

2017 年度 卒業論文
小型ラドン検出器の開発

奈良女子大学 理学部
数物科学科 物理学コース 高エネルギー物理学研究室
港陽子 鈴木彩香

2018 年 4 月 9 日

目次

第 1 章 序論	3
1.1 ラドンについて	3
1.2 α 線の性質	5
1.3 原子核崩壊と放射平衡	6
第 2 章 検出原理	7
2.1 Si PIN フォトダイオード	7
2.2 フォトダイオードでの検出	8
2.3 静電捕集法	9
第 3 章 検出器と波高分析	10
3.1 実験装置の構成	10
3.2 波高分析システムの開発	11
3.2.1 プリアンプ回路の設計と製作	12
3.2.2 シェイパーアンプの設計と製作	13
3.2.3 微分回路	13
3.2.4 積分回路	15
3.2.5 比較回路	16
3.2.6 サンプルアンドホールド回路	16
3.2.7 シェイパーアンプ	17
3.3 波高分析とデータ収集	21
3.3.1 Arduino	21
3.3.2 Processing	23
第 4 章 実験	25
4.1 LED 実験	25
4.1.1 実験概要	25
4.2 ^{241}Am 線源を用いた α 線測定	28
4.2.1 実験目的	28
4.2.2 実験方法	28
4.2.3 結果と考察	29
4.3 ラドンガス測定	33
4.3.1 実験内容	33
4.3.2 実験結果	34
4.3.3 ベーテプロッホの式	35
4.3.4 半減期の測定	37
4.3.5 改善策	39

第5章	まとめ	41
5.1	まとめ	41
5.2	今後の課題	41
5.3	謝辞	41
第6章	付録	42
6.1	Arduino プログラム	42
6.2	Processing プログラム	45

第1章 序論

1.1 ラドンについて

ラドン (^{222}Rn) は無色無臭の放射性希ガス元素で、 α 崩壊する際に α 線を放出する。地中の岩石や土壌にはラドンの親核であるウラン (^{238}U) やトリウム (^{232}Th) などが多く含まれており、その崩壊によって生成されるラドンが自然界に広く存在している。日常生活における私たちの放射線被曝量の約半分がラドンガスによるものと言われているように、ラドンは非常に身近な放射性物質である。

α 崩壊系列はウラン系列、トリウム系列、アクチニウム系列、ネプツニウム系列の4種類ある。ただし、ネプツニウム系列は自然界にほとんど存在しない。一般的にウラン系列中の ^{222}Rn はラドン、トリウム系列中の ^{220}Rn はトロン、 ^{219}Rn はアクチノンと呼ばれており、天然存在比は 90:9:1 である。(図 1.1-1.3) 中でも今回は一番自然界に多く存在するウラン系列に着目する。ウラン系列に属するラドンは 3.8 日という短い半減期で 5.49MeV のエネルギーの α 線を放出し ^{218}Po に壊変する。さらに ^{218}Po は半減期 3 分で 6MeV の α 線を放出し ^{214}Pb に壊変する。また ^{214}Po は半減期 1.6×10^{-4} 秒で 7.7MeV の α 線を放出し ^{210}Pb に壊変する。今回は特に ^{218}Po と ^{214}Po が崩壊する際に放出する α 線のエネルギーを測定することで間接的にラドンを検出する。

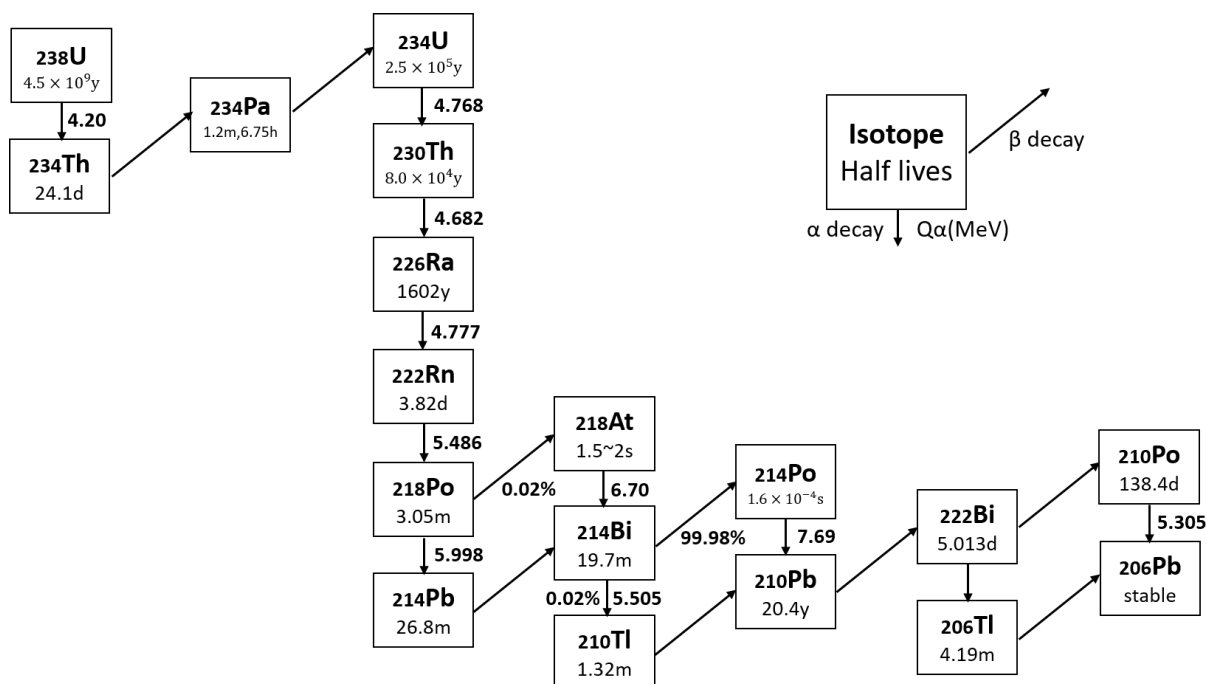


図 1.1 ウラン系列

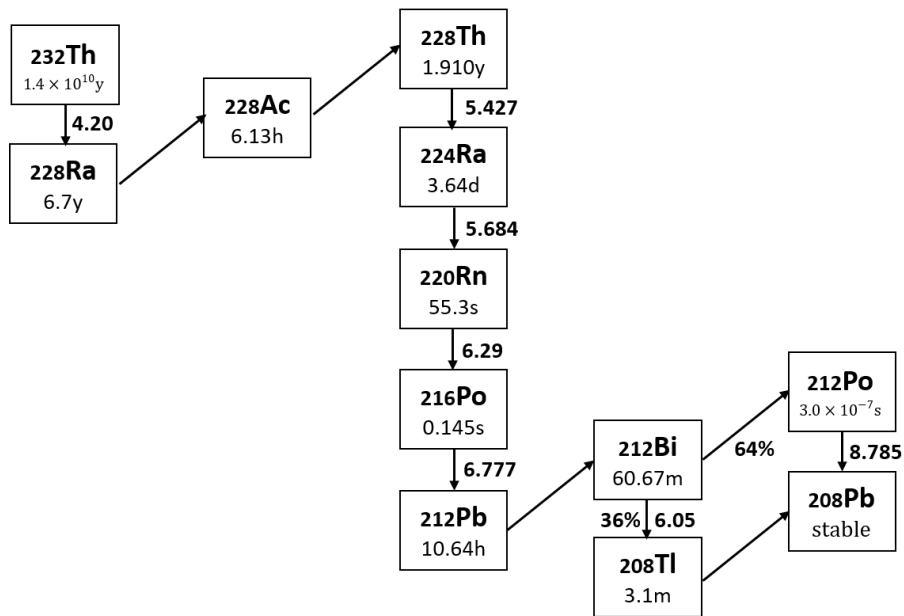


図 1.2 トリウム系列

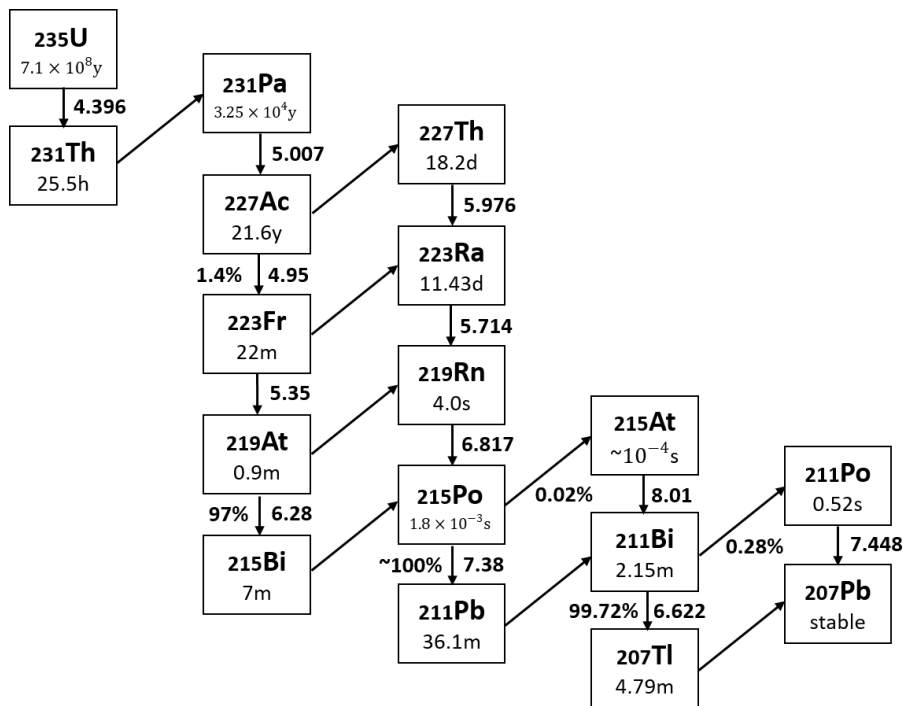


図 1.3 アクチニウム系列

1.2 α 線の性質

α 線は α 粒子の流れのことである。 α 粒子とは、陽子2個と中性子2個からなる ${}^4\text{He}$ の原子核である。 α 粒子は2価の陽イオンで電離作用が大きいため透過力が弱く、大気中では2~3cmで止まってしまう。図1.4に示すように、 α 線は紙などで簡単に遮蔽することができる。

放射性元素が α 粒子を放出して崩壊する現象を α 崩壊または α 壊変という。自然に存在するそのような放射性元素はサマリウム(${}_{62}\text{Sm}$)より重い元素に限られており、放出される α 粒子のエネルギーは最大9MeV程度である。これよりエネルギーの高い α 粒子は加速器によって得られ、原子核反応を起こすのに利用される。このような α 粒子の流れは一般的に α ビームと呼ばれる。

放射性核種は不安定なため、 α 線や β 線や γ 線を放出して自然に崩壊していく。 α 崩壊により、 (Z,A) の親核は $(Z-2,A-4)$ の娘核に壊変する。その娘核も不安定であれば、さらに α 線または β 線を放出して壊変する場合もある。

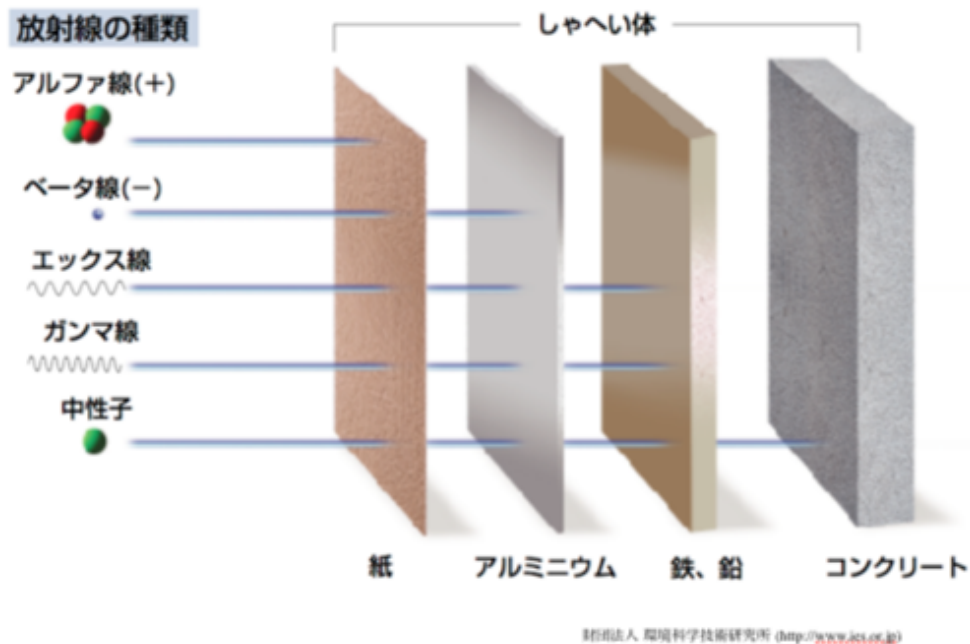


図 1.4 放射線の遮蔽

1.3 原子核崩壊と放射平衡

原子核の崩壊現象は、確率的な現象で互いに独立した事象であるとする以下のような関係が成り立つ。

$$-dN = N\lambda dt \quad (1.1)$$

すなわち、ある1個の原子が与えられると、それが時間間隔 dt のあいだに崩壊する確率は λdt である。ここで、 λ は崩壊定数といい、寿命 τ 、半減期 $T_{1/2}$ を用いて

$$T_{1/2} = \frac{\ln 2}{\lambda} \quad (1.2)$$

と表せるので、崩壊定数はある与えられた物質の崩壊時間を特徴づけるものである。

以上を基本として、単一の放射性物質の崩壊を考える。時刻 $t = 0$ のとき、 $N(0)$ 個の原子を持っていたとすると、式 (1.1) の解は

$$N(t) = N(0)e^{-\lambda t} \quad (1.3)$$

となる。

次に、放射性物質の崩壊が崩壊系列をなす場合を考える。それぞれの原子が時刻 $t = 0$ のとき、 $N_1(0), N_2(0), N_3(0) \dots$ 個あったとして、ある時刻 t におけるそれぞれの原子数 $N_1(t), N_2(t), N_3(t) \dots$ を求める。ある物質 X_1 が X_2 に崩壊するときは基本に則って

$$dN_1 = -N_1\lambda_1 dt \quad (1.4)$$

となる。続いて X_2 が X_3 に崩壊するときは、 X_1 が1個崩壊するごとに X_2 が1個つくられるので

$$\frac{dN_2}{dt} = \lambda_1 N_1 - \lambda_2 N_2 \quad (1.5)$$

と表せる。ここで、 λ_1 と λ_2 はそれぞれの核種の崩壊定数である。 X_3 以降についても

$$\frac{dN_i}{dt} = \lambda_{i-1} N_{i-1} - \lambda_i N_i \quad (1.6)$$

となる。これらを解けば $N_1(t), N_2(t), N_3(t) \dots$ の時間変化がわかる。

一連の崩壊過程の連鎖において、半減期の長い元素がある場合には以下の関係が成り立つ。これを放射平衡という。 $\lambda_1 \ll \lambda_2, \lambda_3$ のとき

$$N_1(t) = N_0 e^{-\lambda_1 t} \quad (1.7)$$

$$N_2(t) \simeq \frac{\lambda_1 N_0}{\lambda_2 - \lambda_1} e^{-\lambda_1 t} \quad (1.8)$$

$$N_3(t) \simeq N_0 \lambda_1 \lambda_2 \frac{e^{-\lambda_1 t}}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)} \quad (1.9)$$

より

$$\frac{N_2(t)}{N_1(t)} \simeq \frac{\lambda_1}{\lambda_2 - \lambda_1} \sim \frac{\lambda_1}{\lambda_2} \quad (1.10)$$

$$\frac{N_3(t)}{N_1(t)} \simeq \frac{\lambda_1 \lambda_2}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)} \sim \frac{\lambda_1}{\lambda_3} \quad (1.11)$$

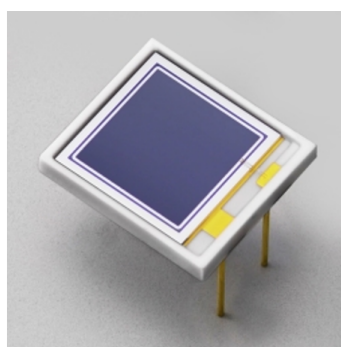
したがって、核種比は一定かつ崩壊定数 λ に逆比例する。

本研究ではこの性質を使って $t=0$ でのラドンの個数を求めラドン濃度を知ることが目的である。

第 2 章 検出原理

2.1 Si PIN フォトダイオード

フォトダイオードとは、半導体の PN 接合部に光を照射すると電流を発生する受光素子のことである。今回使用した Si PIN フォトダイオードは、Hamamatsu S3590-09 である（図 2.1）。仕様を表 2.1 に示す。



浜松ホトニクス Hamamatsu S3590-09
<https://www.hamamatsu.com/jp/ja/product/category/3100/4001/4103/S3590-09/index.html>

図 2.1 Si PIN フォトダイオード

表 2.1 Hamamatsu S3590-09 の仕様

受光面サイズ	10 × 10 mm
素子数	1
パッケージ	セラミック
パッケージカテゴリ	未封入
逆電圧 max.	100V
感度波長範囲	340～1100 nm
最大感度波長 typ.	960 nm
受光感度 typ.	0.66 A/W
暗電流 max.	6000 pA
遮断周波数 typ.	40 MHz
端子間容量 typ.	40 pF

PIN フォトダイオードは、正孔のある p 型半導体、電子のある n 型半導体、キャリアの存在しない真性半導体で構成されている。PIN は p-intrinsic-n の略である。PIN フォトダイオードに順バイアス、つまり n 型半導体側が負、p 型半導体側が正の電圧をかけると電子が陽極に、正孔が陰極に引きつけられ電流が流れる（図 2.2）。反対に、逆バイアス、つまり n 型半導体側が正、p 型半導体側が負の電圧をかけると電流が流れず、電子が陽極に、正孔が陰極に引きつけられることで空乏層が広がる。その空乏層に α 線が入射すると、電離作用により電子・正孔対ができる。このとき α 線はエネルギー損失をし、エネルギーがなくなるまで電子・正孔対

を繰り返す。エネルギーがなくなった α 線は空乏層中で止まる (図 2.3)。そして発生した電子は陽極へ、正孔は陰極へ向かい電気信号として現れる。

今回は PIN フォトダイオードに逆バイアスをかけて、ラドンの娘核が崩壊時に放出する α 線を検出する。

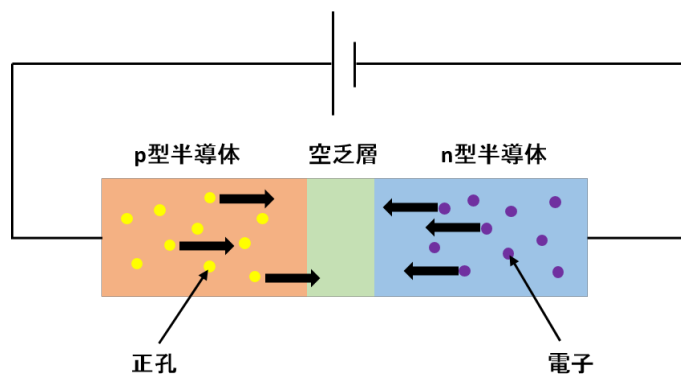


図 2.2 フォトダイオードに順バイアスかけた状態

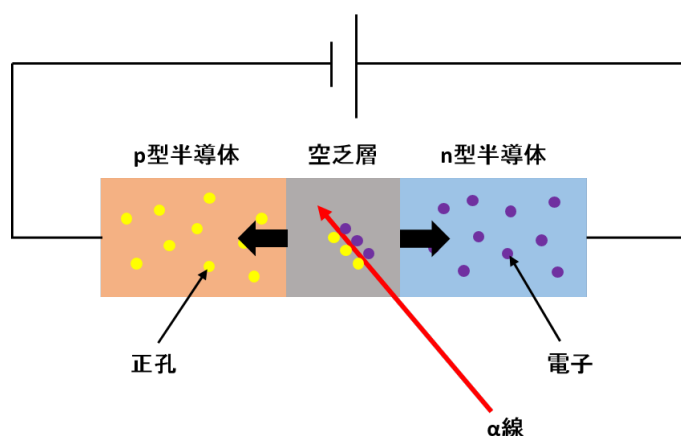


図 2.3 フォトダイオードに逆バイアスかけた状態

2.2 フォトダイオードでの検出

1つの電子正孔対を作るのに必要なエネルギーは3.6 (eV) である。検出する α 線を Am だとすると、Am から崩壊する α 線のエネルギーは 5.4(MeV) であるので

$$5.4(\text{MeV})/3.6(\text{eV}) = 1.5 \times 10^6 \text{ 個}$$

の電子対ができる。これは、

$$1.5 \times 10^6 \times 1.6 \times 10^{-19}(\text{C}) = 2.4 \times 10^{-13}(\text{C})$$

の電気量に相当する。

この電気量を、電流に換算する。

$I = dQ/dt$ より、 $t = 1 \mu (s)$ なら、流れる電流は、

$$I = 2.4 \times 10^{-7}(A)$$

である。この数値が α 線がフォトダイオードに入ったときに流れる電流であり、とても微量であることがわかる。

2.3 静電捕集法

α 線は大気中では数 cm で止まってしまうため、検出することは簡単ではない。そこで、静電捕集法を用いて α 線を検出する。図 2.4 にラドン検出器内での静電捕集法の原理を示す。

密閉した容器の底に岩石や土などの被測定物を置くと、時間の経過とともに被測定物から発生したラドンが容器内に充満していく。ラドンの娘核である ^{218}Po は陽イオンとなって容器内に存在している。ここでフォトダイオードの表面に負の高電圧をかけ容器壁面をグランドにし、検出器内に静電場を作ることで、 ^{218}Po はフォトダイオード表面に引き寄せられ効率よく捕集することができる。フォトダイオード表面に集められた ^{218}Po は α 崩壊するので、このとき放出する α 線のエネルギーを測定する。さらに同じ崩壊系列中の ^{214}Po も α 崩壊するため同様に測定を行う。こうして α 線を識別測定することで、間接的にラドンを検出する。

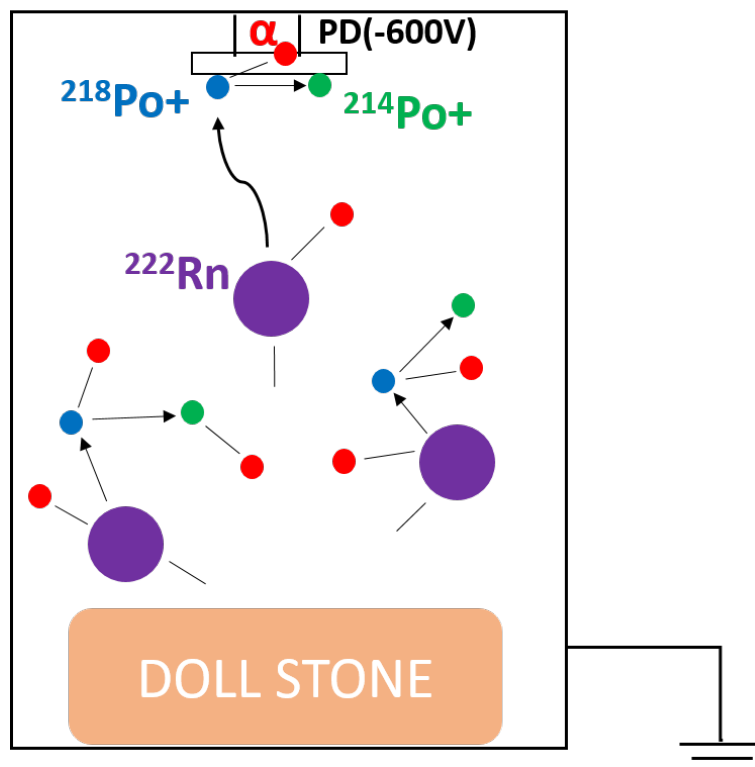


図 2.4 静電捕集法

第 3 章 検出器と波高分析

3.1 実験装置の構成

ラドン検出器は高さ 13.0cm、外径 5.5cm の円筒形であり、ステンレスの円筒形の管でできている。ラドン検出器内にはフォトダイオードをとりつける。そして、フォトダイオードに α 線が入射したときに流れる電気信号を波高分析システム (1.2) に出力するための回路の製作を行った。この回路は、フォトダイオードに約 60V の逆バイアスをかけ、かつ静電捕集のためにフォトダイオード表面に負の高電圧をかけることのできる回路の設計、製作を行った。図 3.1 は検出器の外形である。



図 3.1 ラドン検出器

3.2 波高分析システムの開発

今回の実験で使うラドン検出器の流れを説明する。まず、プリアンプからの出力信号が入ると、シェイパーアンプの中の比較回路が Arduino へトリガー信号を送る (図 3.2 の ①)。トリガー信号を受けた Arduino はゲート信号をサンプル&ホールド回路へ送る (図 3.2 の ②)。サンプル&ホールド回路はゲート信号がきたタイミングでの入力信号の電圧を保持する (図 3.2 の ③)。Arduino でその電圧を ADC で読みだし PC へ送られる (図 3.2 の ④)。

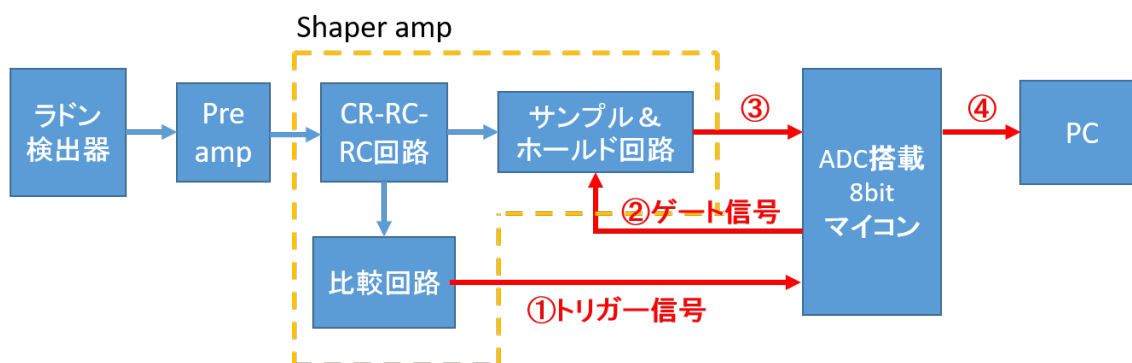


図 3.2 Arduino が行う波高分析の手順

1.2 章より、 α 線がフォトダイオードを通過したときに作られる電気量は、 $2.4 \times 10^{-13}(C)$ であり、とても小さい信号であることがわかった。そのため、増幅しないと観測は難しい。信号を観測するために、今回はプリアンプ、シェイパーアンプの二つの増幅整形回路を用いた。さらに、Arduino (3.14) を用いて波高分析環境を整えた。この章ではそれぞれの開発について説明する。

3.2.1 プリアンプ回路の設計と製作

まず、Am 線源 (5.4MeV) を用いて α 線がフォトダイオードに入射した場合を考える。前述したとおり、 α 線の信号を計測するには小さすぎる。そのため、電荷積分型増幅器を今回用いた。電荷積分型増幅器とは、入力から流れてくる電荷を電圧に変換して出力するというものである。

今、フォトダイオードを通過したときに作られる電気量は、 $Q_{in} = 2.4 \times 10^{-13}(C)$ であるので、フォトダイオードから出力される電圧 V_{in} は、

$$V_{in} = 2.4 \times 10^{-5}(V) = 0.024(mV)$$

であるので、電荷積分型増幅器後の電圧 V_{out} は

$$\begin{aligned} V_{out} &= Q_{in}/C_F \\ &= 2.4 \times 10^{-13}/10^{-11} \\ &= 24(mV) \end{aligned}$$

となる。図 3.3 にフォトダイオード後の電荷積分型増幅器を表す。今回入力電流が微弱なため、入力バイアスが小さい pA のオペアンプ OPA134 を使用した。OPA134 は $\pm 2.5V \sim \pm 18V$ の電源で動作し、FET 入力段を備えている。図 3.4 がプリアンプの回路図である。

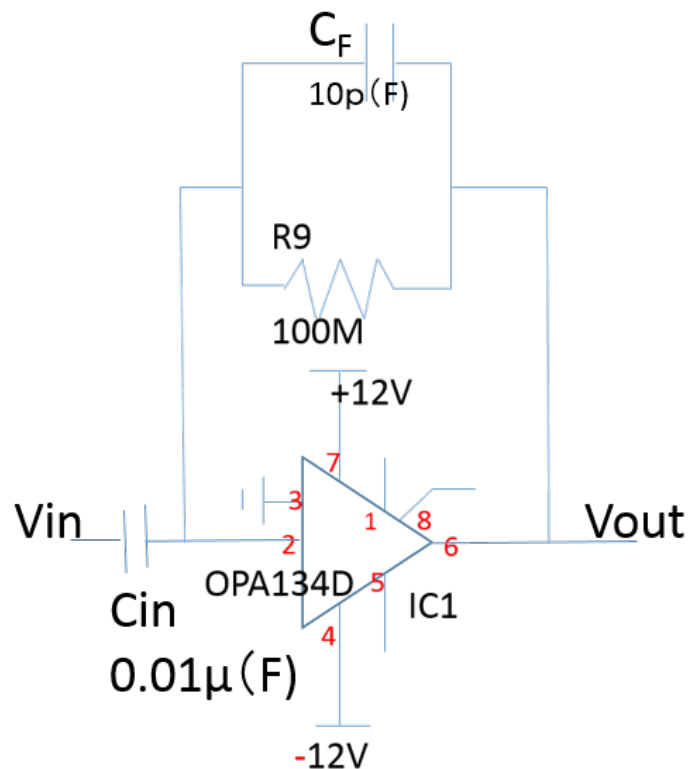


図 3.3 フォトダイオード後の電荷積分型増幅器

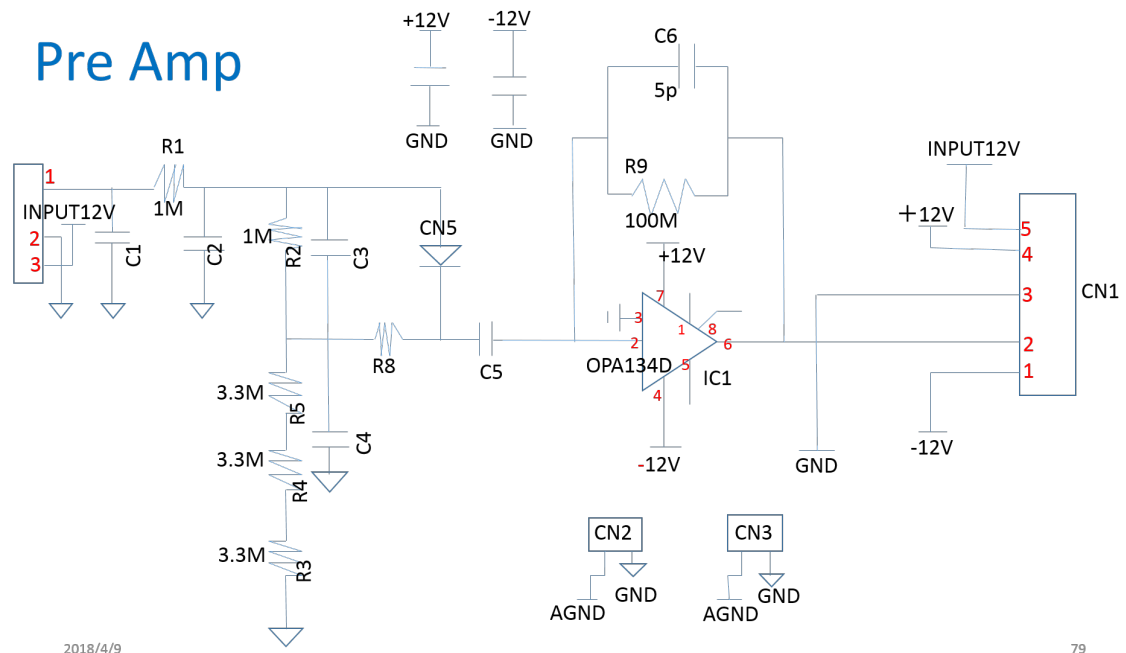


図 3.4 プリアンプ

3.2.2 シェイパーアンプの設計と製作

プリアンプからの出力は、計算したように、24mV で波高分析を行うにはまだまだ微小である。そのため、波形整形しながら増幅するためのシェイパーアンプ回路の設計・製作を行った。シェイパーアンプでは、波形整形と増幅のための微積分回路及び増幅回路に加え、trigger 信号を出すための比較回路と入力信号の電圧値を一時保持させるサンプル&ホールド回路の機能について簡単に説明したうえで実際に設計・製作したシェイパーアンプについて説明する。

3.2.3 微分回路

入力の時間微分(変化、傾き)を出力する回路微分回路といい、図 3.5 に示すように、コンデンサーと抵抗で構成され、一般的に CR 回路と呼ばれている。入力信号が大きくなった時には正の方向に変化成分が出力され、入力信号が小さくなったときには負の方向に変化成分が出力される。

図 3.6 は回路シミュレータ SPICE で微分回路をシミュレーションした結果である。ここでの入力信号は短形パルスとした(図 3.6 の緑線)。短形パルスを入力して出力した波形が図 3.6 中の赤線の結果である。また、時定数 τ は $\tau = CR$ と書き表されるため、図 3.5 の場合 $C = 0.01F$ 、 $R = 2k\Omega$ より、 $\tau = 20s$ ということになる。微分回路は、変化の小さい低周波成分をあまり通さず、変化の大きい高周波成分をよく通す特性があるため、ハイパスフィルタとして機能する。

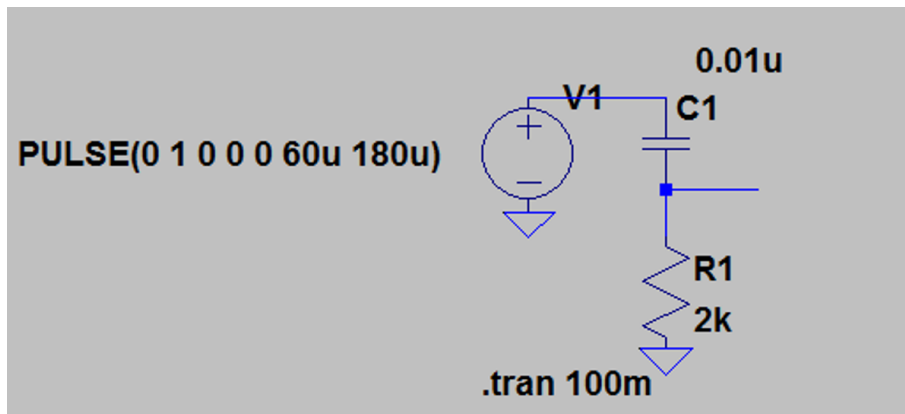


図 3.5 微分回路

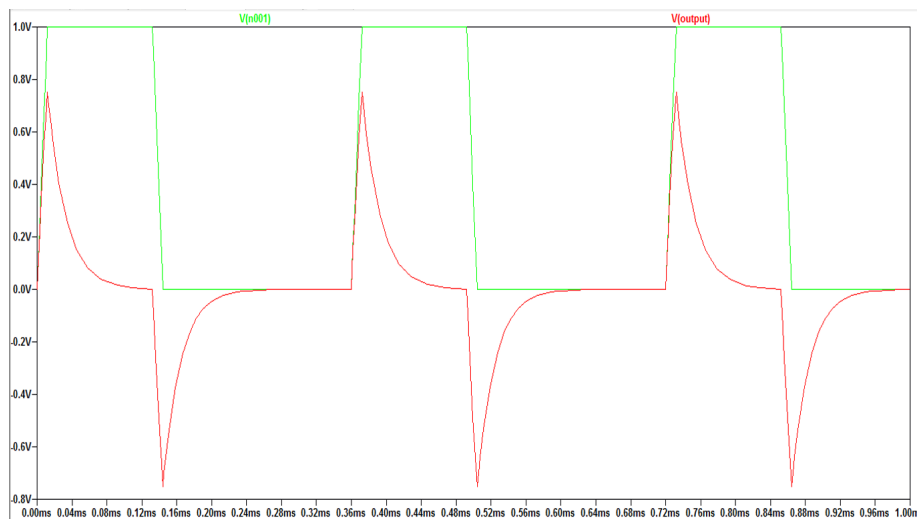


図 3.6 SPICE での微分回路シミュレーション結果

3.2.4 積分回路

入力の時間積分を出力する回路を積分回路という。図 3.7 のように抵抗とコンデンサーで構成される。この回路は一般的に RC 回路と呼ばれている。図 3.8 は SPICE で積分回路をシミュレーションした結果である。ここでの入力信号は矩形パルスとした (図 3.8 緑)。この時の出力波形が図 3.8 の赤線で示された結果である。また、時定数は図 3.7 の場合、微分回路と同様 $\tau = 20s$ である。積分回路は、微分回路とは逆で、変化の小さい低周波成分をよく通し、変化の大きい高周波成分をあまり通さない特性があるため、ローパスフィルターとして機能する。

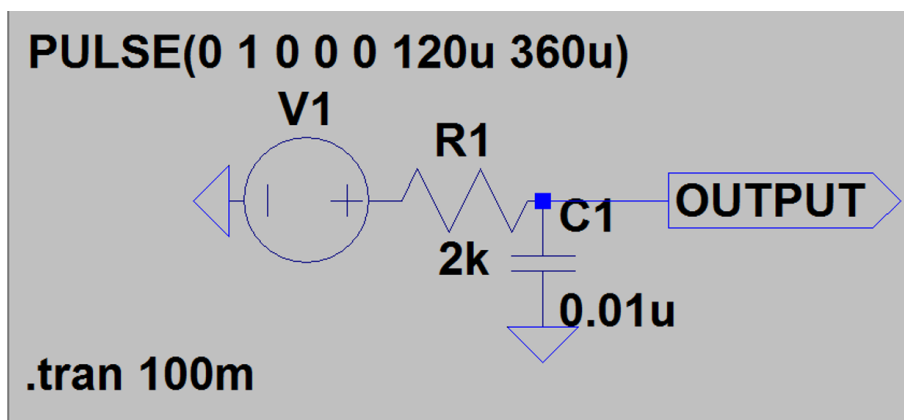


図 3.7 積分回路

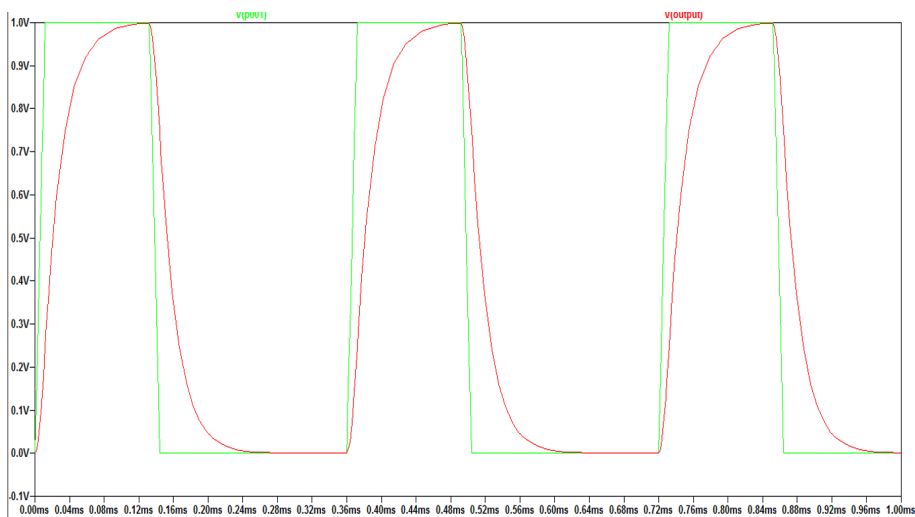


図 3.8 SPICE での積分回路シミュレーション結果

3.2.5 比較回路

比較回路は二つの入力信号に対してどちらの信号が大きいか小さいかを示す回路である。オペアンプの非反転入力端子に入力電圧をかけ、反転入力端子に何らかの比較対象となる電圧(しきい値)を与える。入力電圧がしきい値を超えた場合、正の最大電圧を出力する。また、しきい値を下回ると負の最大電圧を出力する。つまり、比較回路の出力はデジタル的な信号となる。このような動きをする理由として、オペアンプの増幅度は理想的には無限大であり、電源電圧まで一気に立ち上がる、または、立ち下がることになるからである。ただし、出力電圧はオペアンプの供給電源やスルーレートに制限される。比較回路はセンサーのオンオフやデジタル回路的に信号を直したいときなどに用いられる。

3.2.6 サンプルアンドホールド回路

サンプルアンドホールド回路とは、刻々と変化するある一瞬の信号のレベルを外からのトリガーのタイミングに合わせて保持する回路のことである。例として、サンプルアンドホールド回路に、Arduino から 50 % の PWM 波を入力した (V_{in}) 時のことを考える。PWM 波とは、一定電圧の入力から、パルス列のオンとオフの一定周期を作り、オンの時間幅を変化させる電力制御方式のことである。この時、サンプルアンドホールド回路の ON,OFF は、スイッチで制御した。 V_{sh} が 5.0V の時、スイッチが ON の状態で、0V のとき、スイッチは OFF の状態である。

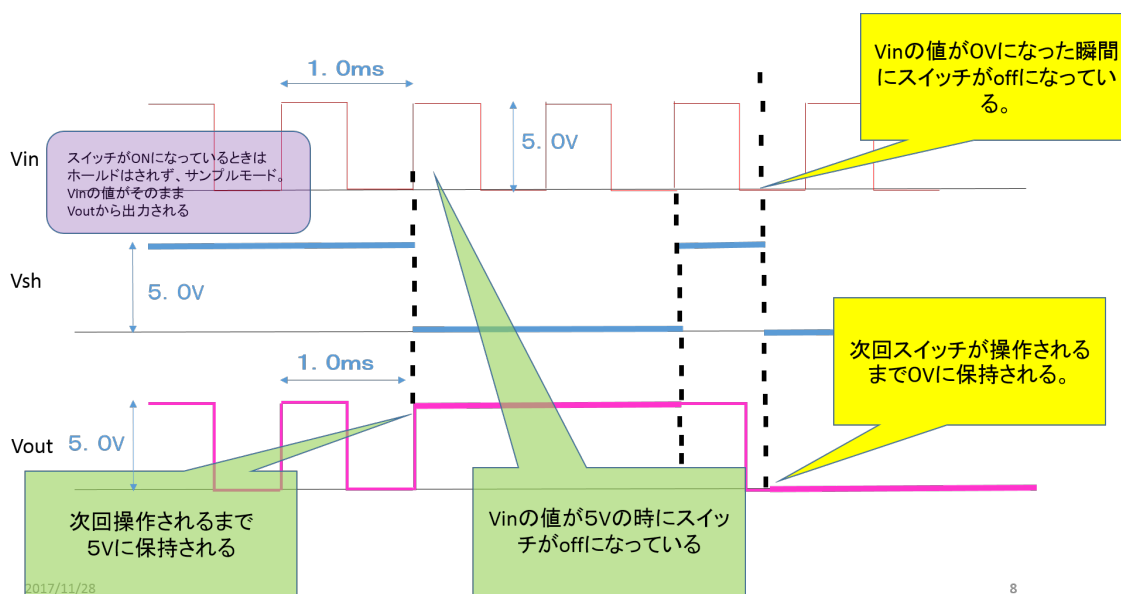


図 3.9 サンプルアンドホールド回路解説

図 3.9 を見ると、スイッチが OFF になった瞬間の V_{in} の値を保持して、 V_{out} として出力していることがわかる。実際の ON/OFF の切り替えは、次章で説明する、Arduino に制御させた。

3.2.7 シェイパーアンプ

シェイパーアンプの回路を図 3.10 のように設計・製作した。

時定数は $20 \mu\text{s}$ であり増幅率は、可変抵抗 R5、R6 を調整することで変えることができる。プリアンプからの立ち上がり時間が速く、立下り時間が遅い信号を微分回路、積分回路、オペアンプによる増幅回路の組み合わせを用いて波形整形、増幅を行う。また微分回路、積分回路を組み合わせることは波形整形だけでなく、フィルタ回路としてノイズを低減させる役割も果たしている。

今回は CR-RC-CR 回路を採用した。実際に製作した CR-RC-CR 回路の動作確認を Am 線源 (5.4MeV) を用いて行った。ラドン検出器からの信号を既存のプリアンプを用いて増幅し、その出力信号をシェイパーアンプの CR-RC-CR 回路の入力信号とした。図 3.11 の黄色線は 1 段目の微分回路を通して、増幅された出力波形、水色線は 2 段目の積分回路後に、反転増幅された出力波形をオシロスコープで確認した図である。

同様に図 3.12 の黄色線は 3 段目の微分回路後に反転増幅された波形を示している。、バイポーラ型の波形になっており、CR-RC-CR 回路が機能していることが確認できた。増幅回路用オペアンプとしては今回小型化のために、1 つの素子に 2 つのオペアンプが搭載されているデュアル・タイプの OPA2134 は、プリアンプで使用した OPA134 と同じシリーズで、FET 入力段を採用したオペアンプである。 $\pm 2.5\text{V} \sim \pm 18\text{V}$ の電源で作用し、入力カスコード回路により優れた同窓モード除去を実現しているうえ、広い入力電圧範囲にわたり低入力バイアス電流を維持して、ゆがみを最小限に抑えている。

次に trigger 信号を出すための回路として、比較回路を採用した。trigger 信号とは、フォトダイオードが一定以上の電圧を観測したときに出る信号のことである。しきい値は R 11 の可変抵抗で調整でき、今回は約 3.0V になるように調整した。3.0V を超える信号が比較回路に入ると、オペアンプの電源電圧の最大値まで立ち上がり、3.0V を下回ると最小値まで落ち下がる。比較回路の回路図を図 3.13 に示す。

図 3.14 の緑線は比較回路の入力信号 (V)、ピンク線は比較回路の出力信号をオシロスコープで見た波形である。しきい値 3.0V を黄色の補助線として表した。

図 3.13 より、比較回路で、V としきい値が比較され、V がしきい値を上回ったところで、trigger 信号が出ていることがわかる。オペアンプの出力が最大値となると、エミッタ設置トランジスタ回路に電流が流れ、5V から 0V まで一気に立下り、それが trigger 信号となって Arduino のデジタル入力ピンに送られる。抵抗 R11 はプルアップ抵抗であり、5V は Arduino から供給している。

最後に波高分析用の信号を出力する回路としてサンプル&ホールド回路を使用した。LF398 は $\pm 5\text{V} \sim \pm 18\text{V}$ の電源で動作し低オフセット電圧及び広い帯域幅を達成するためにバイポーラ入力段を用いているため、1MHz オペアンプのフィードバック・ループ内に入れることができる。また、 $10^{10}\Omega$ の入力インピーダンスにより、制度を低下させることなく、ソース・インピーダンスを高くすることができる。サンプル&ホールド回路からの出力は、回路の設計上最大 12V となる可能性があるが、Arduino のアナログ入力電圧の最大値は 5V であるため、間に抵抗を挟むことで Arduino に入る最大電圧を 5V 以下に調整した。

図 3.14 のピンク線が trigger 信号、水色線が GATE 信号、緑線がサンプル&ホールド回路の出力波形を示している。

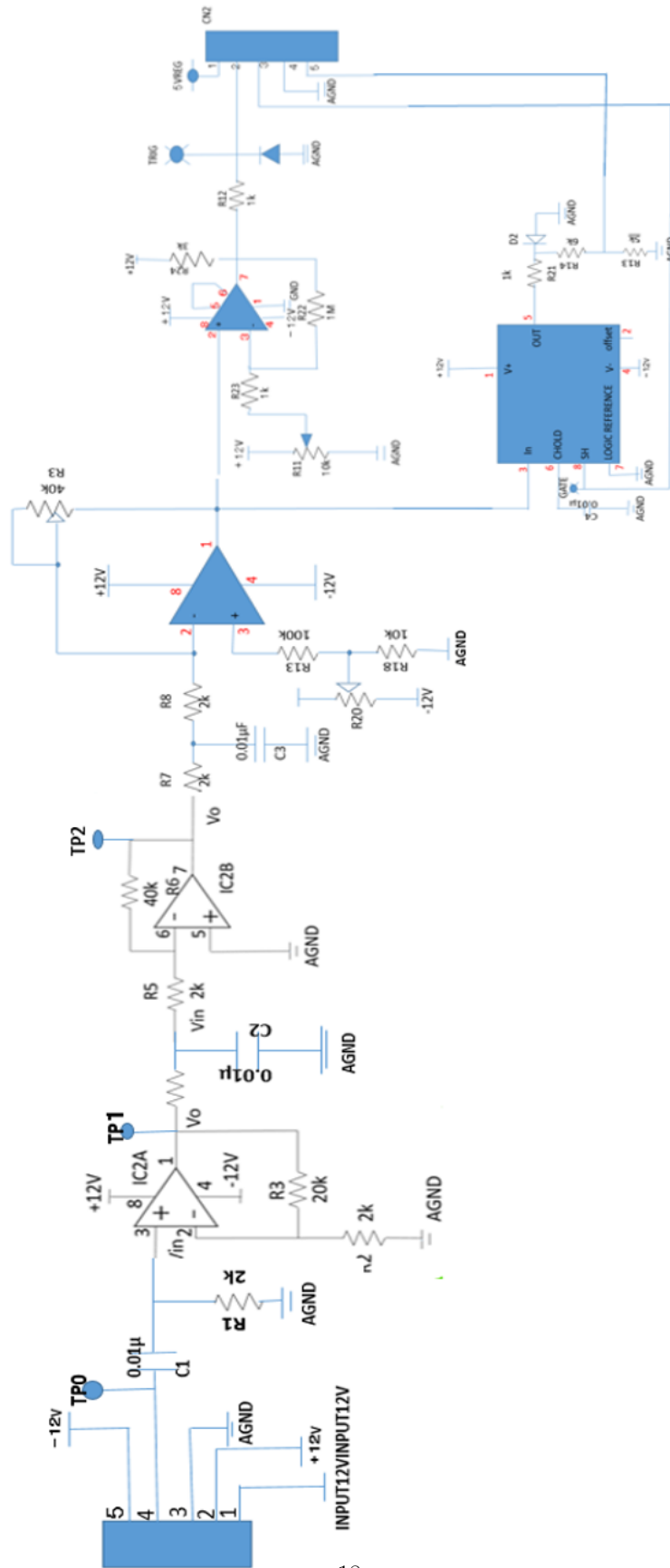


図 3.10 シェイパーアンプ

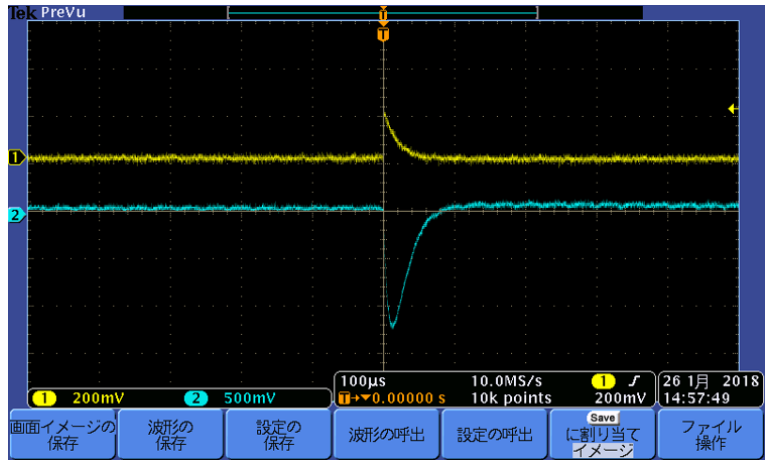


図 3.11 黄色：微分回路後増幅された波形
 水色:積分回路後反転増幅された波形

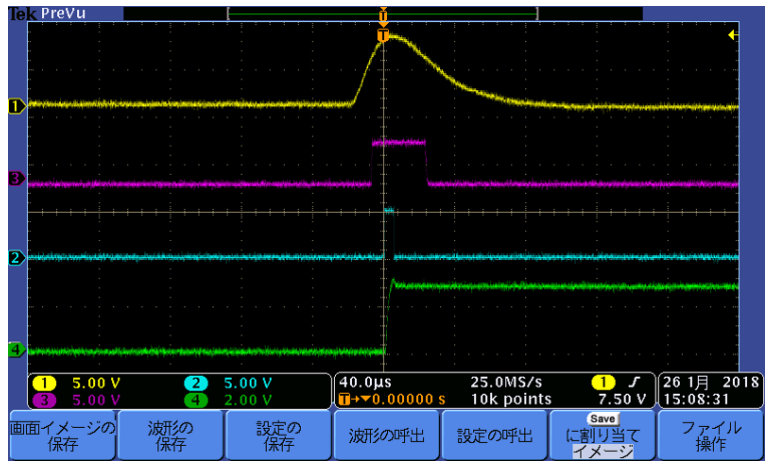


図 3.12 黄色：微分回路後反転増幅された波形
 ピンク：TRIGGER 信号
 水色：GATE 信号
 緑：サンプルアンドホールド回路の出力波形

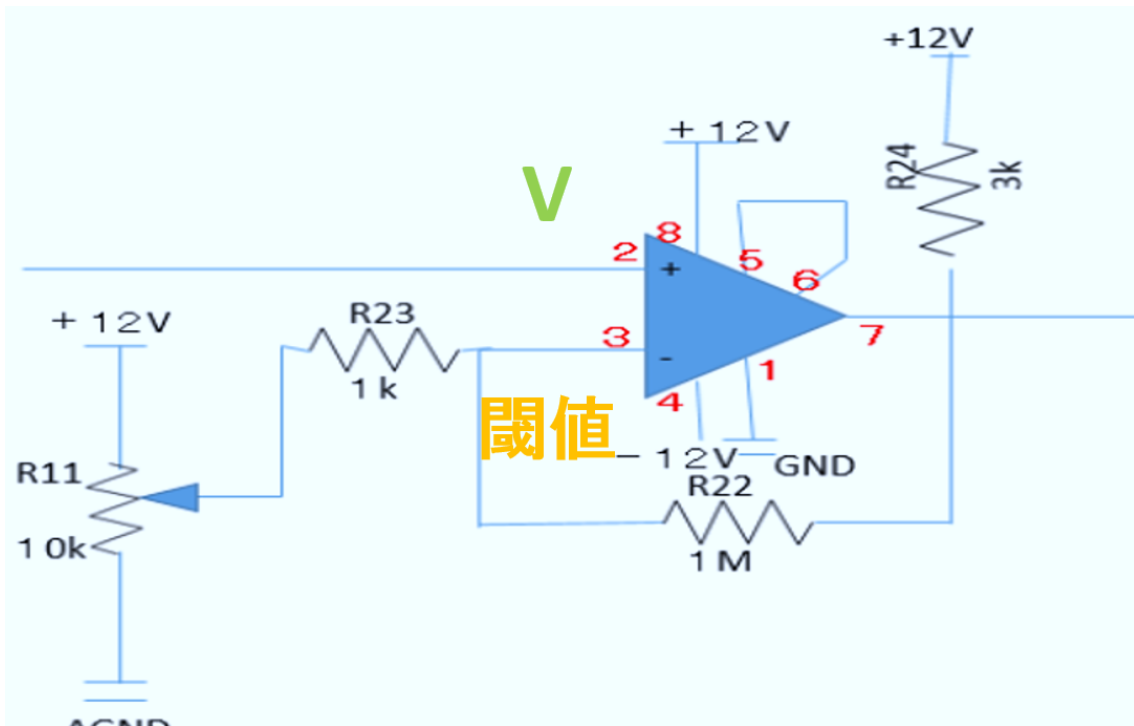


图 3.13 比較回路図

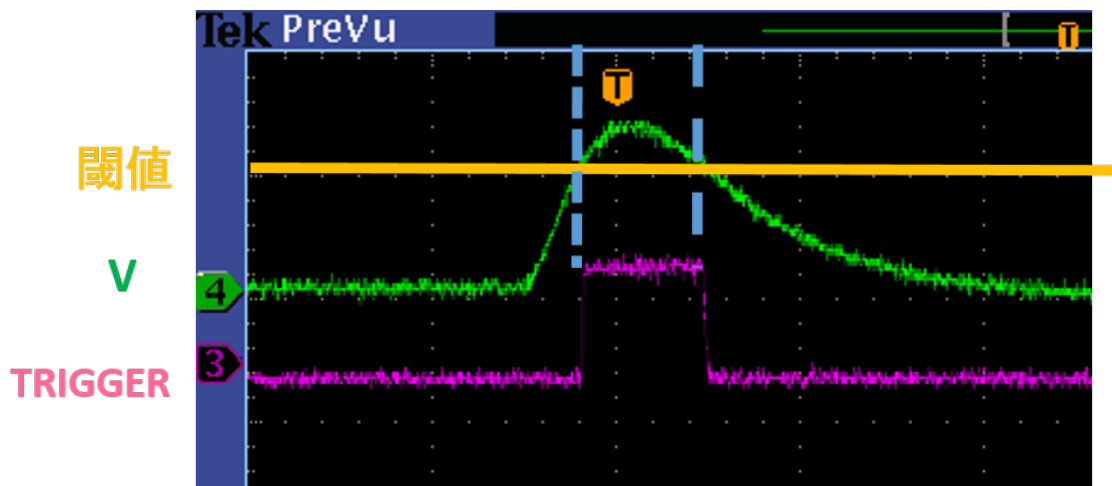


图 3.14 TRIGGER

3.3 波高分析とデータ収集

本研究ではデータ収集に ADC 内臓のワンチップマイコンボードを使用した。そのために、簡単に扱える Arduino を採用した。

3.3.1 Arduino

Arduino は、ハードウェアであるマイコンボードとソフトウェアである統合開発環境 (IDE) の 2 つのことを指し、ともにオープンソースのシステムである。図 3.15 に Arduino のマイコンボード、図 3.16 に Arduino IDE を示す。マイコンボード上に電子部品を接続し、IDE で開発したプログラムをマイコンボード上に書き込み、マイコンボードと電子部品を動かす。Arduino マイコンボードには種類があるが、今回は使いやすく最も普及している Arduino Uno を使用する。Arduino の仕様を表 3.1 に示す。



図 3.15 Arduino のマイコンボード

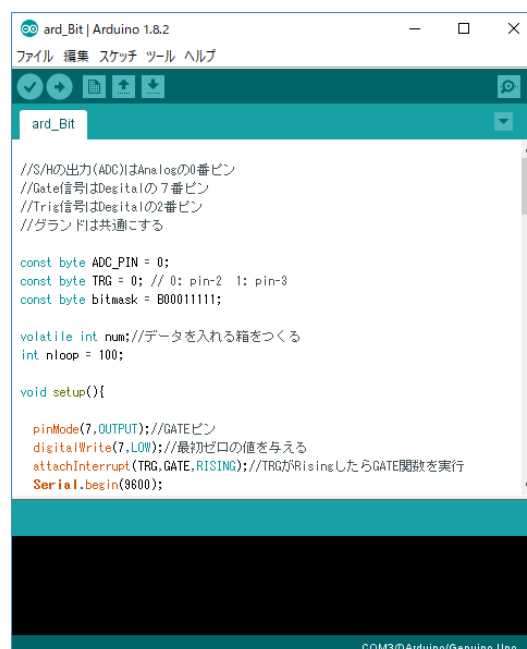


図 3.16 Arduino IDE

表 3.1 Arduino の仕様

Arduino 仕様	Arduino Uno R3
マイクロプロセッサ	ATmega328
動作電圧	5V
推奨入力電圧	7~12V
制限入力電圧	6~20V
デジタル I/O ピン数	14(うち 6 ピン PWM 出力)
PWM チャンネル	6
アナログ I/O ピン	6
I/O ピン電流	40mA
供給可能な最大電流	50mA
フラッシュメモリー	32K(うち 0.5KB はブートローダ用)
SRAM	2KB
EEPROM	1KB
クロック周波数	16MHz

今回の実験で、Arduino が行う波高分析の手順を図 3.17 に示す。

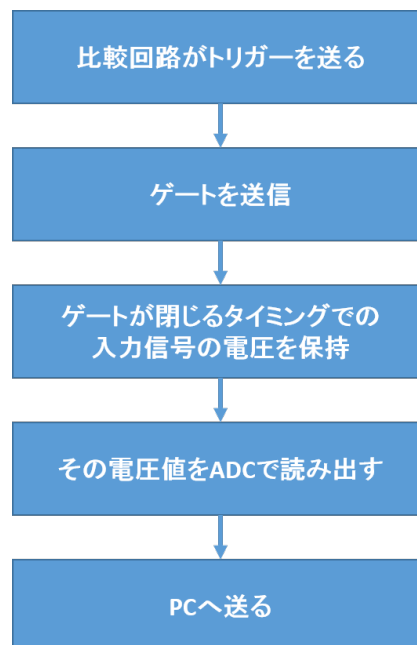


図 3.17 Arduino が行う波高分析の手順

3.3.2 Processing

Processing はプログラミングによる視覚的な表現を簡単に得るために作られたプログラミング言語であり、その言語は JAVA をベースにしている。また、オープンソースで開発されており、無料で利用できる。

本研究では Arduino で行われる波高分析の結果を PC に送り、ファイルに保存する処理を Processing が行う。図 3.18 は Processing の IDE である。



```
pro_Bit_OnMol_kai | Processing 1.5.1
File Edit Sketch Tools Help
STANDARD
pro_Bit_OnMol_kai
import processing.serial.*; //Serial libraryを読めるおまじない

//int totalTime = 1000; //(s)
int totalTime = 1000; //(s)
int startTime;
int currentTime;
int entry;
int missedEvent;
int raw_data;
int isHoldData;
int[] data = new int[1024];

String graph_t = "LinearTest 2016.6.23"; //リアルタイムで更新されるグラフの表示
//String graph_t = "Pedestal 2016.6.23"; //リアルタイムで更新されるグラフの表示
String filename = "Linear9mV_signal.txt"; //保存するファイルの名前
//String filename = "Linear9mV_pedestal.txt"; //保存するファイルの名前
//String filename = "Mantle_w_Pedestal150_2016Jun21.txt"; //保存するファイルの名前
//String filename = "Pedestal150_2016Jun21.txt"; //保存するファイルの名前

//GUI variables for data monitoring
```

図 3.18 Processing の IDE

図 3.19 に Processing が行う波高分析の手順を示す。Processing は Arduino から波高データを受け取り 8bit から 10bit へ変換、そしてファイルに書き込む。指定した時間を過ぎるとファイルを保存し新たなファイルを生成、そして波高データを受け取る作業に戻る。これにより時間変化の測定も可能である。

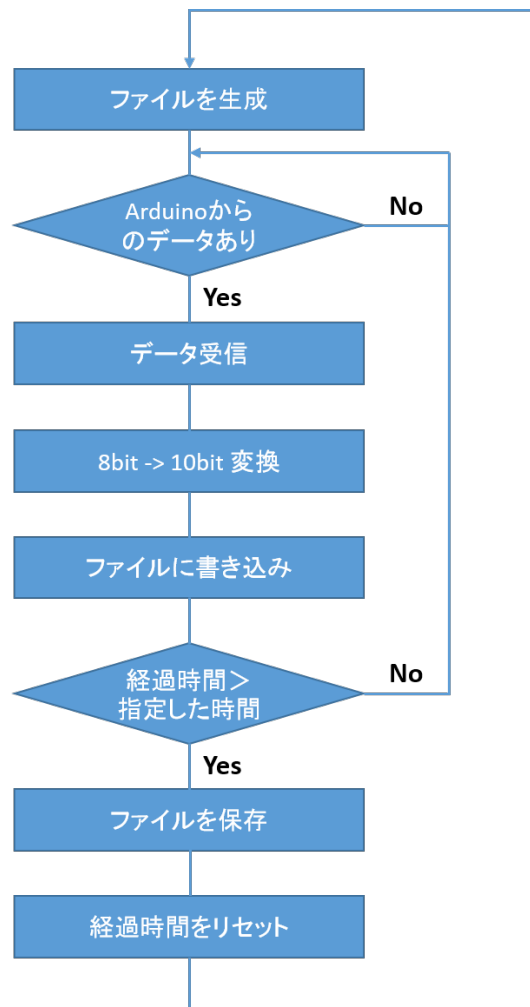


図 3.19 Processing が行う波高分析の手順

第4章 実験

4.1 LED 実験

4.1.1 実験概要

{241} Am やラドンガスから実際の α 線を検出する前に、シェイパーアンプが正しく動作しているか検証を行った。実験のセットアップを図1に示す。ラドン検出器のPDとシェイパーアンプとプリアンプとHVモジュールがついているふたの部分を縦に置き(図1左部分)、PDの高さに合うようにLEDをセットした(図1右部分)。LEDはパルスジェネレーターにつなぎ、周波数を1kHzに調整し、LEDを2.5Vで光らせるように抵抗を設定した。

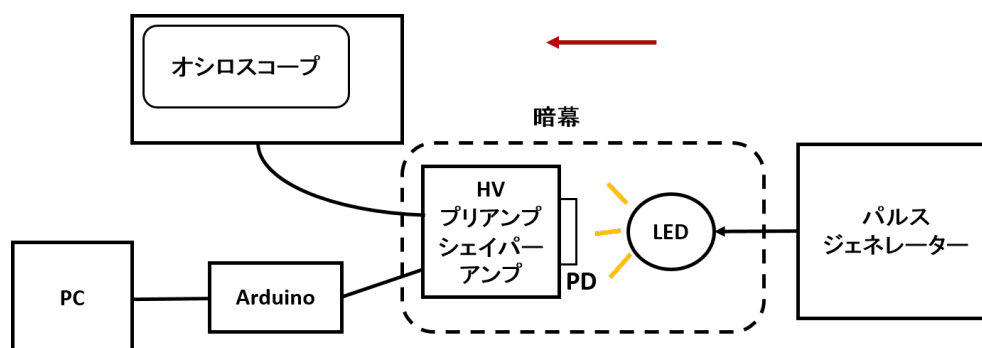


図4.1 LED 実験セットアップ

このLED実験によってシェイパーアンプ内の増幅回路、比較回路、サンプルアンドホールド回路の動作を各回路の出力の部分で確認した。シェイパーアンプ内の回路の構造は図2で示しており、シェイパーアンプ内のオシロスコープの読み取り点は図3で示した。図3でTP0がプリアンプからの出力を見る波形、TP1がハイパスフィルターを通して非反転増幅された後の波形、TP2は反転増幅された後の波形、TRIGGERが比較回路後の波形、GATEがArduinoからの出力波形、ADCがサンプルアンドホールドの出力波形を確認する点をそれぞれ示している。

プリアンプからの出力がシェイパーアンプのハイパスフィルターを通ったときの電圧を V_{in} 、TP1での信号電圧を V_1 、TP2での信号の電圧を V_2 とすると増幅回路の増幅率は次の式でそれぞれ表される。

$$V_1 = \frac{R2 + R3}{R2} V_{in} \quad (4.1)$$

$$V_1 = 11V_{in} \quad (4.2)$$

$$V_2 = \frac{R6}{R5} V_1 \quad (4.3)$$

$$V_2 = -20V_1 \quad (4.4)$$

式(2)、(4)より、非反転増幅回路後の信号は、プリアンプ回路を通して来た時の信号より、約11倍されており、反転増幅回路を通ると約20倍されていることがわかる。今回使用したシェイパーアンプでは、反転増幅回路と非反転増幅回路の間にローパスフィルターの抵抗が挟まっているので、倍率は、約半分になることが予想される。

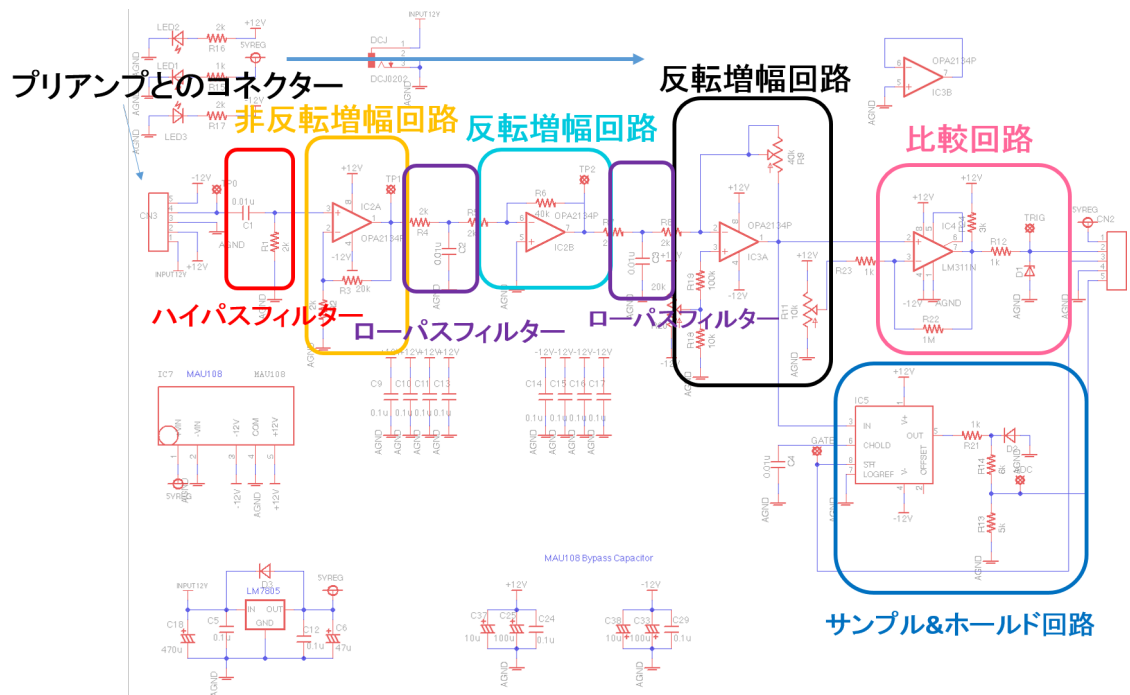


図 4.2 シェイパーアンプの回路説明

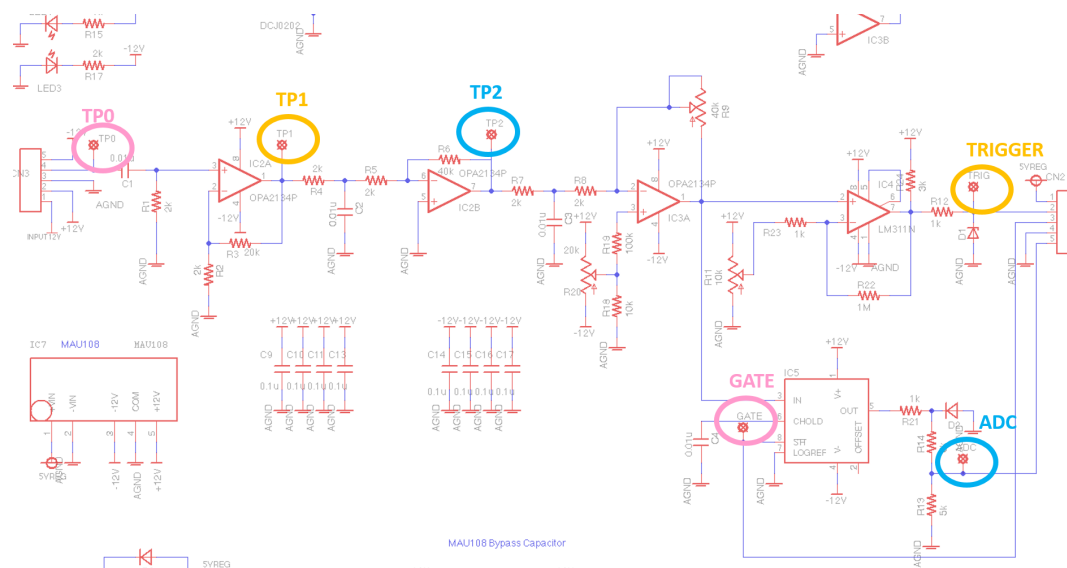


図 4.3 オシロスコープで確認した点

図 3 の各点をオシロスコープで観測した波形を図 4、5 に示した。

図 4 から、プリアンプ出力後の波形では、時定数の関係で見えなかった波形が、ハイパスフィルターと非反転増幅回路を通ることによって、TP1 では、1V ほどに見えていることがわかる。また、さらにハイパスフィルターを通過し、反転増幅回路を通った TP2 の点では、波形が反転し、増幅しているので、10V ほどになっているのが見える。理論式と一致しているので、シェイパーアンプの増幅回路は正常に動作していることがわ

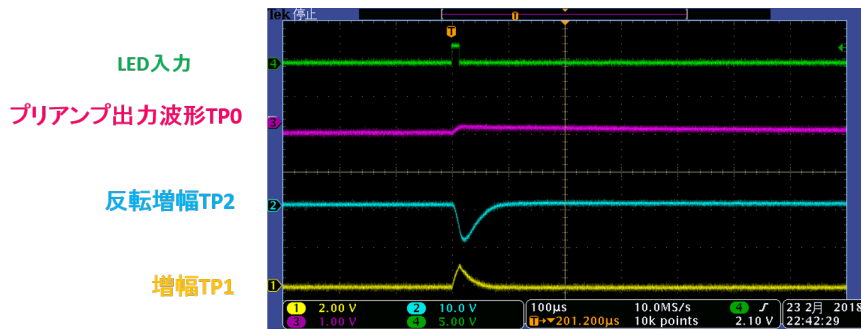


図 4.4 LED 実験 1：増幅回路部分のオシロスコープ

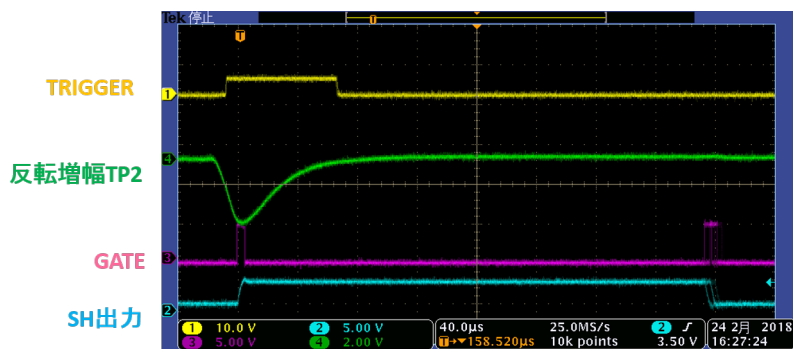


図 4.5 LED 実験 2：比較回路、GATE 信号、サンプルアンドホールド回路の出力を確認

かった。次に比較回路、サンプルアンドホールド回路、Arduino からの出力信号をそれぞれ TRIGGER ピン ADC ピン GATE ピンにオシロスコープを当てることで確認した。この時、比較回路の閾値は 5.0V に設定した。この時、図 3 より反転増幅回路後の TP 2 の波形はもう一度反転増幅される。反転増幅された後の波形を TP3 とすると、TP3 で測定される電圧の値が、比較回路の変可抵抗で設定した閾値と比較されることで、TP3 が閾値電圧より高くなった時に TRIGGER が出る仕組みになっている。LED 実験では、装置の都合上、TP3 の値をオシロスコープで見ることができなかった。そのため、この実験では、TRIGGER がどんなタイミングで出ているのか、正確なタイミングは知ることができない。図 5 より、TRIGGER がたったタイミングで GATE が立ちあがっているのがわかる。また、GATE の立下りで出力の値が保持されていることが SH 出力の波形からわかる。この結果から、図 2 で示した回路が TRIGGER 信号のタイミング以外正しく動作していることがわかった。

4.2 ^{241}Am 線源を用いた α 線測定

4.2.1 実験目的

今回製作したラドン検出器の動作確認およびエネルギー分解能の評価を行うため、 ^{241}Am 線源を用いて α 線を測定した。 ^{241}Am 線源のエネルギーは 5.4MeV である。

4.2.2 実験方法

今回使用した ^{241}Am 線源は強度が非常に強く、信号処理をする前に次ぎの信号がきてしまうので、 α 線の頻度を下げするために図 4.6 のように約 1mm 径の穴を開けた紙をつけた。また、 ^{241}Am 線源から放出される α 線はフォトダイオード表面に到達するまでに空気との相互作用によってエネルギーを失うので、可能な限りフォトダイオード表面に近づけて 5 分間測定を行った (図 4.7)。実験条件を表 4.1 に示す。また、0ch に対応するエネルギーを知るために、信号がないときの電圧値を 5 分測定した。このときの条件も同様に表 4.1 に示す。セットアップは図 4.8 のようにした。ここで GATE 幅とは、出力値を決める幅のことで Arduino で変えることができる。



図 4.6 穴の開いた紙をつけた ^{241}Am 線源



図 4.7 PD に限りなく近づけた ^{241}Am 線源

表 4.1 実験条件

	α 線	0 点
フォトダイオード表面にかかる電圧	-600V	-600V
オフセット	0.6V	0.6V
閾値	3.0V	3.0V
GATE 幅	8 μ s	100 μ s

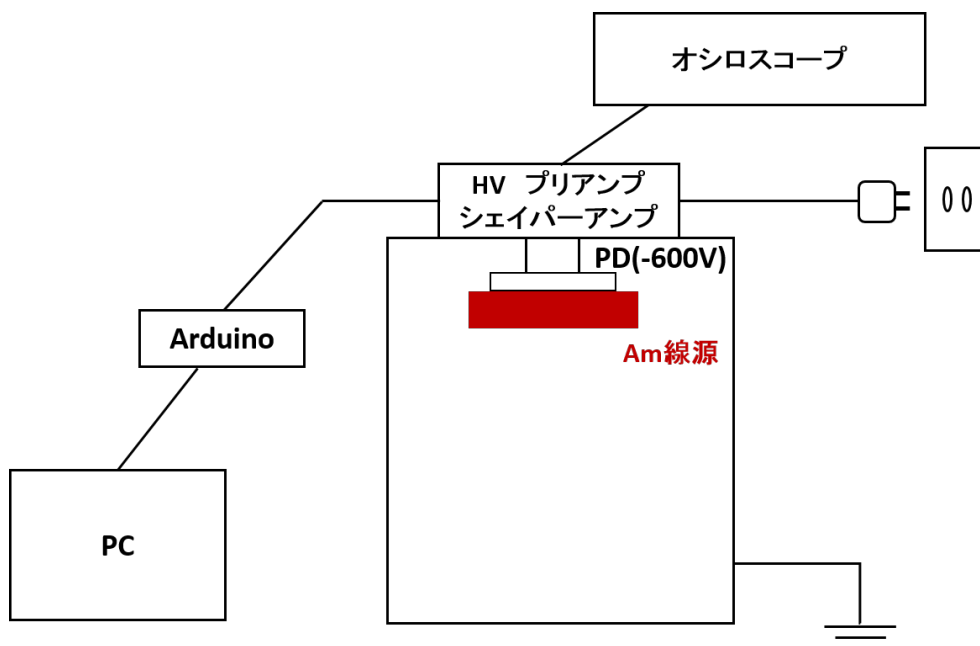


図 4.8 セットアップ

4.2.3 結果と考察

図 4.9 は GATE 幅が $8 \mu\text{s}$ のときの測定結果である。横軸は Pulse Height(ch)、縦軸はカウント数である。ガウス関数でフィットしたところ、490.5ch にピークが見られた。このピークが Am による α 線である。このときオシロスコープで観測した波形の様子を図 4.10 に示す。青が入力波形、赤がトリガー信号、黄色がゲート信号、緑が出力波形を表す。

次に、GATE 幅が $100 \mu\text{s}$ のときの測定結果を図 4.11 に示す。GATE 幅を長くすることで、入力波形が 0V を測定できる。このとき、オフセットを 0V の状態にしていると、0ch にカウント数がたまっていく現象が起きてしまったが、オフセットをあげることでピークが見やすく改善された。このときオシロスコープで観測した波形の様子を図 4.12 に示す。青が入力波形、赤がトリガー信号、黄色がゲート信号、緑が出力波形を表す。この二つの結果からエネルギーと Pulse Height の関係がわかるので、ラドン検出で見られるピークのエネルギーを知ることができる。

得られた結果を使ってエネルギー分解能の評価を行った。まずガウス関数でフィットしたときの平均値 M 、標準偏差 σ を求める。平均値は 490.5 で、標準偏差は 5.03 となった。また Am の α 線のエネルギー E は 5.4MeV である。

$$\sigma_E = \frac{\sigma}{M} = \frac{5.03}{490.5} = 1.03(\%) \quad (4.5)$$

よって、今回製作した波高分析システムのエネルギー分解能は 1.03 % であることがわかった。

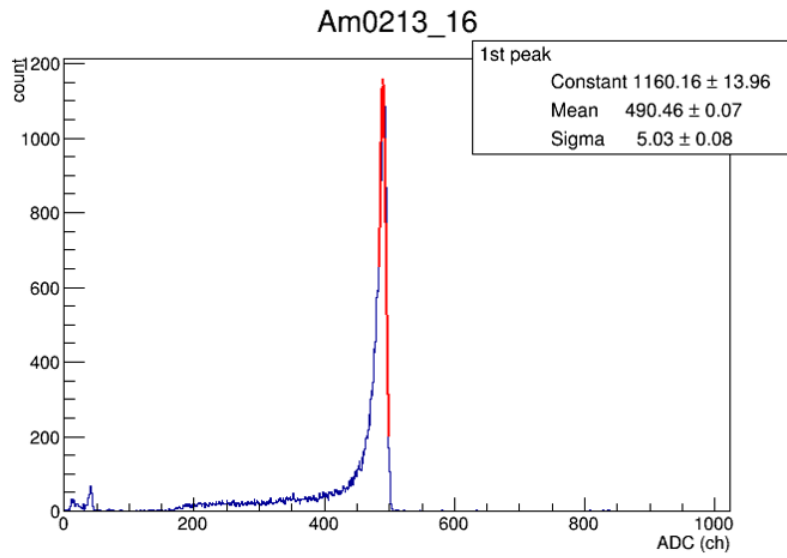


図 4.9 GATE 幅が $8 \mu s$ のときの測定結果

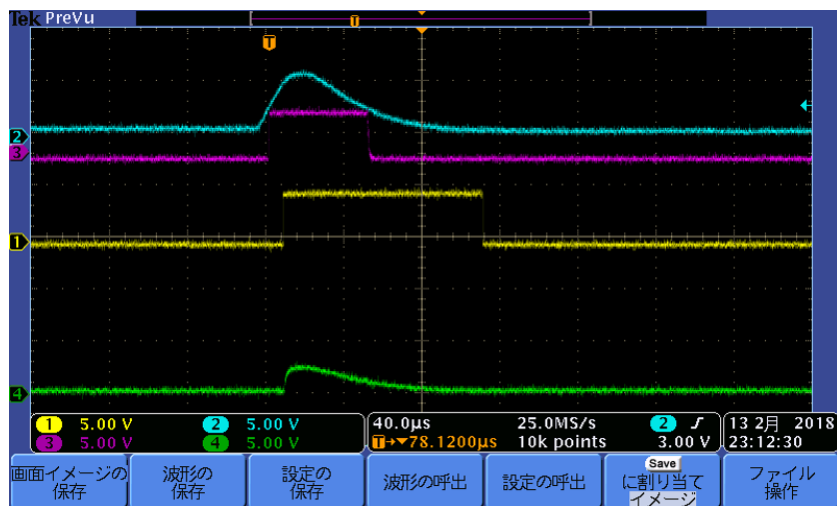


図 4.10 GATE 幅が $8 \mu s$ のときのオシロスコープの様子

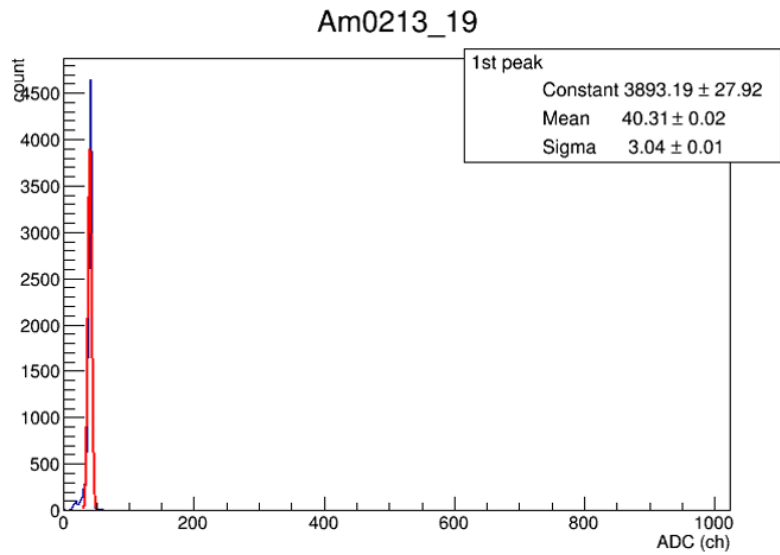


図 4.11 GATE 幅が $100 \mu s$ のときの測定結果

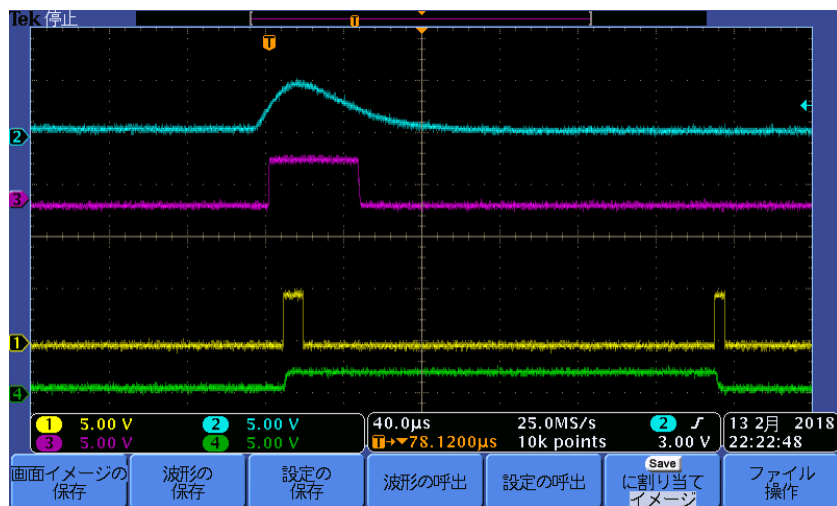


図 4.12 GATE 幅が $100 \mu s$ のときのオシロスコープの様子

また、Am のエネルギーと 0 点となるエネルギーをパルスハイトに対応させてエネルギー較正直線を描いた (図 8)。それぞれの数値は表にまとめて示した。

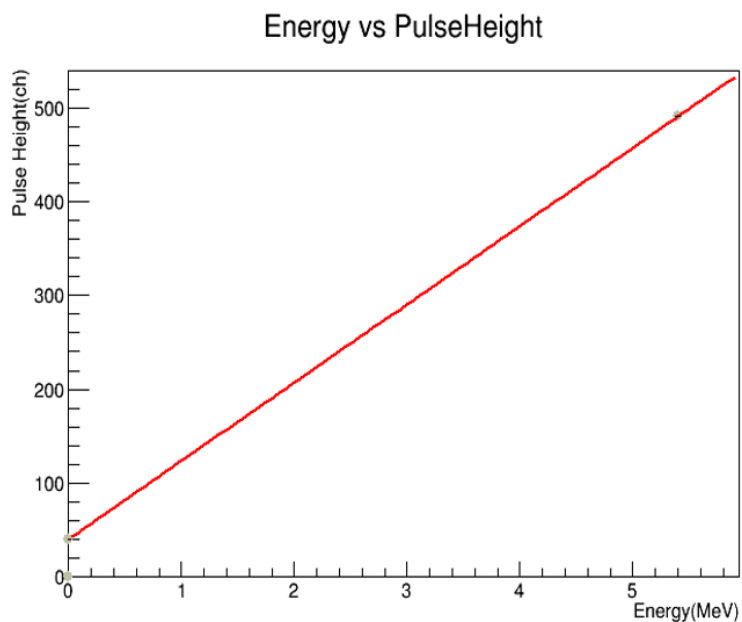


図 4.13 Am のエネルギー結果で作成したエネルギー較正直線

表 4.2 ch 数とエネルギーの関係

	x(MeV)	y(ch)	xerror	yerror
0 点	0.0	40.3	0.0	0.02
Am	5.4	490.5	0.0	0.07

この結果を用いてラドンを測定したときに見えるピークの値が何 ch でどのエネルギーに対応するか計算した。

4.3 ラドンガス測定

4.3.1 実験内容

ラドン検出器内に Doll Stone を 3 日ほど置いてラドンガスを充満させ Doll Stone は検出器内から取り出して 19 時間測定した。ここで Doll Stone とは、微量の天然ウランを含む岩石・土砂を原材料とした素焼きのタイルで、天然のウランが約 0.05 % 含まれている。サイズは 100mm × 100mm × 15mm。このように Doll Stone は純粋なラドン石ではなくピークがたくさん見えてしまうので測定する前に取り出した。フォトダイオード表面にかかる電圧は-600V である。セットアップを図 4.15 に示す。



図 4.14 Doll Stone

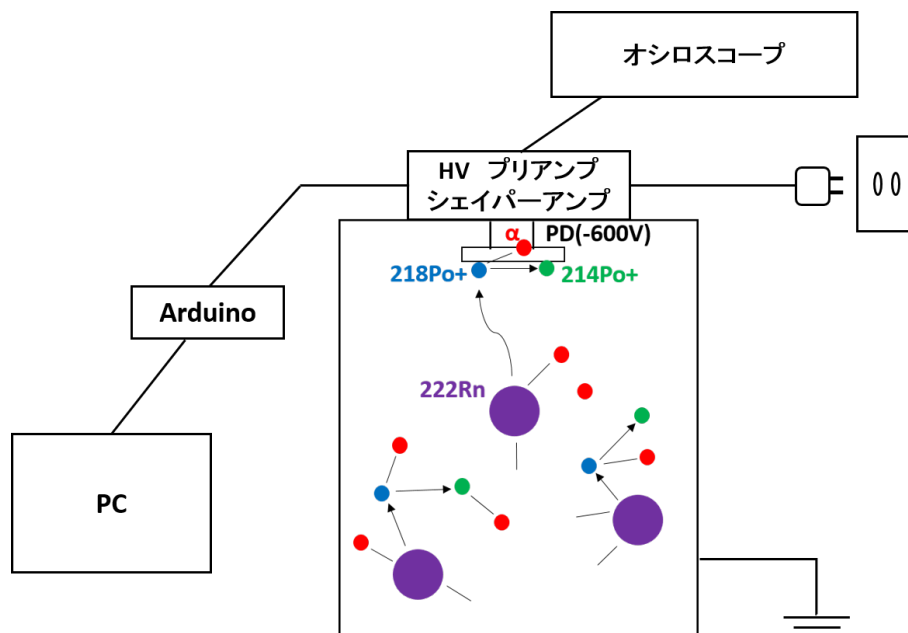


図 4.15 セットアップ

4.3.2 実験結果

ラドン検出器でラドンを上記の方法で測定した結果、図 4.16 でしめすような結果となった。この時のオフセットは 600 mV, GATE 幅は 8 μ 閾値は 2.997V で行った。

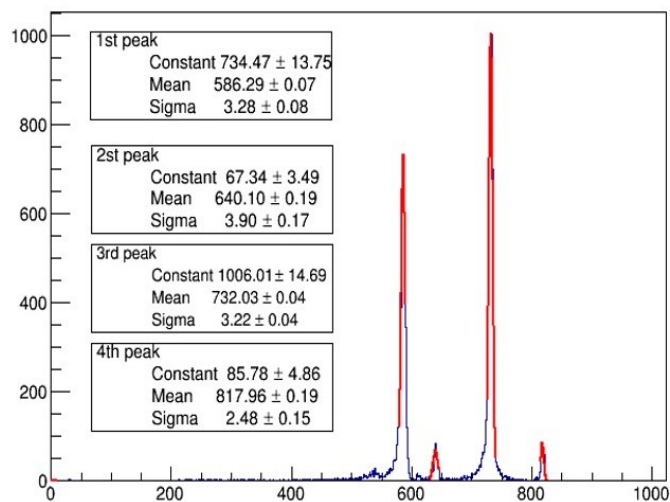


図 4.16 ラドンガスを時間計測したときの波高分布

この結果を、3章で示した Am エネルギー較正直線に当てはめると、図 4.17 で示す結果となる。この結果から、Po は誤差では表せられないほどの大きさで、エネルギー較正直線から外れていることがわかる。

ここで、Po 静電捕集法で捕集しているため、比較的能量ロスがなく捕集できていると仮定できる。そのため、Po を軸にしてエネルギー較正直線を作成し、そこに、実験で得られた Am の ch 数を当てはめてみると、次のような図が得られた。

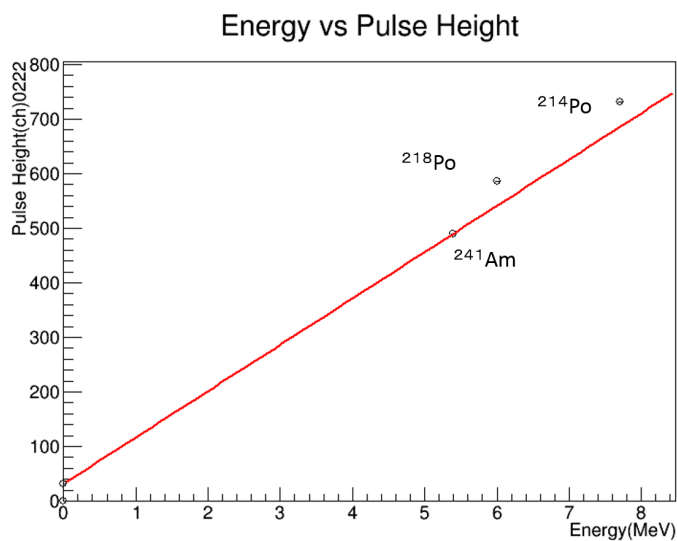


図 4.17 測定した Po の ch 数を Am のエネルギー較正直線に当てはめた結果

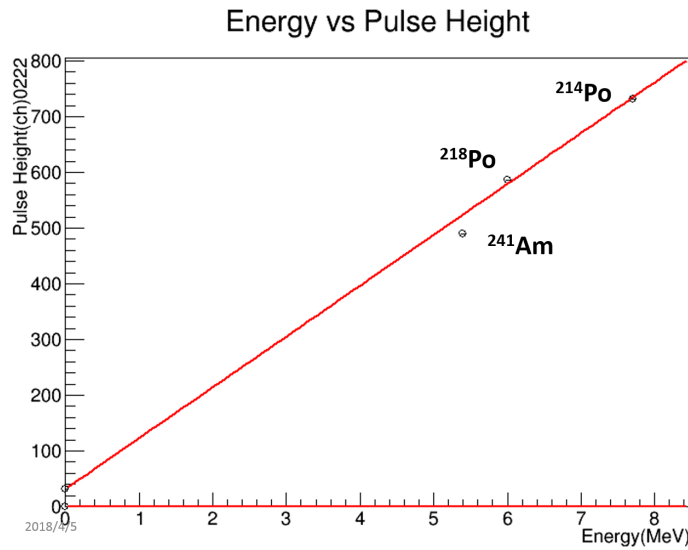


図 4.18 測定した Am の c h 数を Po のエネルギー較正直線に当てはめた結果

図 4.18 を見てみると、測定された Am が実際の c h 数よりも低い ch 数で観測されたのではないかと仮定できる。Am のエネルギーは 5.4MeV であることから、Po で作ったエネルギー較正直線に当てはめて考えると、約 36.0 ch の実験値との差分 Am からの α 線のエネルギーが失われていることがわかった。この差を、エネルギーに換算すると、Po のエネルギー較正直線の式は

$$y = 91.338x + 32.903 \quad (4.6)$$

であることから、0.36MeV のエネルギーが失われていることがわかった。

4.3.3 ベーテプロッホの式

0.36MeV のエネルギーがどこで損失されたのかを考える。今回 Am の宣言として使用した α 線源は密封線源を使用している。この密封線源の幕を通過したときにエネルギーが失われているのではないかと仮定して、ベーテプロッホの式でその厚さを計算した。この時、密封線源の幕を Al だと仮定して計算を行った。ベーテプロッホの式は次式で表される。

$$\frac{-dE}{dx} = 2 \pi N_{\alpha} (e)^2 m_e c^2 \rho \frac{Z}{A} \frac{z^2}{\beta^2} \log\left(\frac{2m_e^2 v^2 W_{max}}{I^2 - 2\beta^2}\right) [MeV/cm] \quad (4.7)$$

ここで、

$$2 \pi N_{\alpha} (e)^2 m_e c^2 = 0.1535 MeV (cm^2/g) \quad (4.8)$$

とした。また、

$$\begin{aligned} N_{\alpha} &: \text{アボガドロ数} = 6.022 \times 10^{23} \\ e &: \text{古典電子半径} = 2.817 \times 10^{-13} \\ m_e &: \text{電子素量} = 1.602 \times 10^{-19} C \\ \rho &: \text{吸収体の密度} (g/cm^2) \end{aligned}$$

Z : 吸収体の原子番号
 A : 吸収体の原子量
 z : 入射粒子の素電荷比での電荷
 $\beta = v/c$
 $= 1/\sqrt{1 - \beta^2}$
 I : 平均励起ポテンシャル
 W_{max} : 1回の衝突で電子が得る最大エネルギー

$M_{ij}m_e$ の範囲では、

$$W_{max}m_e c^2 \eta^2$$

今回、吸収体を Al として $\rho=2.7(\text{g}/\text{cm}^3)$ 、 $Z=13$ 、 $A=26.98$ で計算を行うと、

$$\frac{-dE}{dx} = 1727.47(\text{MeV}/\text{cm})$$

なので、Al が密封線源の幕に使われているときその厚さは、

$$\begin{aligned} \frac{0.36}{1727.47(\text{MeV}/\text{cm})} &= 2.08 \times 10^{-4}(\text{cm}) \\ &= 2.08(\mu\text{m}) \end{aligned}$$

この値は密封線源の膜厚にもっともらしい値といえる。

つまりエネルギー較正直線から計算した Am の 0.36eV のエネルギーの損失は、Am から放出される α 線が密封線源の窓でエネルギーを損失しているといえる。

この結果から、図 4.18 のエネルギー較正直線は Pulse Height と ch 数の関係を正しく示しているとともに、今回製作したラドン検出器は ^{218}Po と ^{214}Po からの α 線を検出できたといえる。

4.3.4 半減期の測定

測定で得られた α 線のエネルギーとカウント数を用いて、時間変化を調べることで測定した放射線の半減期を求めることができる。図 4.19 はラドンガスを時間計測したときの波高分布である。図 4.20 は横軸がラドンガスを検出器に充満させて測定を開始してからの経過時間、縦軸が 5 分ごとにピークの 2σ の範囲にくるエントリー数を経過時間でプロットしたものである。これを ^{218}Po と ^{214}Po のそれぞれのピークに対してプロットした。 ^{222}Rn 、 ^{218}Po 、 ^{214}Pb 、 ^{214}Bi 、 ^{214}Po は各々次の微分方程式に従い連鎖的に崩壊する。

$$^{222}\text{Rn} : -dN_1 = -\lambda_1 N_1 dt \quad (4.9)$$

$$^{218}\text{Po} : -dN_2 = (\lambda_1 N_1 - \lambda_2 N_2) dt \quad (4.10)$$

$$^{214}\text{Pb} : -dN_3 = (\lambda_2 N_2 - \lambda_3 N_3) dt \quad (4.11)$$

$$^{214}\text{Bi} : -dN_4 = (\lambda_3 N_3 - \lambda_4 N_4) dt \quad (4.12)$$

$$^{214}\text{Po} : -dN_5 = (\lambda_4 N_4 - \lambda_5 N_5) dt \quad (4.13)$$

N_1 、 N_2 、 N_3 、 N_4 、 N_5 はそれぞれ ^{222}Rn 、 ^{218}Po 、 ^{214}Pb 、 ^{214}Bi 、 ^{214}Po の原子数、 λ_1 、 λ_2 、 λ_3 、 λ_4 、 λ_5 はそれぞれ ^{222}Rn 、 ^{218}Po 、 ^{214}Pb 、 ^{214}Bi 、 ^{214}Po の崩壊定数、 t は経過時間である。

また、半減期を求める式は

$$e^{-\lambda T_{1/2}} = e^{-T_{1/2}/\tau} \quad (4.14)$$

これを解いて、

$$T_{1/2} = (\log_e 2) / \lambda \quad (4.15)$$

^{222}Rn の半減期は ^{218}Po に比べて十分大きいので、 ^{218}Po の半減期よりも十分大きいスケールでは、 ^{218}Po の減少の仕方は ^{222}Rn の半減期に従って減少するはずである。このように親核の半減期が娘核の半減期より十分に長い時、親核である Rn の半減期に従って娘核である Po が減少しているように見える現象を永続平衡という。

従って式 (1) より、 ^{222}Rn の単位時間当たりの崩壊数を求めると、

$$\lambda_1 N_1 = -\lambda_1 N_1(0) e^{-\lambda_1 t} \quad (4.16)$$

となり、この式 (8) で ^{218}Po のピークに対するデータ点を Fit した結果が図 4.20 の赤の曲線である。この結果得られた崩壊定数 λ_1 及び、 λ_1 から式 (7) で得られる半減期を ^{222}Rn の文献値と比べた結果を表 4.3 に示す。同様に、 ^{222}Rn の半減期は ^{218}Po 、 ^{214}Pb 、 ^{214}Bi 、 ^{214}Po の半減期に比べて十分大きい。 ^{214}Po の半減期よりも大きいスケールでは、 ^{214}Po のピークに対するデータ点を式 (4.14) で Fit を行った。その結果が図 4.20 の赤線の結果であり、得られた崩壊定数 λ_1 および半減期を、 ^{222}Rn の文献値と比べた。その結果を表 4.4 に示す。

表 4.3 ^{222}Rn 崩壊数の時間変化の式の fitting 結果 : ^{218}Po

	fitting 結果	文献値
$\lambda_1 (/s)$	$(5.10 \pm 0.05) \times 10^{-5}$	2.09×10^{-6}
半減期 (day)	0.1541 ± 0.0014	3.82

表 4.4 ^{222}Rn 崩壊数の時間変化の式の fitting 結果 : ^{214}Po

	fitting 結果	文献値
λ_1	$(5.705 \pm 0.011) \times 10^{-5}$	2.09×10^{-6}
半減期 (day)	0.1396 ± 0.0026	3.82

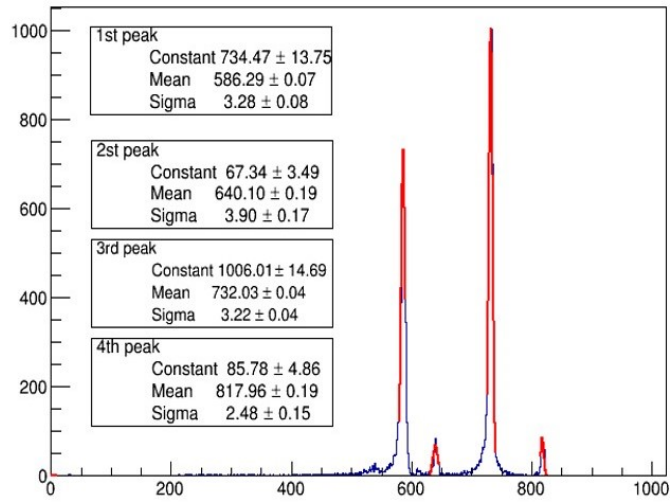


図 4.19 ラドンガスを時間計測したときの波高分布

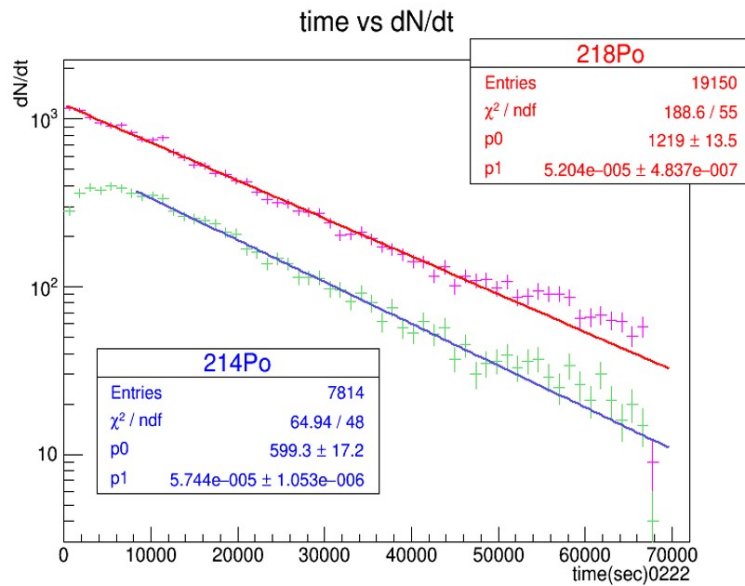


図 4.20 式 (8) でそれぞれ ^{218}Po を Fit した結果 (赤)、 ^{214}Po を Fit した結果 (青)

次に、 ^{218}Po の崩壊も考慮した式で fit を行った。式 (4.10) を用いて ^{218}Po の崩壊数の時間変化の式を求めると、

$$\lambda_2 N_2 = N_1(0)(e^{-\lambda_1 t} - e^{-\lambda_2 t}) \quad (4.17)$$

となる。

しかし、

4.3.5 改善策

半減期を文献値に近づけるために 2 つの改善を試みた。1 つ目は、缶の蓋に隙間を埋めるようにテープを巻き密封度を上げるよう改善した (図 4.21)。

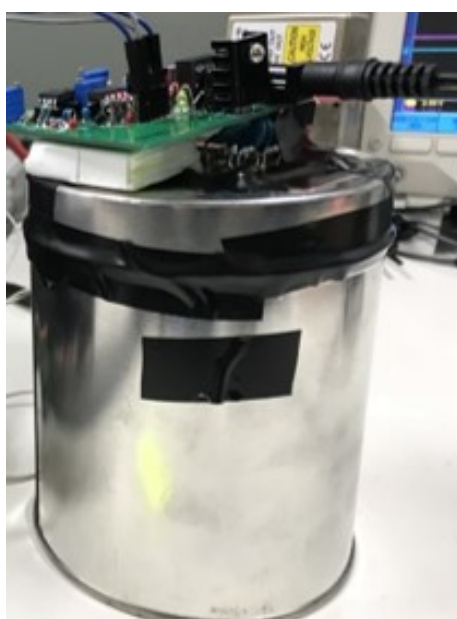


図 4.21 缶の蓋にテープを巻いた

その結果、図 4.22 と図 4.23 に示すように傾きがわずかに緩やかになった。半減期になおすと、約 50 分長くなった。ただし、文献値には程遠いのでまだまだ改善は必要である。

2 つ目は、缶の中に乾燥剤 (シリカゲル) を入れ湿気による静電捕集の妨げを改善した。このとき缶の蓋はテープで完全に塞いでいる。その結果、図 4.24 と図 4.25 に示すように傾きがわずかに緩やかになった。半減期になおすと、この方法でも同様に約 50 分長くなった。

よって、2 つの改善策から半減期が約 100 分長くなったので、もう少し工夫をすることでより文献値に近づくことが期待できる。

ただし、ここで一つ問題がある。本来、フォトダイオード表面に吸着した ^{218}Po が崩壊した後 ^{214}Po が崩壊するため図 4.22 や図 4.25 のように ^{214}Po が ^{218}Po の上にくるグラフが得られるはずである。しかし、図 4.23 や図 4.24 (二つの結果は同じ実験結果である) は反対に ^{218}Po が ^{214}Po の上にきている。これはフォトダイオード表面に吸着したはずの ^{218}Po がなんらかの影響で取れてしまったか、あるいはまったく別の理由なのか、なぜこのような再現性のない結果になってしまったのかはまだわかっていない。

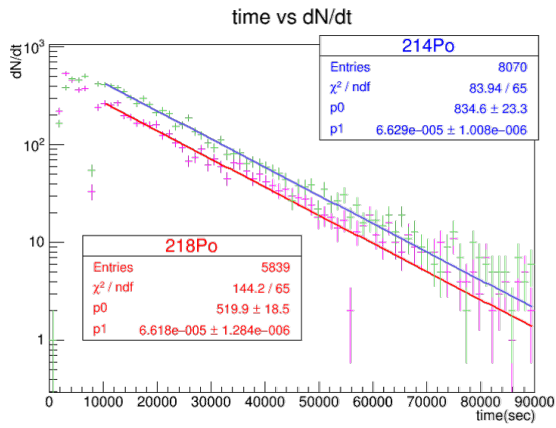


図 4.22 テープを巻く前

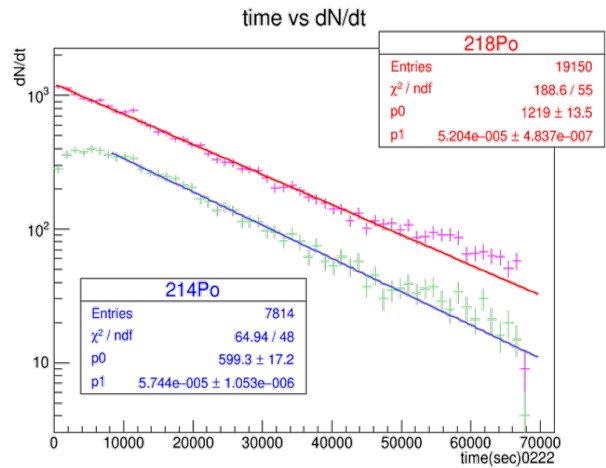


図 4.23 テープを巻いた後

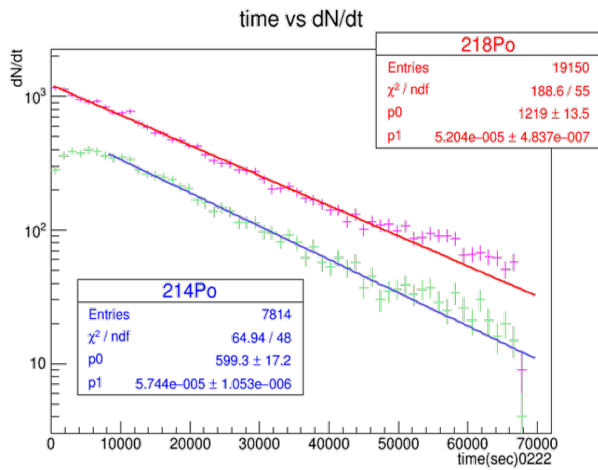


図 4.24 乾燥剤を入れる前

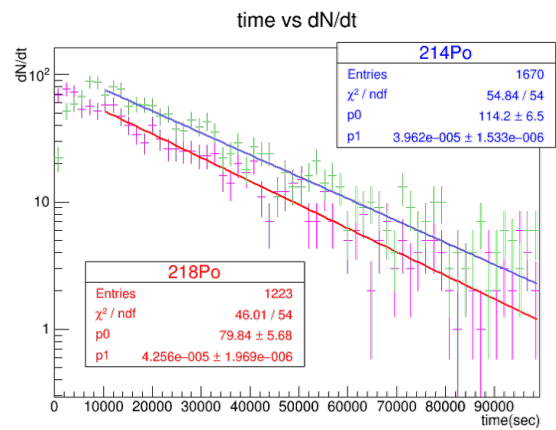


図 4.25 乾燥剤を入れた後

第5章 まとめ

5.1 まとめ

筑波大学の協力のもと、小型ラドン検出器の開発・製作を行った。LEDを使用した実験により波高分析回路が正常に動作していることを確認した。エネルギーと波高の関係をj知るために Am 線源から放出される α 線を測定した。この結果からエネルギー較正直線を作り、ラドンガスの測定で得られた二つのピークが ^{218}Po と ^{214}Po であることがわかった。さらに放射性崩壊の式を用いて半減期を求めようとしたが、理論値とはるかに異なる結果になってしまった。その原因が缶の密封性、湿気などにあると考え改善した結果、半減期が 100 分ほど長くなった。

5.2 今後の課題

缶の密封度や湿気が半減期に影響することがわかったので、さらに高性能かつ安価な容器を製作する。Doll Stone から収集したラドンガスだけでなくより高濃度のラドンを測定する。 ^{218}Po と ^{214}Po の崩壊する順番の問題を解決する。

5.3 謝辞

本研究に取り組むにあたって、下村真弥先生、蜂谷崇先生にはお忙しい時間を割いていただき、あらゆる面でご指導くださったことに深く感謝しております。林井久樹先生には、回路の仕組みなど基本的なことを講義も交えて教えていただきました。私たちが実験をしている様子を気にかけてはご助言をくださりお世話になりました。宮林謙吉先生には、機器の使い方や動作確認などでつまづいていたところを助けていただきました。深く感謝すると共に御礼申し上げます。

また、筑波大学の皆様には先行研究として非常に参考にさせていただくことばかりでした。私たちの至らない質問にも親切に答えてくださり、深く感謝すると共に御礼申し上げます。

本研究を進めるにあたり、支えてくださったすべての方々に深く感謝すると共に、この場を借りて御礼申し上げます。

第6章 付録

6.1 Arduino プログラム

Arduinoのプログラム

```
//S/Hの出力(ADC)はAnalogの0番ピン ←Arduinoのアナログ入力は10bit  
//Gate信号はDigitalの7番ピン  
//Trig信号はDigitalの2番ピン  
//グラウンドは共通にする
```

```
const byte ADC_PIN = 0;  
const byte TRG = 0; // 0: pin-2 1: pin-3  
const byte bitmask = B00011111;
```

```
volatile int num;//データを入れる箱をつくる
```

```
void setup(){  
  
    pinMode(7,OUTPUT);//GATEピン  
    digitalWrite(7,LOW);//最初ゼロの値を与える  
    attachInterrupt(TRG,GATE,RISING);//TRGがRisingしたらGATE関数を実行  
    Serial.begin(9600);  
  
}
```

1

```
void GATE(){  
  
    detachInterrupt(TRG); //not to interrupt during this function  
  
    //Gate信号  
    digitalWrite(7,HIGH);  
    delayMicroseconds(10);//単位はusec  
    digitalWrite(7,LOW);  
  
    num = analogRead(ADC_PIN);//アナログピンの値を読み込む  
  
    delayMicroseconds(100);//時間をおく ←オシロスコープと一致しない  
  
    //リセット用のGate信号  
    digitalWrite(7,HIGH);  
    delayMicroseconds(10);  
    digitalWrite(7,LOW);  
  
    senddata( num );  
    attachInterrupt(TRG,GATE,RISING);  
  
}
```

2

```
byte getParity( byte val ){
```

3

```
    byte x;
```

```
    x = val >> 4; //4ビット右シフトする
```

```
    val = val^x; //XOR演算子
```

```
    x = val >> 2; //2ビット右シフトする
```

```
    val = val^x;
```

```
    x = val >> 1; //1ビット右シフトする
```

```
    val = val^x;
```

```
    return ( val & 1 ); //
```

```
}
```

3 と 4 は 10bit を 8bit に変換する作業（Processing へ一度にデータを送ることができる値は 0 255 に限られている）

analogRead で読んだ値をビット演算で 10bit から 8bit に変換し 2 つに分けて Processing に送っている

4

```
void senddata( int val ){  
  
    //convert data  
    byte lowVal = ( val & bitmask );  
    byte higVal = ( ( val >> 5 ) & bitmask );  
  
    //horizontal parity check  
    byte lowParity = getParity( lowVal );  
    byte higParity = getParity( higVal );  
  
    //vertical parity check  
    byte andParity = getParity( higVal & lowVal );  
    byte orParity = getParity( higVal | lowVal );  
  
    //make packet  
    byte higBit = B10000000;  
    byte lowBit = B00000000;  
    higBit += ( higVal << 2 );  
    lowBit += ( lowVal << 2 );  
    higBit += ( andParity << 1 );  
    lowBit += ( orParity << 1 );  
    higBit += higParity;  
    lowBit += lowParity;  
  
    Serial.write( higBit );  
    Serial.write( lowBit );  
  
}
```

6.2 Processing プログラム

Processingのプログラム

```
import processing.serial.*; //Serial libraryを読めるおまじない

//int totaltime = 1000; // (s)
int totaltime = 100000; // (s)
int startTime;
int currentTime;
int entry;
int missedEvent;
int raw_data;
int isHoldData;
int[] data = new int[1024];

String graph_t = "radon0123_3"; //リアルタイムで更新されるグラフの表示タイトル
String filename = "radon0123_3.txt"; //保存するファイルの名前
```

```

//GUI vairables for data monitoring
int lens = 5; //拡大縮小倍率
float zoom_max = 125.0;
float zoom_min = 0.04;
float fZoom = 1.0;
float plotX1, plotX2;
float plotY1, plotY2;
float labelX, labelY;
float tmp_y;
boolean press=false;
String words;

PrintWriter outFile;//保存用の変数を用意
Serial usbPort;
int port_id;
int nport;
boolean selectedPort;

String user_filename;
char letter;

//for serial communication
class packet {
    int digit;
    int data;
    int check;
    int parity;
    packet(){}}

};
packet pkt1 = new packet();
packet pkt2 = new packet();

int mask_digit = unbinary("10000000");
int mask_data = unbinary("01111100");
int mask_check = unbinary("00000010");
int mask_parity = unbinary("00000001");

```

```

//初期設定
void setup(){

    size(1124,600);

    //ファイル生成
    outFile = createWriter( filename );//ファイルを生成
    outFile.print("TIME(s)");//1行ごとの書き込み
    outFile.print(TAB);
    outFile.println("PulseHeight(ch)");

    //canvas area
    plotX1 = 50;
    plotX2 = width - plotX1;
    plotY1 = 90;
    plotY2 = height - plotY1;

    //initialization
    entry = 0;
    isHoldData = 0;
    missedEvent = 0;
    selectedPort = false;

}

void draw() {

    currentTime = (millis() - startTime)/1000;
    if( currentTime >= totaltime ){
        outFile.flush();//データの保存
        outFile.close();//ファイルを閉じる
        exit(); // Stops the program
    }

    //Online monitoring
    makeCanvas();
    makeButton();
    drawTitle();
    drawTime(currentTime);
    drawAxis();
    drawHistogram();
    strokeWeight(1);

    if( !selectedPort ) setSerialPort();
}

```



```
void analyzePacket( int val, packet p){  
  
    p.digit = ( val & mask_digit ) >> 7;  
    p.data  = ( val & mask_data  ) >> 2;  
    p.check = ( val & mask_check ) >> 1;  
    p.parity = ( val & mask_parity );  
}
```

```
int getParity( int val ){  
  
    int x;  
    x = val >> 4;  
    val = val^x;  
    x = val >> 2;  
    val = val^x;  
    x = val >> 1;  
    val = val^x;  
  
    return ( val & 1 );  
}
```

```

int getData( packet p1, packet p2 ){

    int p_sum1 = 0;
    int p_sum2 = 0;

    //horizontal parity check
    if( getParity(p1.data)!=p1.parity ) return -1;
    if( getParity(p2.data)!=p2.parity ) return -1;

    //vertical parity check
    int andParity = getParity( p1.data & p2.data );
    int orParity = getParity( p1.data | p2.data );
    if( andParity!=p1.check ) return -1;
    if( orParity!=p2.check ) return -1;

    //convert raw data to ADC value
    int fValue = ( p1.data << 5 );
    fValue += p2.data;
    if( fValue<0 || fValue>1023 ) return -1;

    return fValue;
}

```

```

void serialEvent(Serial usbPort){ //勝手に始まる関数

    if( usbPort.available() > 1 ){

        if( isHoldData!=1 ){

            raw_data = usbPort.read();
            analyzePacket( raw_data, pkt1 );
        }else{

            pkt1 = pkt2;
            //check if pkt2 is lower bit
            if( pkt2.digit==0 ){

                int value = getData( pkt1, pkt2 );

                if( value==1 ) missedEvent++;
                else saveData( value );
                isHoldData = 0;

            }else{
                isHoldData = 1;
            }
        }

        //check if pkt1 is higher bit
        if( pkt1.digit!=1 ){

            missedEvent++;
            return;

        }else{

            raw_data = usbPort.read();
            analyzePacket( raw_data, pkt2 );

        }

        //available

    }

}

```

2018/4/9

```
void saveData( int adc ){  
  
    entry++;  
    data[adc]++;  
  
    int passedtime = millis() - startTime;  
    int new_passedime = floor(passedtime/1000);  
  
    outFile.print(new_passedime);//1行ごとの書き込み  
    outFile.print(TAB);  
    outFile.println(adc);  
    outFile.flush();//データの保存  
  
}
```