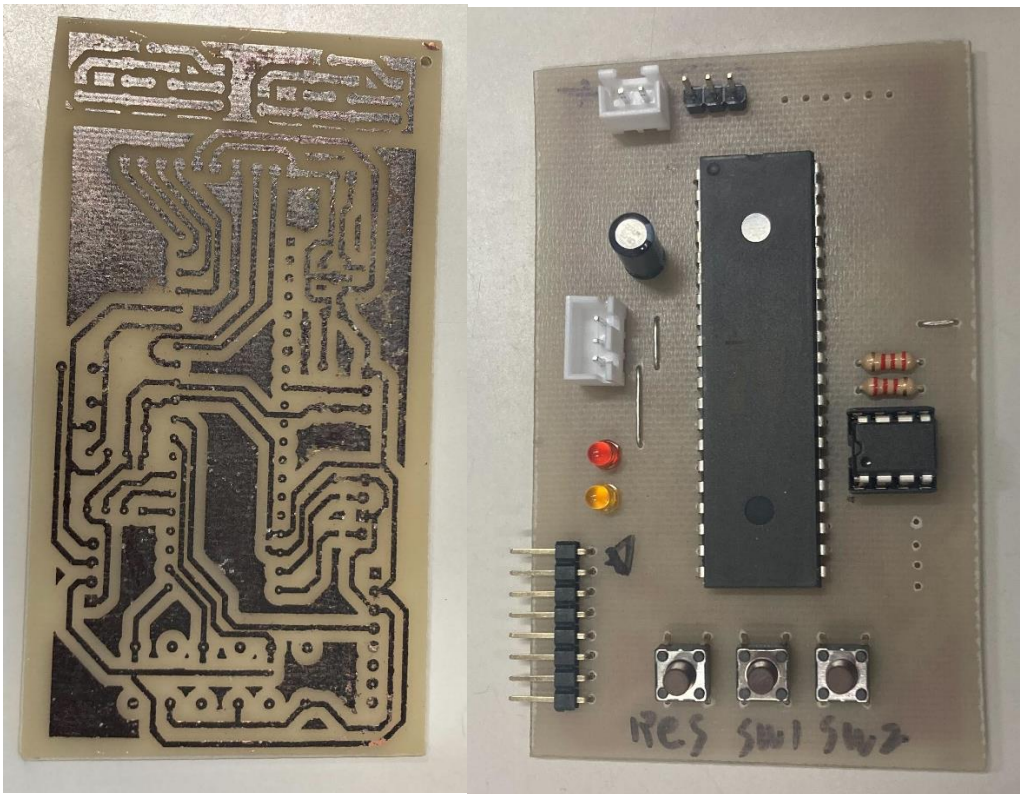

図 3.5　観測器

図 3.4 の回路図をもとに製作した基板の外観を図 3.4 に示すこれは、銅箔の不要部分を酸で溶かすエッチングという手法を用いて製作した。図 3.5 左はエッチングが終わった後の基板で、図 3.5 右はそれに部品を載せるための穴あけ加工をして、部品を実装した状態である。回路図では別のセンサを載せるための拡張用コネクタがあったが今回は不要であり、その部分のみ基板への実装を行っていない。

3.3　観測器の製作（ソフトウェア）

3.3.1　マイコンに実装するための効率的な演算

イベントの検出に必要な演算は STA/LTA 法で用いる 2 つの移動平均と比を計算するための除算である。まず移動平均については新しいデータが来るたび合計を計算するのは効率的ではないので、新しいデータを1つ足して最も古いデータを1つ引く

という方法を使ってデータの合計数が常に一定になるようにする。図 3.5 に計算イメージを示す。

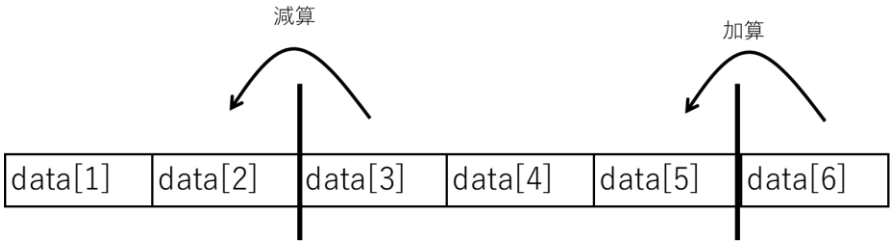

図 3.6　合計値の計算イメージ

これを何度も行う必要があるため、通常の配列ではなくリングバッファを使用した。リングバッファは古いデータを消しながら新しいデータを保存することができ、配列の先頭と末尾がつながっているバッファである。

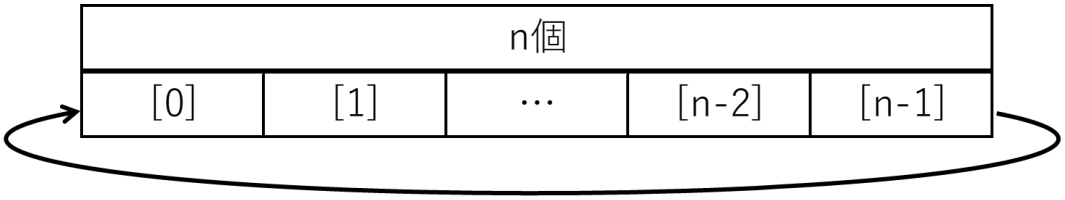

図 3.7　配列とリングバッファ

図 3.7 は配列を使ってリングバッファを実現するための模式図である。配列の要素数が n-1 を超えたら 0 に戻すようにする。これをするには配列の要素数の剰余をとるとできるが、今回はビット演算のみで済むようにした。n を 2 の累乗に指定しておけば、n-1 で論理積をとると剰余をとった時と同じ結果になることを利用して実装した。

平均値や比を求める際の除算は加算減算と比較して時間のかかる演算である。そして除算でも実数と整数の除算では速度に差があり、今回は整数の除算のみを行う。整数の除算では小数点以下が算出されないため計算精度が落ちる。これを解決するために除算する前に左にビットシフト（今回は 10 bit 分で固定）することで桁数を増やした後に除算している。

3.3.2 イベント発生時のデータの記録

イベントが発生した際のデータの記録は 24LC256(Microchip 社製)という EEPROM を使っている。この EEPROM は 256 kbit なので 2046 byte のデータを 16 個保存することができる。保存をする際には I2C で書き込みを行っており、その際には通信に CPU を占有されるのでイベントの検出は行えなくなっている。このため書き込み中は AD 変換を停止し EEPROM への書き込みが完了したときに AD 変換を再開する仕様とした。これによる欠測時間は約 2 秒である。

3.3.3 電源投入時の機能選択

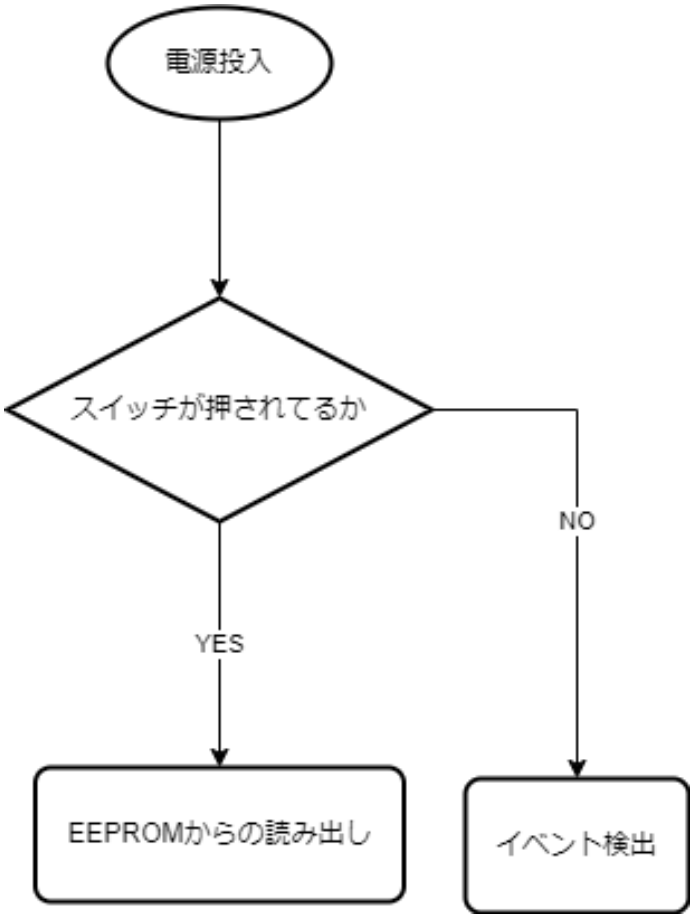

図 3.9 電源投入時のフローチャート

図 3.9 は電源投入した時のフローチャートである。スイッチの ON/OFF によりイベント検出をするか EEPROM からの読み出しをするか決めることができ EEPROM を読み出す際に他の機器を使う必要がなくなる。

3.3.4 イベント検出のための処理

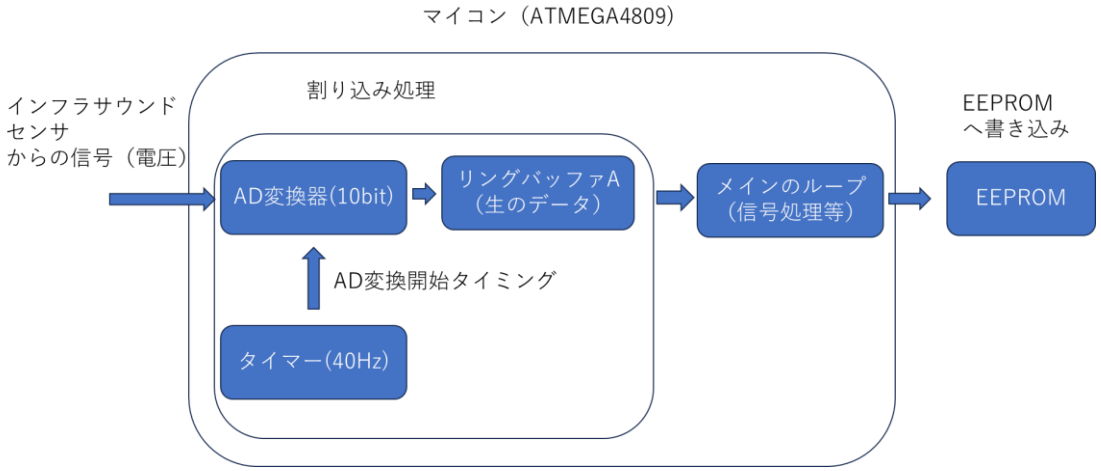

図 3.10　イベント検出での処理フロー

図 3.10 はマイコンでの処理の流れを説明するためのブロック図である。タイマー割り込みを用いて定期的に（ここでは 40 Hz で）インフラサウンドセンサの出力電圧をサンプリングしてリングバッファ A に格納する。メインのループではリングバッファ A に格納された結果を用いてイベントが発生していないか確認し、イベントが発生していればその前後のデータを EEPROM に書き込む。

ATMEGA4809-PF マイコンの AD 変換器は分解能が 10bit であり、量子化単位(LSB)は以下の式(3.1)のようになる。

$$LSB = \frac{vref}{2^n} \tag{3.1}$$

LSB: 量子化単位(V)　$vref$: 基準電圧(V)　n: 分解能

ここで、基準電圧はマイコンの電源電圧と同じ 5 V であり、分解能（量子化ビット数）10 bit の ADC なので量子化単位は約 4.9 mV になる。従って 4.9 mV より低い電圧の変

化は測定することができない。

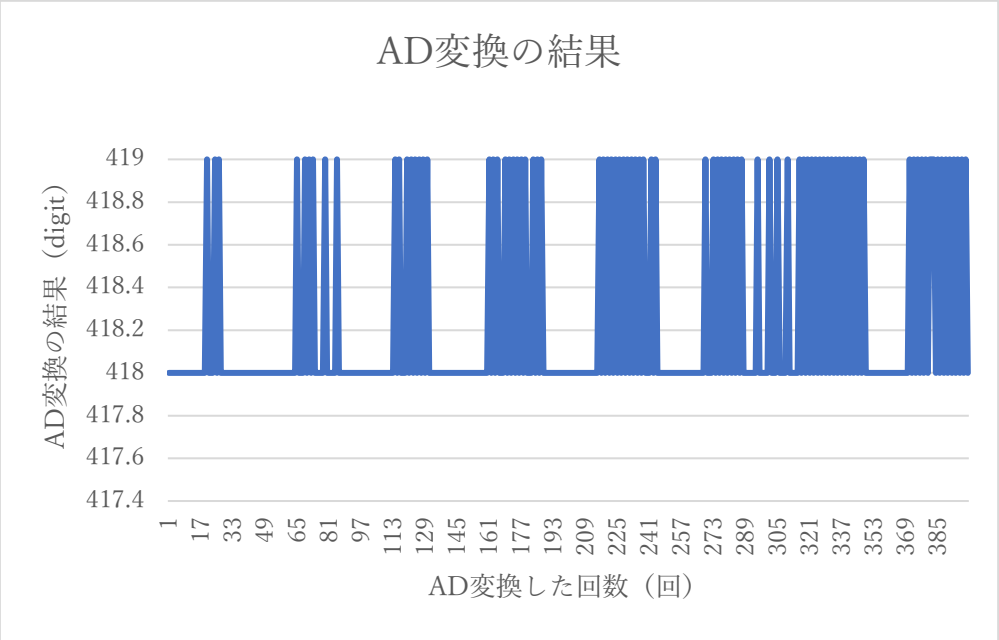

図 3.11　AD 変換の結果

図 3.11 は AD 変換の結果を PC シリアル通信で送りプロットしたものである。インフラサウンドセンサ INF03S はほぼ一定のバイアス電圧（ここでは約 2 V）を中心にインフラサウンドによる差圧成分（微小）気圧変化を電圧として出力しているので、ADC の結果が 419digit 付近を前後している。STA/LTA 法では 2 つの移動平均の比を使ってイベントの判断をしているため、オフセットをこのまま取り扱うと算出される比が小さくなる。このため計算前にオフセットの位置を 0 としてオフセットを除去する必要がある。そのために一つ前のデータとの差分をとることにした。

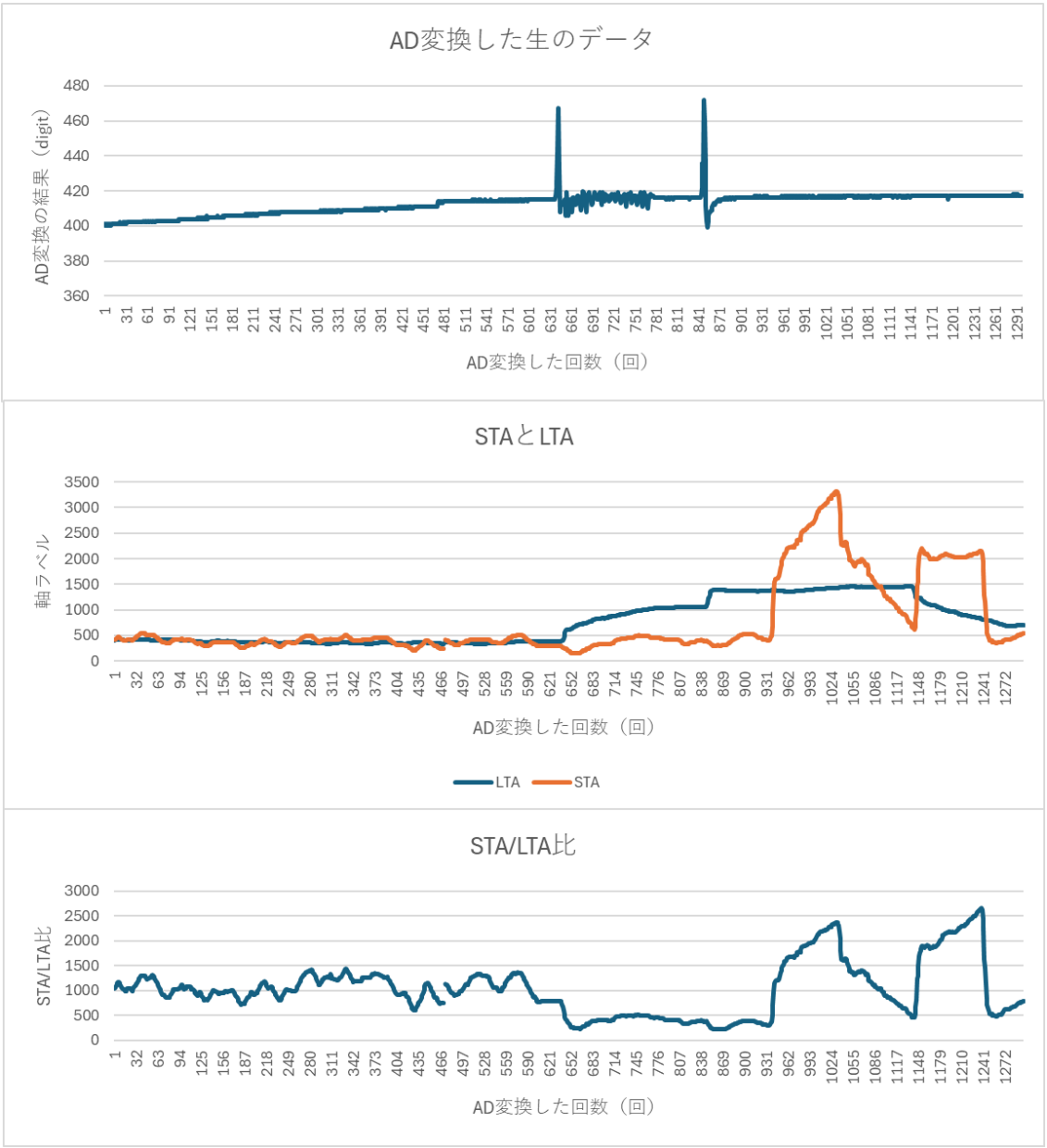

図 3.12　マイコンでの計算結果

図 3.12 のグラフは 40 Hz で AD 変換したイベントデータ例について STA/LTA 法の計算をした結果で、1 回ごとの計算が終わるたび PC にシリアル通信で送信しそれをエクセルでプロットしたものである。約 30 秒間の結果で、約 1200 個のデータが来ており STA/LTA 法をリアルタイムに計算できることが確認できた

3.3.5 EEPROM からの読み出しの処理

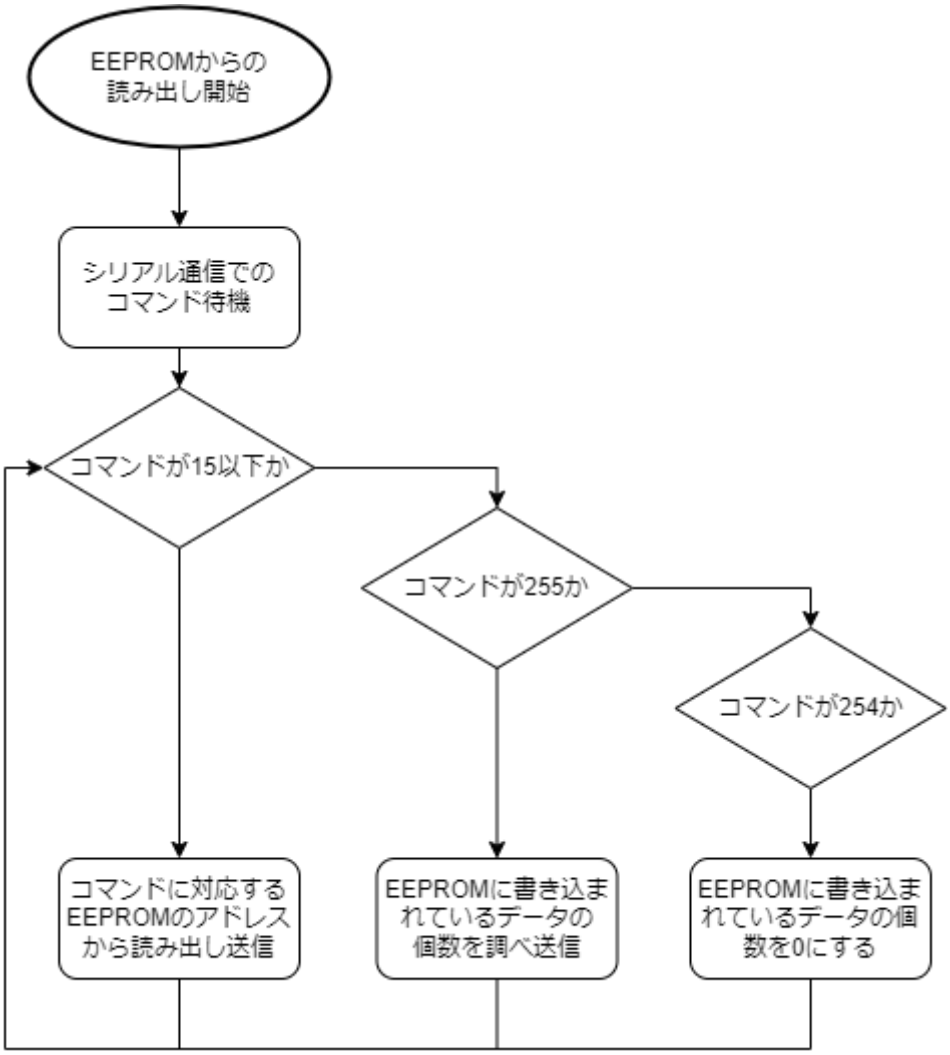

図 3.13 読み出しのフローチャート

EEPROM に記録されたデータをシリアル通信で読み出す際の処理を図 3.13 に示す。このフローでは、USB ケーブルで接続された PC より 1 バイトのコマンドを送信し結果を返すようになっている。結果は ascii 形式で送信されるので、PC 上の適当なシリアルモニタで読み出しの結果を受け取ることができる。その結果をエクセルに張り付けグラフにすることも容易である。ここで用いる 1 バイトコマンドは0～15, 254, 255 の 18 種類である(図 3.13 参照)。

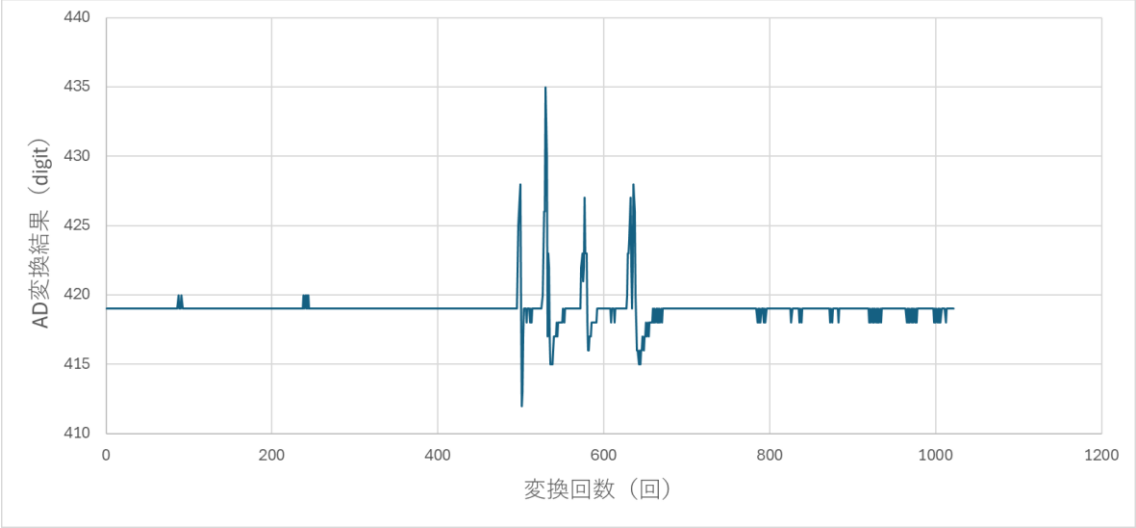

図 3.14　センサの直上（約 10 cm）で軽く手をたたいた際の波形

図 3.14 は EEPROM への保存と読み出しのテスト結果である。手を 4 回たたいた時の音波の波形が再現できているため、リングバッファを用いた AD 変換および EEPROM への読み書き（1021 バイト長）が問題なく実装できていることが確認された。

3.3.7　ソースコード解説

```
typedef struct{
    uint32_t sum;//合計を入れとくとこ
    uint32_t ave;/////平均値入れとく
    uint16_t sum_head;//足したとこまでのヘッド
    uint16_t sum_tail;//足したとこまでのテール
    uint16_t sum_num;//足すつもりのデータ数
    uint16_t offset;//オフセット
    uint16_t addr;
    uint16_t current_num;//足したデータ数の合計
}Sum_obj;
```

図 3.15　移動平均のための構造体

本稿では実装したプログラムのソースコードの一部を解説する。まず 3.3.1 項の手法を実装するために必要な変数を構造体にしてまとめている図 3.15。こうすることで複数の移動平均を求める際に多くの変数を用意する必要がなくなり、簡潔にまとめて記述することができる。

```
Sum_obj sum_add(Sum_obj obj,uint16_UNION* buff,uint16_t head){
    if (obj.current_num == 0)
    {
        obj.sum_head = buff_addr(head,-obj.offset);
        obj.sum_tail = obj.sum_head;
        obj.sum = 0;
    }
    /*
    if(obj.offset != 0)//最初のいくつかを飛ばす　たぶん使わない
    {
        //obj.sum_head = buff_addr(obj.sum_head,1);
        obj.offset--;
        return obj;//offsetがゼロになるまでこれする
    }
    */
    obj.sum += buff[obj.sum_head].big;//buffから足してく current_numも増やしとくヘッドも増やす
    obj.sum_head = buff_addr(obj.sum_head,1);//次のアドレス
    obj.current_num++;

    if (obj.current_num>obj.sum_num)//足す数が予定を超えたらtailから引いてく　tailも増やす
    {
        obj.sum -= buff[obj.sum_tail].big;
        obj.sum_tail = buff_addr(obj.sum_tail,1);
        obj.current_num--;
    }
    if (obj.current_num == obj.sum_num)
    {
        obj.ave = (obj.sum<<10) /obj.sum_num;
    }
    return obj;
}
```

図 3.16　移動平均計算のための関数

図 3.16 に示す(sum_add)関数は引数に図 3.15 で説明した構造体を渡し、それをもとに移動平均を求めるための関数である。3.3.1 項の手法をほぼそのまま実装しているが、移動平均の合計値をリセットするための機能があり、上から 2 行目の if 文の中に記述している。足したデータ数の合計の値を0にしたとき、移動平均の合計値をリセットし最初からやり直せるようにしている。

```
#define long_time 500
#define short_time 96
#define throw 800-long_time /2
#define trigger 2048
#define bit_wari 4
#define ctrlByte 0b1010000
#define eeprom_siri 32767
```

図 3.17　パラメーターの定義

図 3.17 のように STA と LTA の移動平均の長さなどイベントの判定に使うパラメーターをプログラムの初めに定義しており、これらの変更を容易にして実験を行いやすくしている。

第4章　評価・考察

4.1　イベント検出

4.1.1　測定した条件

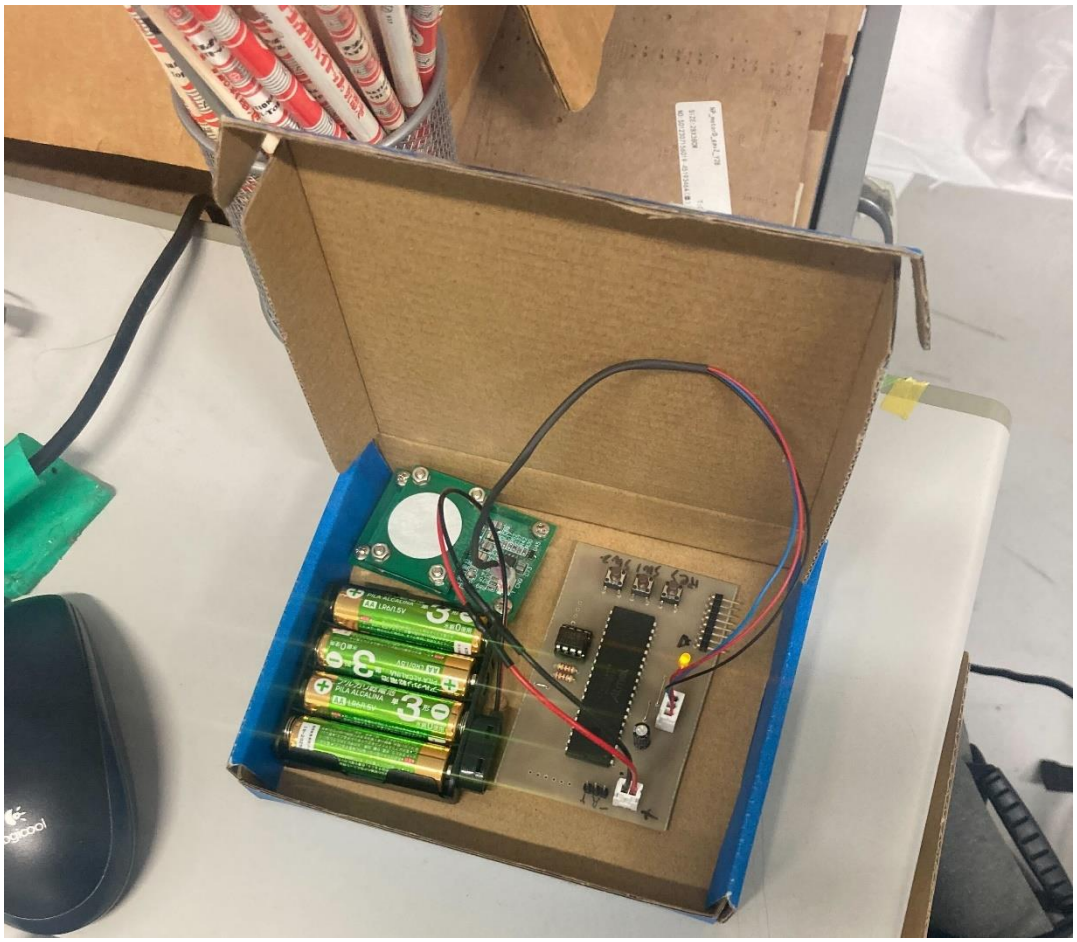

図 4.1　観測器の外観

（A151 実験室）の入り口から 6 m ほど離れた机の上に設置し、ドアの開閉を行うことで開閉による圧力の変化をイベントとして検出できるか試験した。

4.1.2　イベント検出のためのパラメーター設定

パラメーターを設定するためにドアの開閉でどのような波形が記録されるのかを確認した。

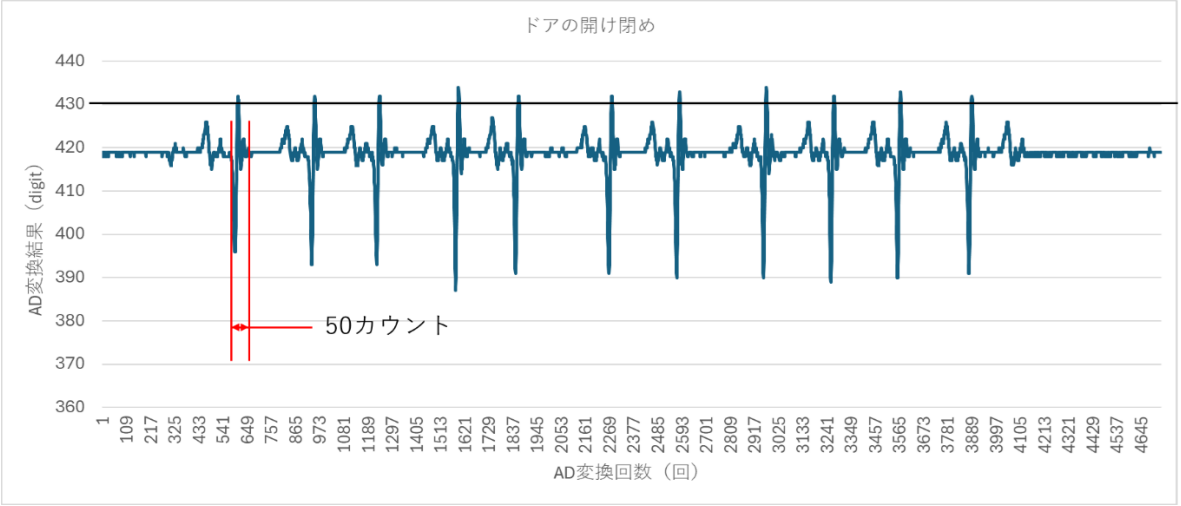

図 4.2 ドアの開閉時の電圧の変化

11 回のドアの開閉で図 4.2 のような結果になった。この図からパラメーターを決める。しきい値のパラメーターは 11 回すべてが図 4.2 で示したように 430 digit を超えていたので、ここに設定して実験する。STA の時間幅は信号の変化が特に大きい場所として、図 4.2 の赤線で示した範囲の 50 カウントを時間幅とした。LTA の時間幅と STA/LTA 比は、それぞれ暫定的に 200 カウントと 3072 とした。

4.1.3 結果

しきい値法ではドアを 5 回開閉して 4 回イベントを検出することができた。

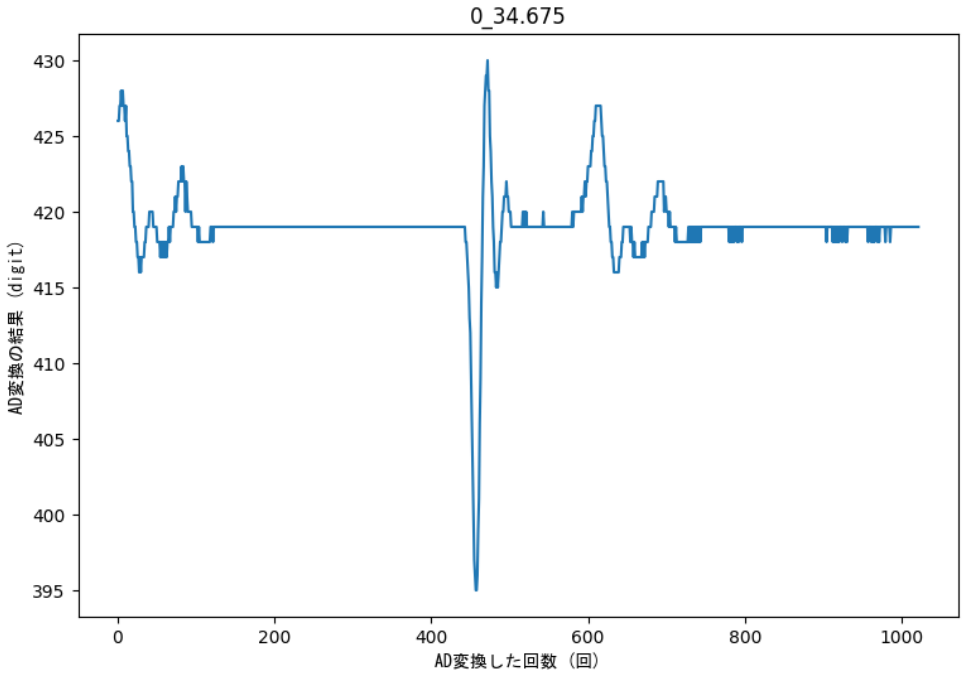

図 4.3　しきい値法イベント 1 回目

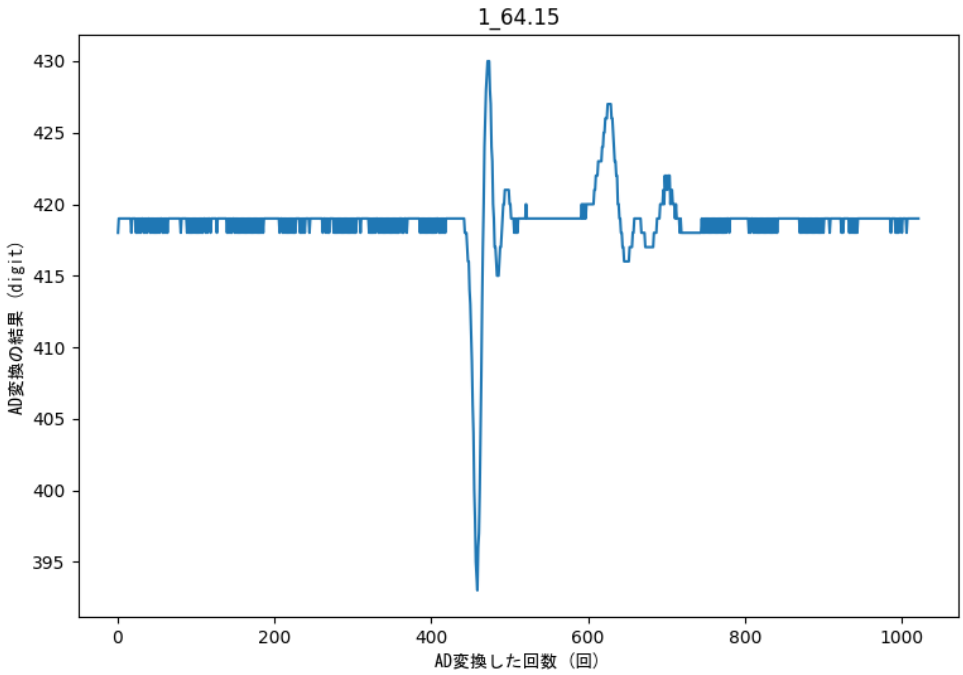

図 4.4　しきい値法イベント2回目

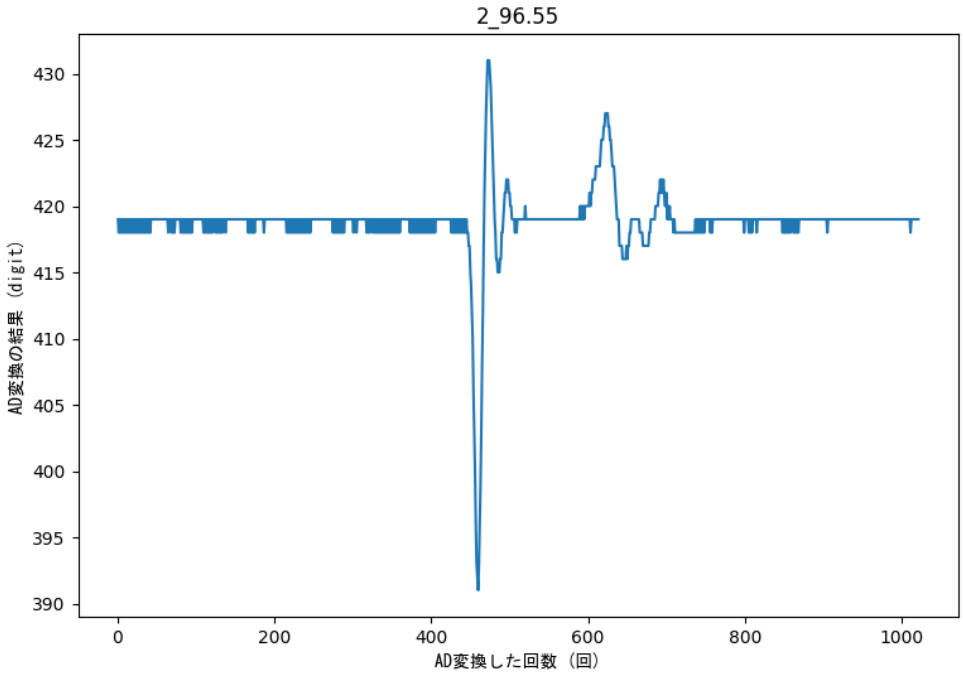

図 4.5 しきい値法イベント3回目

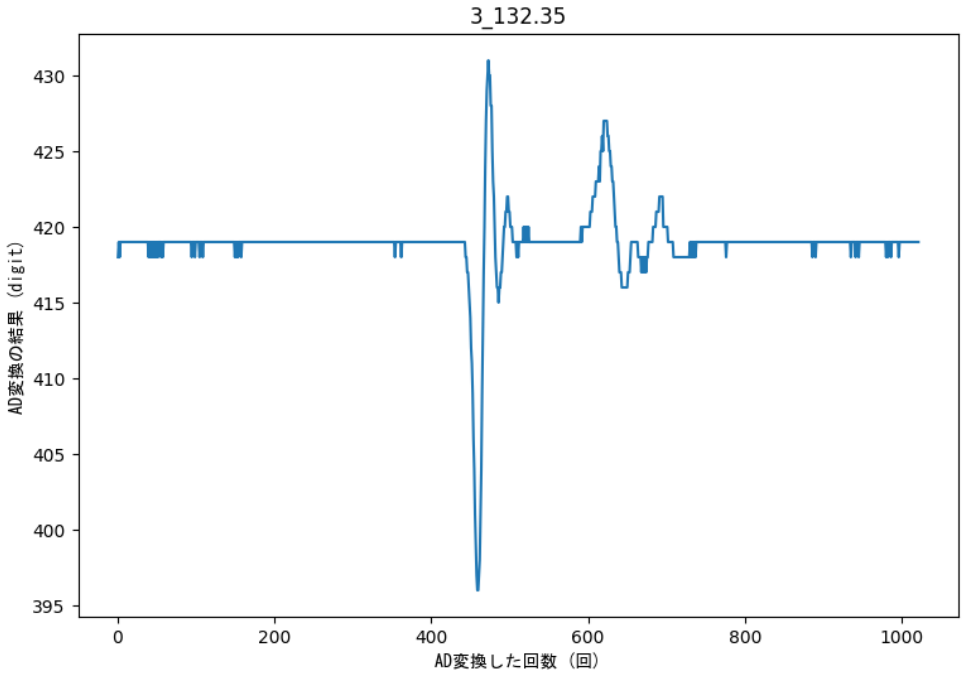

図 4.6　しきい値法イベント4回目

図 4.3～図 4.6 はしきい値法を用いてイベントの検出を実験した結果である。各グラフのタイトル部分の数字は、前の数字が何回目に記録されたものかを表していて、後ろの数字は起動してから何秒後にイベントが発生したかを表している。これらのグラフからわかることは室内ではドアの開閉による圧力変動に比べてノイズ成分は十分に小さいということがわかる。

次に、STA/LTA 法の結果を示すと同じ 5 回の扉の開閉に対して 3 回イベントを検出することができた。

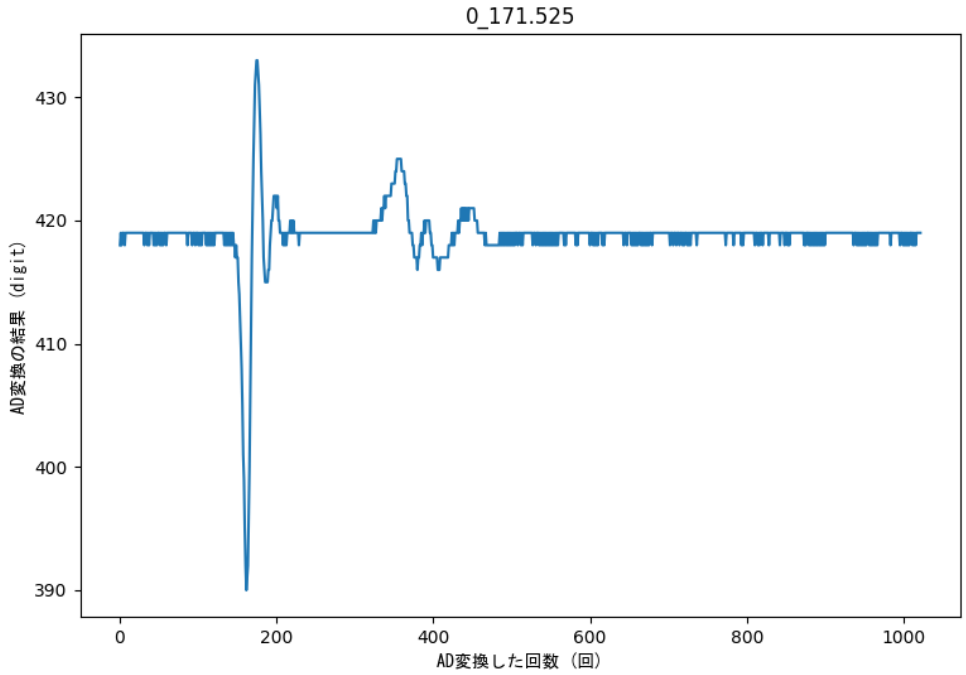

図 4.7　STA/LTA 法イベント1回目

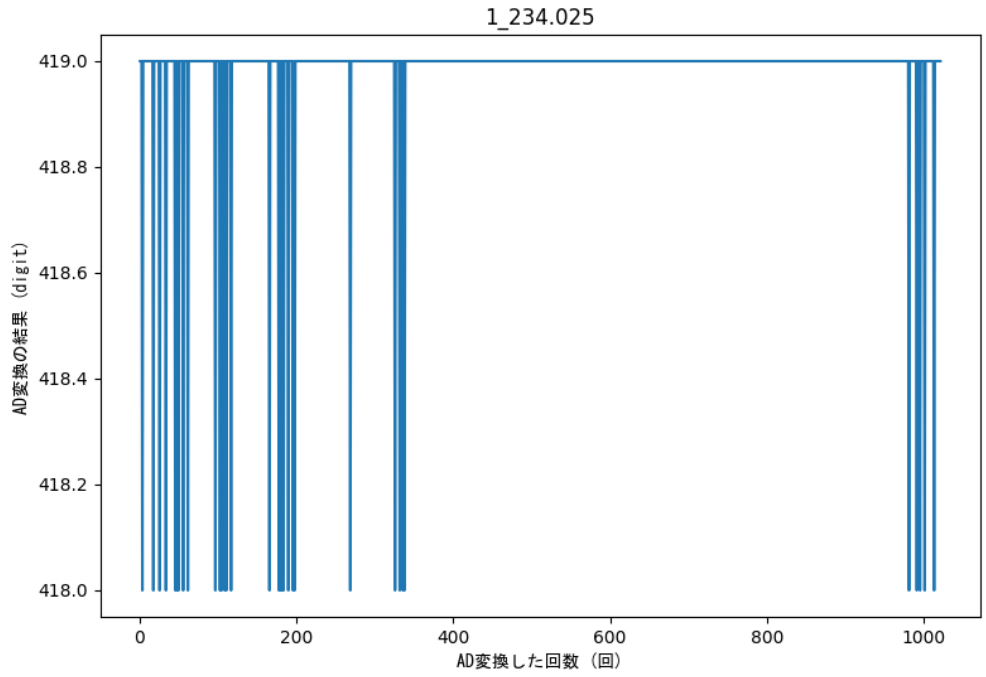

図 4.8　STA/LTA 法イベント 2 回目

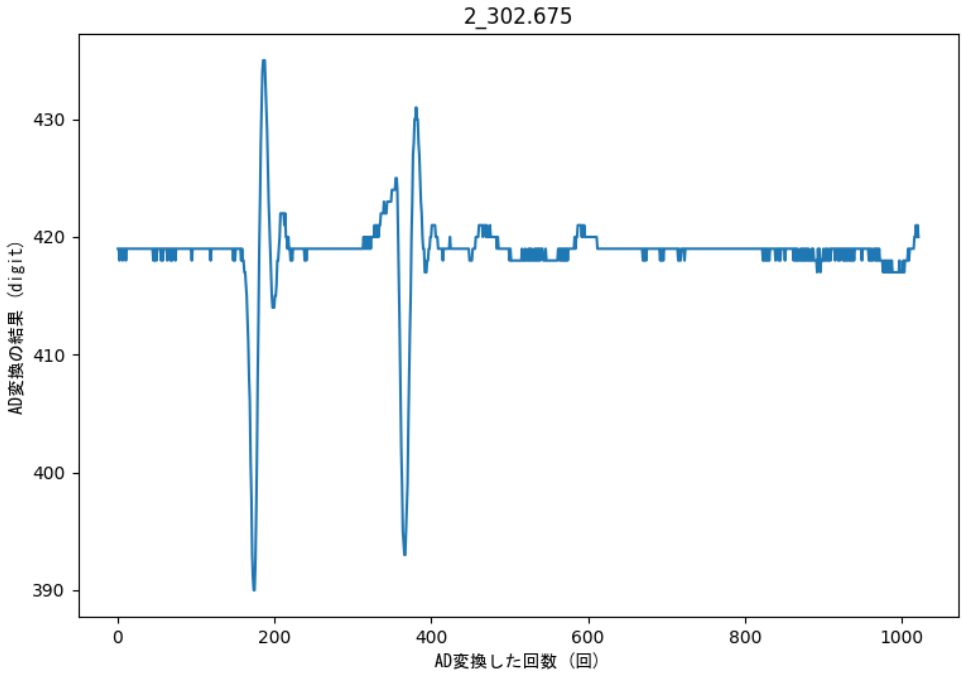

図 4.9　STA/LTA 法イベント 3 回目

図 4.7～図 4.9 は STA/LTA 法を用い、イベントの検出を実験した結果である。イベントとして検出された例は 3 回分あるが、図 4.8 は図 4.7、図 4.9 と比較して明らかに違いがみられるので、ドアの開閉としては 2 回分を検知することができた。

4.2ノイズに関する評価

インフラサウンドセンサは風の影響を受けやすく屋外に設置することを考えると風が最も大きいノイズ源になることが考えられる。よって風による影響を評価するために扇風機の風を当てノイズの影響を調べる。

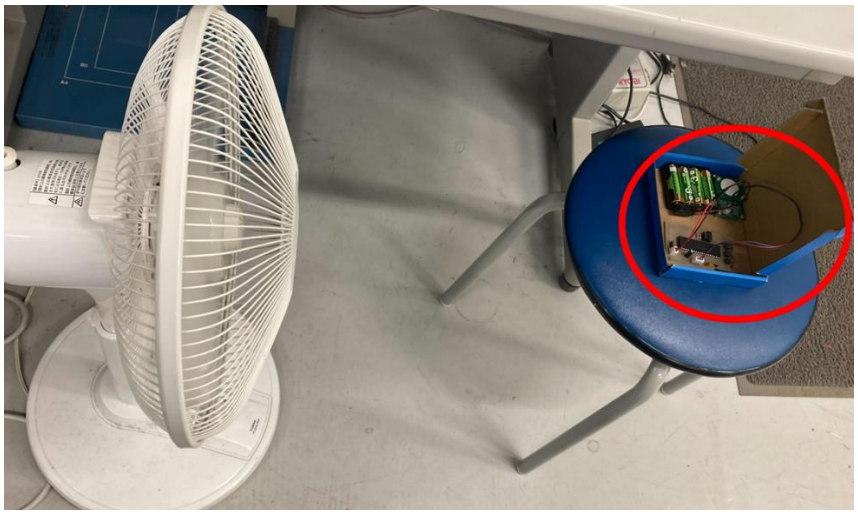

図 4.10　実験の様子

図 4.10 に示すように扇風機から約 30 ㎝離した椅子の上に置かれている赤線で囲った部分が観測器である。扇風機を首振りさせ間欠的に風が当たるようにした。これは実際の自然環境でも風が吹き続けるわけではないのでそれを模擬している。この手法でしきい値法を試験し、検出されたイベント一覧を図 4.11 に示す。

0_91.55.csv	2024/02/09 10:25	Microsoft Excel CS...
1_290.475.csv	2024/02/09 10:25	Microsoft Excel CS...
2_340.375.csv	2024/02/09 10:25	Microsoft Excel CS...
3_502.4.csv	2024/02/09 10:25	Microsoft Excel CS...
4_552.725.csv	2024/02/09 10:25	Microsoft Excel CS...
5_669.1.csv	2024/02/09 10:25	Microsoft Excel CS...
6_883.45.csv	2024/02/09 10:25	Microsoft Excel CS...
7_1261.5.csv	2024/02/09 10:25	Microsoft Excel CS...
8_1360.1.csv	2024/02/09 10:26	Microsoft Excel CS...
9_1458.225.csv	2024/02/09 10:26	Microsoft Excel CS...
10_1640.025.csv	2024/02/09 10:26	Microsoft Excel CS...
11_1673.125.csv	2024/02/09 10:26	Microsoft Excel CS...
12_1735.925.csv	2024/02/09 10:26	Microsoft Excel CS...
13_1804.85.csv	2024/02/09 10:26	Microsoft Excel CS...
14_1904.525.csv	2024/02/09 10:26	Microsoft Excel CS...
15_2036.075.csv	2024/02/09 10:27	Microsoft Excel CS...

図 4.11　しきい値法で検出されたイベント一覧

起動から 2036 秒で 16 個すべてに記録がされていた。

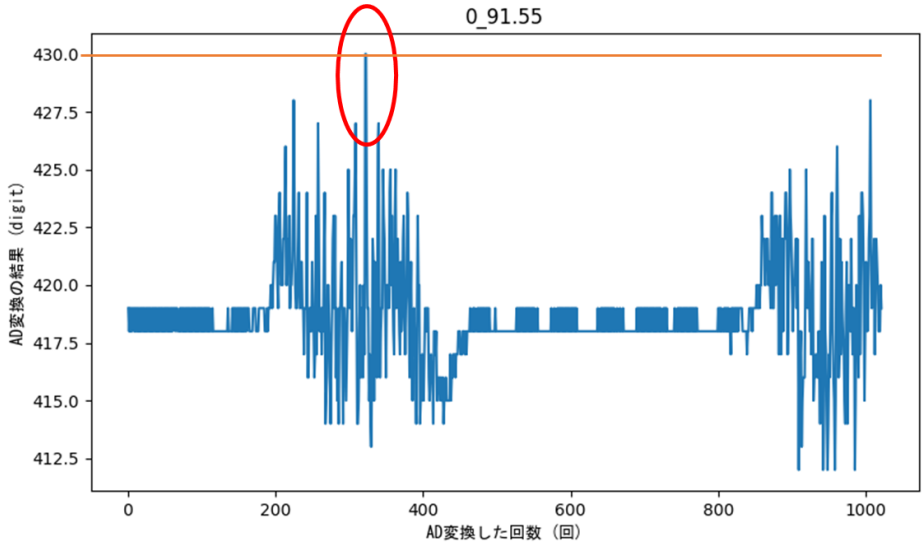

図 4.12　しきい値法ノイズ試験イベント 1 回目

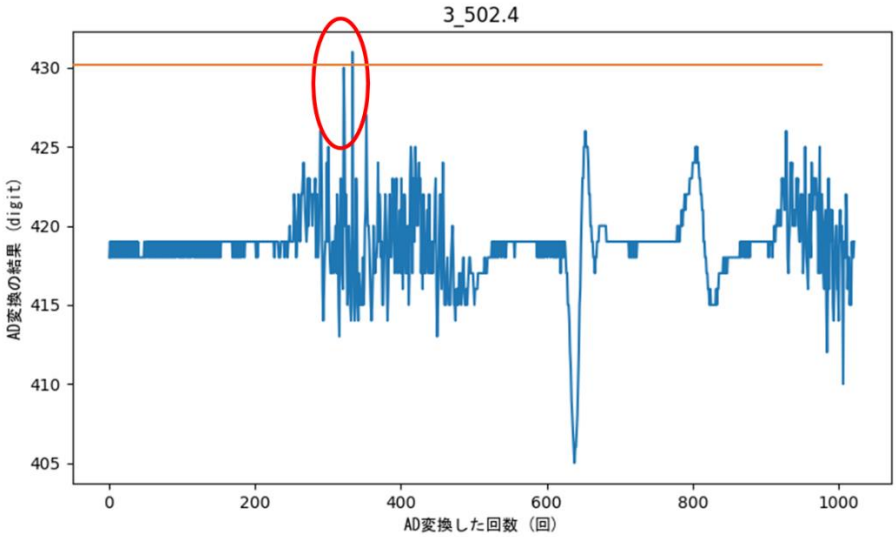

図 4.13　しきい値法ノイズ試験イベント 4 回目

図 4.12～図 4.13 はしきい値法を用いノイズの試験をしたものである。残りの記録も同じような波形が記録されていた。

0_116.25.csv	2024/02/09 9:10	Microsoft Excel CS...
1_387.275.csv	2024/02/09 9:11	Microsoft Excel CS...
2_499.125.csv	2024/02/09 9:11	Microsoft Excel CS...
3_632.225.csv	2024/02/09 9:11	Microsoft Excel CS...
4_655.15.csv	2024/02/09 9:11	Microsoft Excel CS...
5_782.4.csv	2024/02/09 9:21	Microsoft Excel CS...
6_899.975.csv	2024/02/09 9:11	Microsoft Excel CS...
7_947.15.csv	2024/02/09 9:11	Microsoft Excel CS...
8_998.05.csv	2024/02/09 9:11	Microsoft Excel CS...
9_1162.3.csv	2024/02/09 9:11	Microsoft Excel CS...
10_1192.725.csv	2024/02/09 9:10	Microsoft Excel CS...
11_1226.725.csv	2024/02/09 9:10	Microsoft Excel CS...
12_1322.95.csv	2024/02/09 9:10	Microsoft Excel CS...
13_1389.0.csv	2024/02/09 9:10	Microsoft Excel CS...
14_1587.975.csv	2024/02/09 9:10	Microsoft Excel CS...
15_1621.9.csv	2024/02/09 9:10	Microsoft Excel CS...

図 4.14　STA/LTA 法で検出されたイベント一覧

起動から 1621 秒で 16 個すべてに記録がされていた。

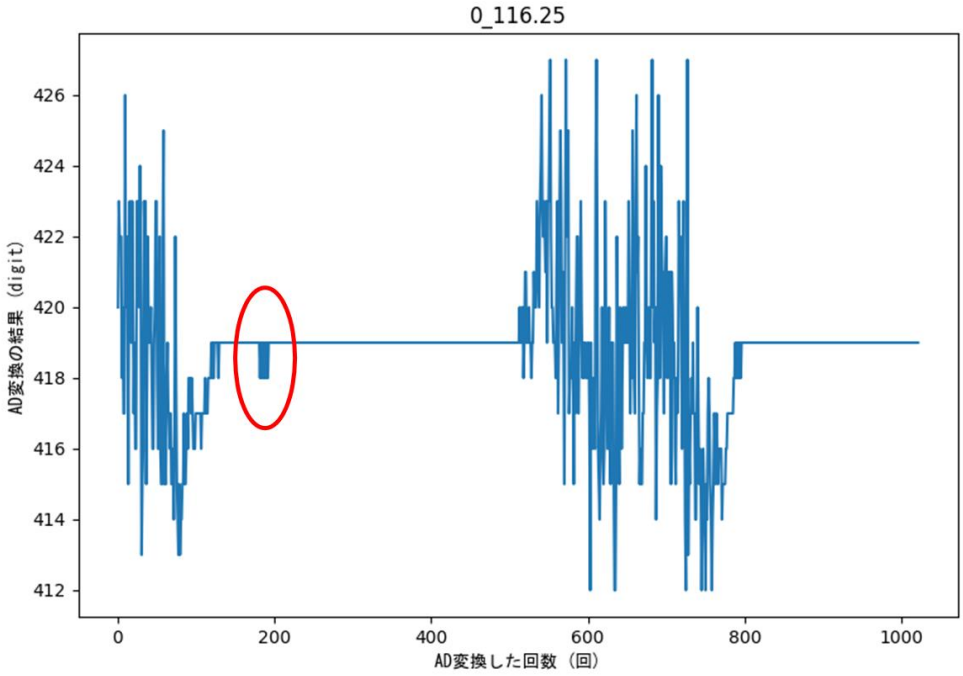

図 4.15　しきい値法ノイズ試験イベント 1 回目

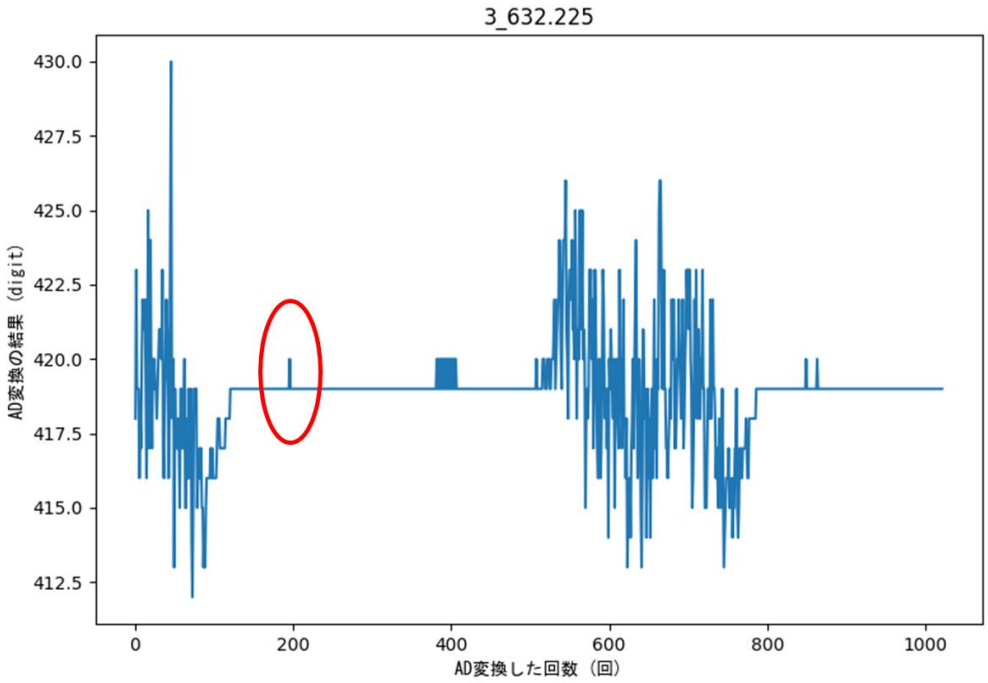

図 4.16　しきい値法ノイズ試験イベント 3 回目

図 4.15～図 4.16 は両社とも 200 回目の AD 変換付近で単発ノイズが入っている。

第5章　まとめ

本研究では、データ容量を小さくすることと、省電力を目的として、インフラサウンド（超低周波音）イベントの検出および自動記録のための装置をマイコンを用いて開発し、イベントの検出精度と、主に風ノイズへの耐性について評価した。マイコン上で実装したアルゴリズムによって、実験室のドア開閉による圧力変化により模擬したインフラサウンドイベントの検出をすることは成功したが、風ノイズの対策としては不十分だった。しきい値法と STA/LTA 法の比較ではイベント検出の精度はしきい値法のほうが良いという結果になったが、これは事前にどのような信号が来るか調べた上でしきい値を設定したため今回のような結果になったと考えられる。STA/LTA 法では目的としていた信号以外に単発のノイズを記録してしまったが、長時間のノイズをイベントとして記録することはなかった。よって単発のノイズと目的の信号とを区別することができれば、STA/LTA 法による自動検出の精度を向上させることができると考えられる。

謝辞

本研究を行うにあたり指導教員として適切な指導や添削をしてくださった、高知工科大学 システム工学群 山本真行教授、イベント検出のアルゴリズムについて指導してくださった高知工科大学 システム工学群 西川泰弘助教に感謝いたします。

参考文献

[1] 井上 祐一郎, ”多地点アレイ観測のための小型インフラサウンドデータロガーの開発”,令和 3 年度 高知工科大学 修士論文, 2021

[2] 吉田邦一 笹谷 努,” 強震観測における STA/LTA トリガー方式の問題点”, *北海道大学地球物理学研究報告*, **69**, 85-95, 2006

[3] Raspberry Pi Ltd.,”Raspberry Pi Pico Datasheet”, https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf, *令和 6 年 2 月 7 日 参照*

[4] Microchip Technology Inc,” ATmega4808-09-DataSheet-DS40002173C”, https://ww1.microchip.com/downloads/aemDocuments/documents/MCU08/ProductDocuments/DataSheets/ATmega4808-09-DataSheet-DS40002173C.pdf, *令和 6 年 2 月 7 日 参照。*