

2018年度卒業論文
小型ラドン検出器の性能評価及び改良

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
青山美嶺 西川愛

2019年3月19日

目次

第 1 章	序論	6
1.1	ラドンについて	6
1.2	ラドンの崩壊系列	6
1.3	α 線とは	8
1.4	半減期	9
1.5	逐次壊変	9
1.6	放射平衡	11
第 2 章	検出原理	13
2.1	Si PIN フォトダイオード	13
2.2	静電捕集法	15
2.3	フォトダイオードに流れる電流	16
第 3 章	ラドン検出器	17
3.1	ラドン検出部	17
3.2	アナログ部	18
3.3	データ制御・収集部	24
第 4 章	実験	29
4.1	^{241}Am 線源を用いた α 線測定	29
4.2	ラドンガス測定	34
4.3	半減期の導出	36
4.4	小型ラドン検出器の改良	38
4.5	密封度の効果	42
第 5 章	まとめ	46
5.1	まとめ	46
5.2	今後の課題・展望	46
付録 A	プログラム	49
A.1	Arduino のプログラム	49

目次	2
A.2 processing のプログラム	50
参考文献	54

目次

1.1	ウラン系列	7
1.2	トリウム系列	7
1.3	アクチウム系列	7
1.4	放射線の透過	8
2.1	Si PIN フォトダイオード	13
2.2	構成/最大絶対定格	14
2.3	PDの電気的および光学的な特性	14
2.4	フォトダイオードに順バイアスをかけた状態	15
2.5	フォトダイオードに逆バイアスをかけた状態	15
2.6	検出器内の様子	16
3.1	ラドン検出器	17
3.2	アナログ部の流れ	18
3.3	プリアンプの回路図	19
3.4	電荷増幅器	19
3.5	シェイパーアンプの回路図 [10]	20
3.6	波形整形回路の配置	21
3.7	ハイパスフィルター	22
3.8	ローパスフィルター	22
3.9	非反転増幅回路	22
3.10	反転増幅回路	22
3.11	比較回路	23
3.12	TP0,TP1,TP2,TP3 の波形	23
3.13	トリガー,TP3,Gate,ADC の波形	23
3.14	Arduino のボード	24
3.15	Arduino のソフトウェア (I D E)	24
3.16	Arduino Uno R3 の仕様	25
3.17	Arduino との配線	25
3.18	Arduino が行う波高分析手順	26

3.19	processing の IDE	27
3.20	processing が行う波高手順	28
3.21	オンラインモニタリングの様子 (縦軸:カウント数 横軸:パルスの高さ)	28
4.1	穴の開けた紙をつけた ^{241}Am 線源	30
4.2	フォトダイオードに出来るだけ近づけた ^{241}Am 線源	30
4.3	オシロスコープでみた波形	30
4.4	実験結果の波高分布	31
4.5	^{241}Am によるエネルギー較正直線	33
4.6	Doll stone	34
4.7	ラドンガスの波高分布	35
4.8	^{241}Am によるエネルギー較正直線に ^{218}Po 、 ^{214}Po の点をプロットした	36
4.9	先行実験の缶を用いた時のラドンガスの波高分布	37
4.10	図 4.9 のデータを指数関数で fit した結果	37
4.11	先行実験の缶の蓋	38
4.12	一つ目の工夫の蓋	38
4.13	二つ目の工夫の蓋	38
4.14	プリアンプ装着時 (表)	39
4.15	プリアンプ装着時 (裏)	39
4.16	開ける穴の大きさ	39
4.17	使用するコネクタ	39
4.18	最終的な完成図: 表 (左) 裏 (右)	39
4.19	プリアンプの装着	40
4.20	プリアンプのグランド接続	40
4.21	市販のゴムパッキン	40
4.22	バスコーク N を塗った小缶の蓋	40
4.23	小缶を用いた時のラドンガスの波高分布	41
4.24	図 4.23 のデータを式指数関数で fit した結果	41
4.25	図 4.23 のデータを指数関数で fit した結果 (単位時間は 100 分)	41
4.26	開けたねじ穴 0 個の時のラドンガスの波高分布	43
4.27	図 4.26 のデータを指数関数で fit した結果	43
4.28	開けたねじ穴 1 個の時のラドンガスの波高分布	43
4.29	図 4.28 のデータを指数関数で fit した結果	43
4.30	開けたねじ穴 2 個の時のラドンガスの波高分布	43
4.31	図 4.30 のデータを指数関数で fit した結果	43
4.32	開けたねじ穴 3 個の時のラドンガスの波高分布	44
4.33	図 4.32 のデータを指数関数で fit した結果	44

表目次

3.1	ラドン検出器のサイズ	18
4.1	エネルギーと ch の関係	33
4.2	図 4.10 の fit 結果	38
4.3	図 4.24 の fit 結果	42
4.4	図 4.25 の fit 結果	42
4.5	図 4.27 の fit 結果	44
4.6	図 4.29 の fit 結果	44
4.7	図 4.31 の fit 結果	44
4.8	図 4.33 の fit 結果	45

第1章

序論

1.1 ラドンについて

ラドン ^{222}Rn は、原子番号が 86、無味無臭、無色の気体の放射性希ガス元素である。 ^{238}U から崩壊してきたラドン (^{222}Rn) や、 ^{232}Th からの崩壊で生じるラドン (^{220}Rn 、トロンと呼ばれている) は自然界の至るところに存在しており、これらが呼吸により体内に入り、内部被曝を起こす。日本人は一年間に約 2.4mSv の放射線を受けているというが、その中の 1.3mSv はラドンなどによる被ばくであり、ラドンは非常に身近にある放射性物質であるといえる。

1.2 ラドンの崩壊系列

α 崩壊系列はウラン系列、トリウム系列、アクチウム系列、ネプツニウム系列が存在しているが、その中のネプツニウム系列のみ天然には存在しない。天然に存在している三つの系列はそれぞれ ^{238}U 、 ^{232}Th 、 ^{235}U を親核とし、それぞれ質量数を n を用いて、 $4n+2$ 、 $4n$ 、 $4n+3$ と表される核種の崩壊系列である。最終的にそれぞれが ^{206}Pb 、 ^{207}Pb 、 ^{208}Pb の安定状態になる。図 1.1 でウラン系列、図 1.2 でトリウム系列、図 1.3 でアクチウム系列の崩壊図を示した。

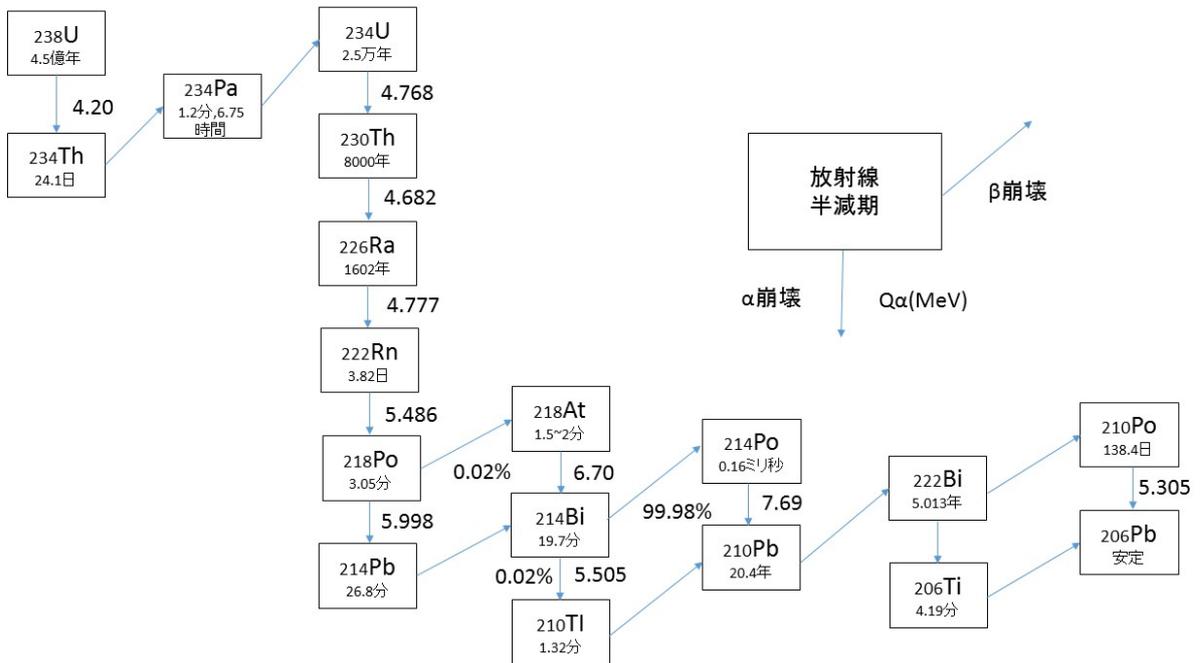


図 1.1 ウラン系列

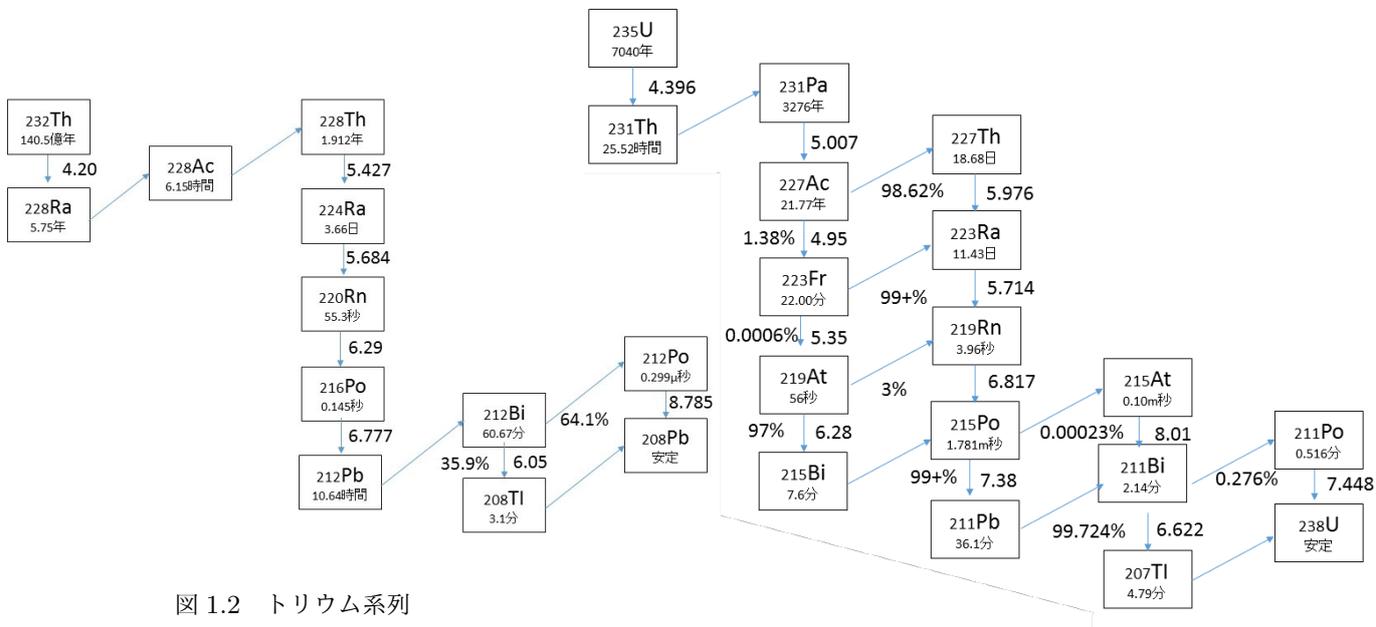


図 1.2 トリウム系列

図 1.3 アクチウム系列

1.3 α 線とは

α 線とは高速で飛ぶ「 α 粒子」の流れのことである。 α 粒子は陽子2個と中性子2個からなる ${}^4\text{He}$ の原子核である。 α 粒子は陽子2個を含んでいて、プラスの電気を帯びており、そのため「荷電粒子」と呼ばれる。 α 粒子は2価の陽イオンであり、強い電離作用を持っているため、透過作用がとても低く、大気中であれば2~3 cm、図1.4で示すように紙なら一枚で遮蔽することができる。

放射性核種が α 線を放出し、自然崩壊する現象を α 壊変または α 崩壊という。 α 線（陽子2個、中性子2個）を放出するために、娘核種の原子番号は2減り、質量数は4だけ小さい原子核になる $(Z, M) = (Z - 2, M - 4) + \alpha(2, 4)$ 。重い原子核の α 崩壊の寿命は実験によると10ns から 10^{17} y と報告されている。 α 崩壊によって発生した α 粒子のエネルギーは、 α 崩壊前の元素の種類とその質量数ごとに決まっているので、 α 線のエネルギーを観測できれば、核融合の結果生まれた崩壊前の元素の種類と質量数がわかる。そのため α 崩壊は新元素の確認などに使われることがある。

α 崩壊する放射線核種の大半は原子番号が83以上のものであるが、 ${}^{144}\text{Nd}$ 、 ${}^{147}\text{Sm}$ なども α 崩壊すると知られていて、崩壊核種は400以上知られている。エネルギーの高い α 粒子は加速器を用いて作られ、原子核反応を起こすのに利用される。

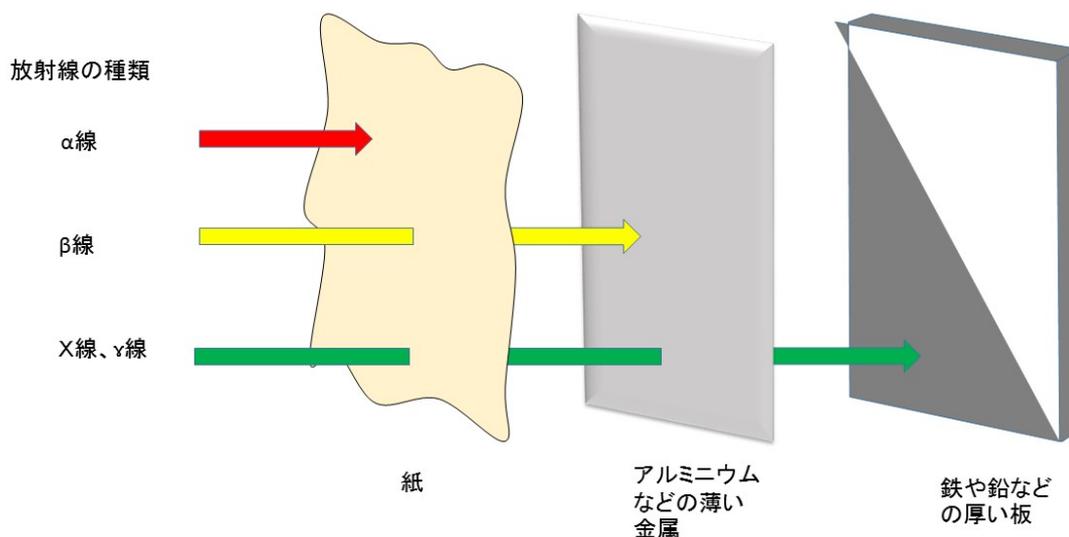


図 1.4 放射線の透過

1.4 半減期

最初にあった原子数が $\frac{1}{2}$ になるのに必要な時間のことを半減期という。半減期が $T_{1/2}$ の原子核が時刻 t のとき $N(t)$ 個あるとすると、時刻 $t = 0$ の時の原子核の個数は $N(0)$ と表すことができ、以下のような関係が成り立つ。

$$N(T_{1/2}) = N(0) \left(\frac{1}{2}\right)^1 \quad (1.1)$$

$$N(2T_{1/2}) = N(0) \left(\frac{1}{2}\right)^2 \quad (1.2)$$

$$N(t) = N(0) \left(\frac{1}{2}\right)^{t/T_{1/2}} \quad (1.3)$$

また微小時間 dt の間に崩壊する核の個数 dN は崩壊定数 λ を用いて

$$-dN = \lambda N dt \quad (1.4)$$

と表すことができ、この微分方程式を解くと以下の式になる。

$$N(t) = N(0) e^{-\lambda t} \quad (1.5)$$

式 1.5 の両辺の対数をとると、

$$\log_e N(t) = \log_e N(0) - \lambda t \quad (1.6)$$

と表されるので、式 1.3 と式 1.5 より、

$$\lambda T_{1/2} = \log_e 2 \quad (1.7)$$

と表すことができる。よって、崩壊定数が分かれば半減期を求めることができる。

1.5 逐次壊変

前の節では単一の放射性物質の崩壊を考えているが、この節ではある放射性物質が崩壊してできた物質が再び放射性物質である場合の崩壊を考える。この時、もともなった放射性物質を親核種、親核種の崩壊によってできた放射性物質を娘核種という。ある親核種の個数を N_1 、崩壊定数を λ_1 、その娘核種の個数を N_2 、崩壊定数を λ_2 とすると以下の関係式が成り立つ。

$$\frac{dN_2}{dt} = \lambda_1 N_1 - \lambda_2 N_2 \quad (1.8)$$

これは娘核種の個数の変化量は、微小時間 dt の間に崩壊した親核種の個数から微小時間 dt の間に崩壊した娘核種の個数を引いたものになることを表している。このことから、核種 1 から核種 2 へ、核種 2 から核種 3 へ、と一連の崩壊を順々に繰り返す崩壊系列を考えた時、 i 番目の核種の個数 N_i の変化量は次のように表せる。

$$\frac{dN_i}{dt} = \lambda_{i-1} N_{i-1} - \lambda_i N_i \quad (1.9)$$

時刻 $t = 0$ の時、核種 1 のみが $N_1(0)$ 個存在する場合を考える。まず、式 1.5 より

$$N_1(t) = N_1(0) e^{-\lambda_1 t} \quad (1.10)$$

となる。次に式 1.8 の解を考えるが、式 1.8 の右辺第 1 項は上の式より t の関数として $f(t)$ と置くことができるので、

$$\frac{dN_2}{dt} = f(t) - \lambda_2 N_2 \quad (1.11)$$

となり、

$$dN_2 = (f(t) - \lambda_2 N_2) dt \quad (1.12)$$

と変形できるので、 N_2 はある関数 $g(t)$ を用いて、

$$N_2 = g(t) e^{-\lambda_2 t} \quad (1.13)$$

の形となることが予想される。よって、 $g(t)$ を求めるためにこの式を式 1.8 へ代入して計算すると、

$$\frac{d}{dt} (g(t) e^{-\lambda_2 t}) = \lambda_1 N_1 - \lambda_2 g(t) e^{-\lambda_2 t} \quad (1.14)$$

$$\frac{dg(t)}{dt} e^{-\lambda_2 t} + \frac{de^{-\lambda_2 t}}{dt} g(t) = \lambda_1 N_1 - \lambda_2 g(t) e^{-\lambda_2 t} \quad (1.15)$$

$$\frac{dg(t)}{dt} e^{-\lambda_2 t} - \lambda_2 g(t) e^{-\lambda_2 t} = \lambda_1 N_1 - \lambda_2 g(t) e^{-\lambda_2 t} \quad (1.16)$$

$$\frac{dg(t)}{dt} e^{-\lambda_2 t} = \lambda_1 N_1 \quad (1.17)$$

ここで、 $\lambda_1 N_1$ は式 1.5 より、 $\lambda_1 N_1(0) e^{-\lambda_1 t}$ と表せるので、

$$\frac{dg(t)}{dt} e^{-\lambda_2 t} = \lambda_1 N_1(0) e^{-\lambda_1 t} \quad (1.18)$$

$$\frac{dg(t)}{dt} = \lambda_1 N_1(0) e^{\lambda_2 t - \lambda_1 t} \quad (1.19)$$

両辺を t で積分して、積分定数を C とすると、

$$g(t) = N_1(0) \frac{\lambda_1}{\lambda_2 - \lambda_1} e^{\lambda_2 t - \lambda_1 t} + C \quad (1.20)$$

時刻 $t = 0$ の時、 $N_2(0) = 0$ であるので、 $g(t) = 0$ となり、

$$C = -N_1(0) \frac{\lambda_1}{\lambda_2 - \lambda_1} \quad (1.21)$$

となる。よって、

$$g(t) = N_1(0) \frac{\lambda_1}{\lambda_2 - \lambda_1} e^{\lambda_2 t - \lambda_1 t} - N_1(0) \frac{\lambda_1}{\lambda_2 - \lambda_1} \quad (1.22)$$

となるので、

$$\begin{aligned}
 N_2(t) &= g(t) e^{-\lambda_2 t} \\
 &= \left(N_1(0) \frac{\lambda_1}{\lambda_2 - \lambda_1} e^{\lambda_2 t - \lambda_1 t} - N_1(0) \frac{\lambda_1}{\lambda_2 - \lambda_1} \right) e^{-\lambda_2 t} \\
 &= N_1(0) \frac{\lambda_1}{\lambda_2 - \lambda_1} (e^{-\lambda_1 t} - e^{-\lambda_2 t})
 \end{aligned} \tag{1.23}$$

となる。同様にして時刻 t における4番目の核種の個数 N_4 は、

$$N_4(t) = N_1(0) \lambda_1 \lambda_2 \lambda_3 \left(\frac{e^{-\lambda_1 t}}{C_1} + \frac{e^{-\lambda_2 t}}{C_2} + \frac{e^{-\lambda_3 t}}{C_3} + \frac{e^{-\lambda_4 t}}{C_4} \right) \tag{1.24}$$

$$C_1 = (\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)(\lambda_4 - \lambda_1)$$

$$C_2 = (\lambda_1 - \lambda_2)(\lambda_3 - \lambda_2)(\lambda_4 - \lambda_2)$$

$$C_3 = (\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)(\lambda_4 - \lambda_3)$$

$$C_4 = (\lambda_1 - \lambda_4)(\lambda_2 - \lambda_4)(\lambda_3 - \lambda_4)$$

と表せる。[9]

1.6 放射平衡

一連の崩壊を順々に繰り返す崩壊系列において、ある親核種やその娘核種の崩壊定数 λ の値の大きさによって、時間が経過したときの親核種と娘核種の関係性が決まっており、過渡平衡、永続平衡、または平衡にならない場合がある。

親核種1の半減期 T_1 が娘核種2の半減期 T_2 に比べて長い、すなわち壊変定数 λ_1, λ_2 の関係が $\lambda_1 < \lambda_2$ の時、十分時間が経過すると式1.23の $e^{-\lambda_2 t}$ は無視することができ、

$$N_2(t) = N_1(0) \frac{\lambda_1}{\lambda_2 - \lambda_1} (e^{-\lambda_1 t}) = N_1(t) \frac{\lambda_1}{\lambda_2 - \lambda_1} \tag{1.25}$$

$$\frac{N_2(t)}{N_1(t)} = \frac{\lambda_1}{\lambda_2 - \lambda_1} \tag{1.26}$$

と表すことができる。よって、十分な時間（娘核種2の半減期 T_2 の7~10倍）が経過すると、親核種1と娘核種2の原子数の比は一定となり、このことを過渡平衡という。

T_1 が T_2 に比べて非常に長い、すなわち壊変定数 λ_1, λ_2 の関係が $\lambda_1 \ll \lambda_2$ の時、時間が経過すると過渡平衡と同様に式1.23の $e^{-\lambda_2 t}$ は無視することができ、それに加えて $\lambda_2 - \lambda_1 = \lambda_2$ と考えることができるので、

$$N_2(t) = N_1(0) \frac{\lambda_1}{\lambda_2} (e^{-\lambda_1 t}) = N_1(t) \frac{\lambda_1}{\lambda_2} \tag{1.27}$$

$$\frac{N_2(t)}{N_1(t)} = \frac{\lambda_1}{\lambda_2} \tag{1.28}$$

と表すことができる。よって、時間（娘核種 2 の半減期 T_2 の約 10 倍）が経過すると、親核種 1 と娘核種 2 の原子数は半減期に比例し、それぞれの壊変率はほぼ等しくなる。このことを永続平衡という。

T_1 が T_2 に比べて短い、すなわち壊変定数 λ_1, λ_2 の関係が $\lambda_1 > \lambda_2$ の時、放射平衡は成立せず、親核種と娘核種の原子数の比は一定になることはない。

第2章

検出原理

今回の実験では、フォトダイオードによってラドンの娘核である Po が α 崩壊したときに放出されるエネルギーを検出することで間接的にラドンを検出する。

2.1 Si PIN フォトダイオード

フォトダイオードとは、光半導体素子の PN 接合部分に光を照射すると電流を発生させる受光素子である。Si フォトダイオードは光の有無、強弱、色などの検知に幅広く使われている。Si PIN フォトダイオードは任意の逆電圧を印加することにより、優れた応答特性を実現する。今回使用したのは Si PIN フォトダイオード、Hamamatsu S3590-09 である。図 2.1 に外観、図 2.2 に PD の定格、図 2.3 に PD の電氣的、光学的特性を示す。

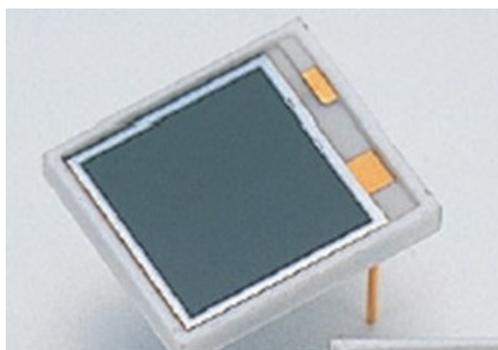


図 2.1 Si PIN フォトダイオード

型名	窓材	受光面サイズ (mm)	空乏層厚 (mm)	絶対最大定格			
				逆電圧 VR max.	許容損失 P (mW)	動作温度 Topr (°C)	保存温度 Tstg (°C)
S3590-08	エポキシ樹脂	10 × 10	0.3	100	100	-20 ~ +60	-20 ~ +80
S3590-09	未封止						
S3590-18	エポキシ樹脂						
S3590-19	未封止						

図 2.2 構成/最大絶対定格

型名	感度波長 範囲 λ (nm)	最大感度 波長 λp (nm)	受光感度 S				短絡電流 Isc 100 lx (μA)	暗電流 ID VR=70 V		暗電流の 温度係数 TCID VR=70 V (倍/°C)	遮断 周波数 fc VR=70 V (MHz)	端子間 容量 Ct f= 1MHz VR=70 V (pF)	雑音等 価電力 VR=70 V (W/Hz ^{1/2})
			λ=λp (A/W)	LSO 420 nm (A/W)	BGO 480 nm (A/W)	CsI(Tl) 540 nm (A/W)		Typ.	Max.				
S3590-08	340 ~ 1100	960	0.66	0.20	0.30	0.36	100	2	6	1.12	40	40	3.8 × 10 ⁻¹⁴
S3590-09			0.22	0.28	0.33	0.41	90	4	10				
S3590-18			0.65	0.28	0.34	0.38	100						
S3590-19			0.58	0.33	0.37	0.4	86						

図 2.3 PDの電気的および光学的な特性

PIN フォトダイオードは、P 型、I 型、N 型の三層構造の真性半導体である。P 層は価電帯中の正孔の数が電子より多い半導体で、positive charge を持つので P 型と名づけられている。N 層はたくさん電子をもつ半導体で、negative charge から N 型と名づけられている。I 層は不純物濃度が低い N 型半導体で電子、正孔どちらが多いわけでもなく、intrinsic の頭文字をとって I 型と名づけられ、また空乏層とも呼ばれている。PIN フォトダイオードに順バイアスをかけると、電子が陽極に、正孔が陰極に流れ、電流が流れる。逆バイアスをかけると、陽極に電子、陰極に正孔が引き寄せられ、電流は流れず、空乏層の幅が広がる。空乏層に α 線が入射すると、空乏層に電子-正孔対ができ、α 線自体のエネルギーがなくなるまで、電子-正孔を作り続け、電界のために正孔は P 型に、電子は N 型へ移動する。このときに P 型、N 型に外部回路を接続すると、P 型に正孔子、N 型に電子が流れ、電流を発生し、電気信号として表すことができる。今回は、PIN フォトダイオードに逆バイアスをかけることで、Rn が α 崩壊し Po になり、その Po が α 崩壊するときに出す α 線を検出して、間接的に Rn を特定する。

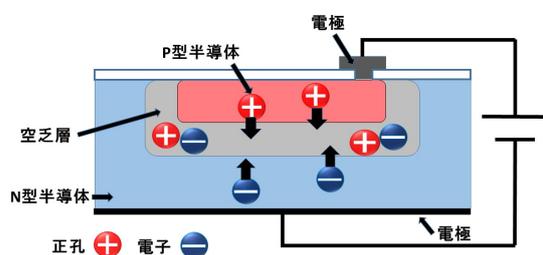


図 2.4 フォトダイオードに順バイアスをかけた状態

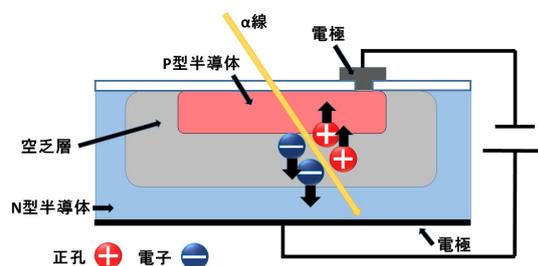


図 2.5 フォトダイオードに逆バイアスをかけた状態

https://www.hamamatsu.com/resources/pdf/ssd/s3590-08_etc_kpin1052j.pdf
[\https://www.hamamatsu.com/resources/pdf/ssd/02_handbook.pdf](https://www.hamamatsu.com/resources/pdf/ssd/02_handbook.pdf)

2.2 静電捕集法

α 線は、前節でも述べたように、大気中では2~3cmで止まってしまう、検出するのは容易ではない。なので今回は静電捕集法を用いて α 線を検出した。図2.6はラドン検出器内での様子である。検出器内にウランを含んでいる岩石であるDoll stoneを入れ、時間の経過とともにウランが次々に崩壊し、ラドンを放出する。缶を密閉することによりラドンガスが充満していく。ラドンの娘核であるポロニウムは崩壊により、プラスの陽イオンとなって缶内に漂う。この時に、缶上部にあるPDに負の電圧をかけ、缶の壁面をグラウンドにすることにより、缶内で静電気場が作られる。この結果、ラドンから崩壊してすぐのポロニウムはPDに引き寄せられ、効率よくポロニウムが崩壊するときに放出する α 線を検出することができる。

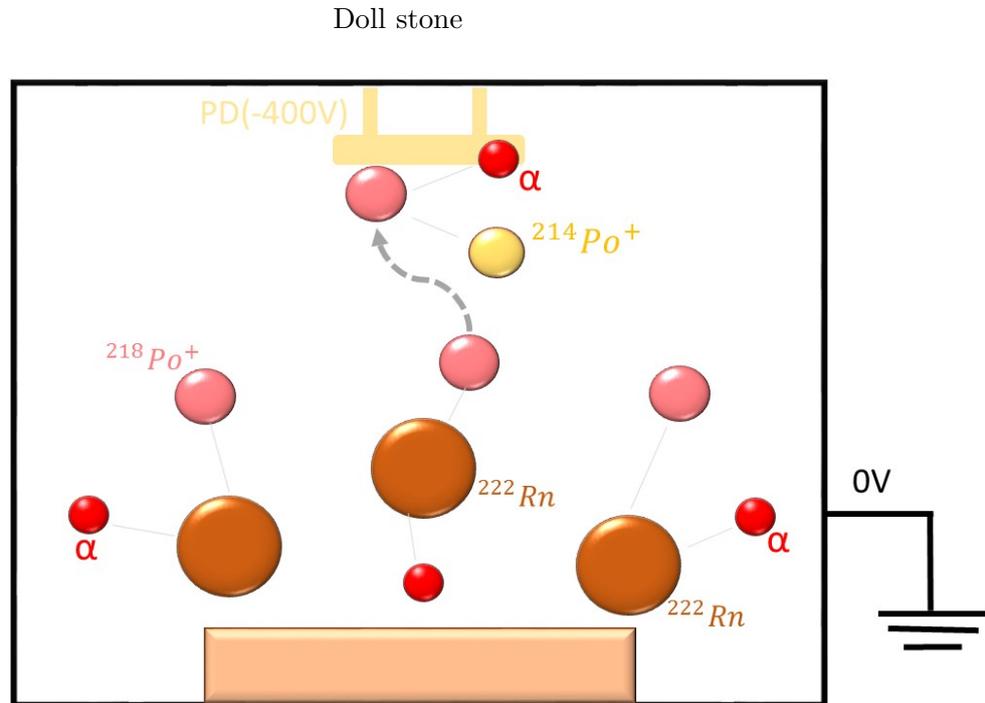


図 2.6 検出器内の様子

2.3 フォトダイオードに流れる電流

シリコンでは一個の電子-正孔対を作るには約 $3.6(\text{eV})$ のエネルギーが必要であると知られている。ウラン系列の ^{218}Po が α 崩壊するとき $5.998(\text{MeV})$ の α 線を放出する。この時に作られる電子-正孔対の数は

$$5.998 \times 10^6 (\text{eV}) / 3.6 (\text{eV}) = 1.7 \times 10^6 \text{個} \quad (2.1)$$

である。電子1個がもつ電荷の絶対値は $1.602 \times 10^{-19} \text{C}$ なので

$$1.67 \times 10^6 (\text{個}) \times 1.602 \times 10^{-19} (\text{C}) = 2.7 \times 10^{-13} (\text{C}) \quad (2.2)$$

の電気量を持つことになる。 $I = dQ/dt$ より、 $t = 1 \mu\text{s}$ の時に流れる電流は

$$2.7 \times 10^{-13} (\text{C}) / 10^{-6} = 2.7 \times 10^{-7} (\text{A}) \quad (2.3)$$

となる。この値からわかるとおり、一つの α 線がフォトダイオードに入射した際に流れる電流はとても微量で、増幅しないとみることもできないことがわかる。

第3章

ラドン検出器

ラドン検出器は、ラドンガスを封入し、ラドンの娘核である Po が崩壊したときに生じる α 線を PD で検出しアナログ部へ信号を送るラドン検出部と、ラドン検出部から来た信号を整形・増幅するアナログ部、整形・増幅された信号のデータを制御し、PC に送りデータを保存するデータ制御・収集部の三つで出来ている。

3.1 ラドン検出部

今回の実験では、ラドン検出部として先行実験の缶、小缶、中缶、大缶の四種類の大きさの缶を使用した。缶の形は円筒形であり、先行実験の缶はステンレス、小缶はブリキ、中缶と大缶は電気メッキブリキで出来ている。以下の図 3.1 にそれぞれの缶の外観、表 3.1 にそれぞれの缶の大きさを示す。



図 3.1 ラドン検出器

表 3.1 ラドン検出器のサイズ

	先行実験の缶	小缶	中缶	大缶
高さ [cm]	13.0	13.0	16.3	21.5
外径 [cm]	5.5	5.0	7.1	8.4

α 線を検出するPDはラドン検出部の缶の蓋の内側に取り付けられており、蓋の反対側にプリアンプを取り付けることにより、プリアンプはPDからの電気信号を受け取ることができ、なおかつPDは逆バイアスがかかり静電捕集するため表面へ負の高電圧がかかるようになっている。

3.2 アナログ部

アナログ部での一連の流れは次のようになっており、以下の図 3.2 にまとめたものを示す。PDで α 線を検出すると、まずプリアンプで電荷信号を増幅、電圧信号に変換して出力し、シェイパースタンプでプリアンプからの出力信号を波形整形し、その信号を受けて比較回路がArduinoへTrigger信号を送る。Trigger信号を受け取ったArduinoはGate信号をサンプル&ホールド回路に送り、Gate信号を受け取ったサンプル&ホールド回路はその時の入力電圧値を保持してArduinoに送る。サンプル&ホールド回路から波高データを受け取ったArduinoはそのデータをPCのProcessingに送る。最後にArduinoがリセット用のGate信号をサンプル&ホールド回路に送る。

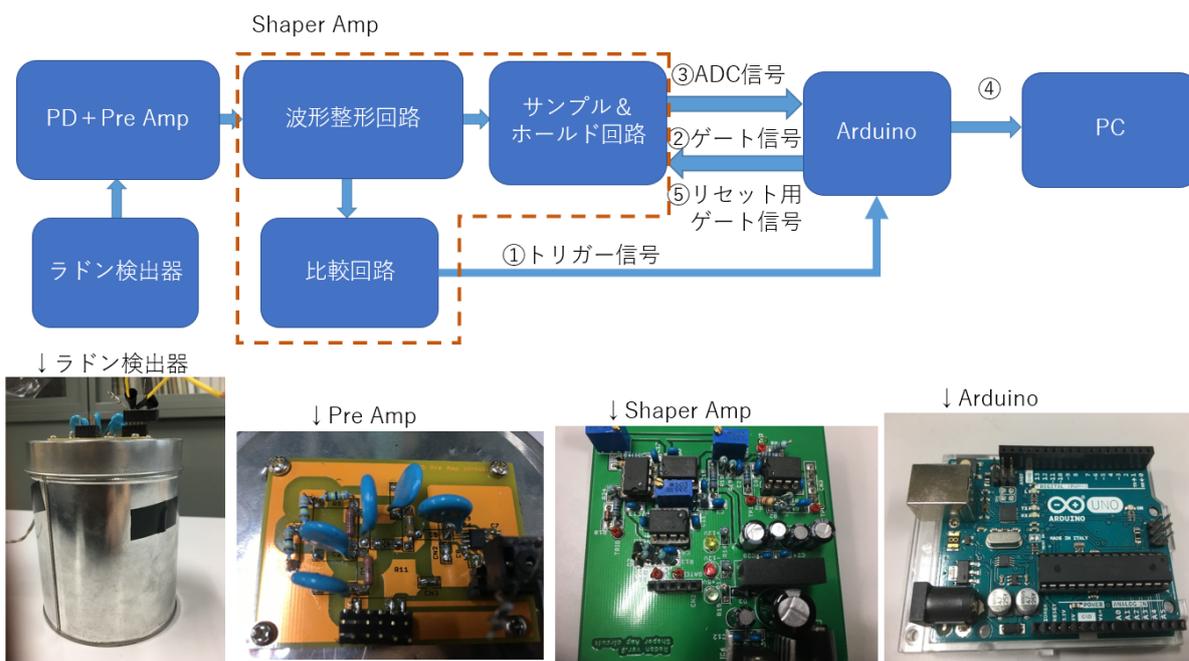


図 3.2 アナログ部の流れ

3.2.1 プリアンプ

今回の実験で使用したプリアンプの回路図を以下の図 3.3 に示す。

Pre Amp 回路図

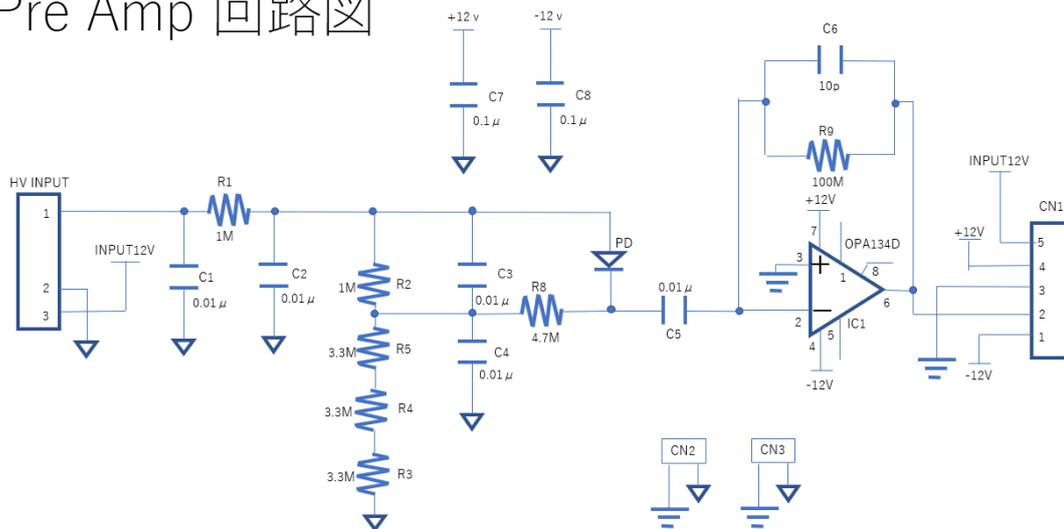


図 3.3 プリアンプの回路図

図 3.3 の中央部分に PD を取り付ける場所があり、PD からの電気信号は回路図の右側の電荷増幅器へと流れるようになっている。回路図の CN1 はシェイパーアンプとの接続部分であり、電荷増幅器で増幅され電圧信号へと変換された信号がシェイパーアンプへと出力されるようになっている。以下の四角で囲まれた部分に電荷増幅器の簡単な説明を記す。

～電荷増幅器 [10]～

電荷信号を増幅し、電圧信号にする。
出力される信号の波高、時定数は使用した抵抗とコンデンサの値によって次のように決まる。

波高

$$V_{out} = \frac{Q}{C_f} \quad (3.1)$$

時定数

$$t = R_f C_f \quad (3.2)$$

図 3.4 電荷増幅器

第 2.3 節で話した通り、 ^{218}Po が α 崩壊したときの α 線を PD で検出したときに作られる電気量は、 $Q_{in} = 2.7 \times 10^{-13} (C)$ である。よって、式 3.1 より、シェイパーアンプへ出力され

る電荷増幅器後の信号の電圧 V_{out} は、

$$V_{out} = Q_{in}/C_f = 2.7 \times 10^{-13}/10^{-11} = 27 (mV) \quad (3.3)$$

となる。

3.2.2 シェイパーアンプ

今回の実験で使用したシェイパーアンプの回路図、波形整形回路の配置をそれぞれ以下の図 3.5、図 3.6 に示す。

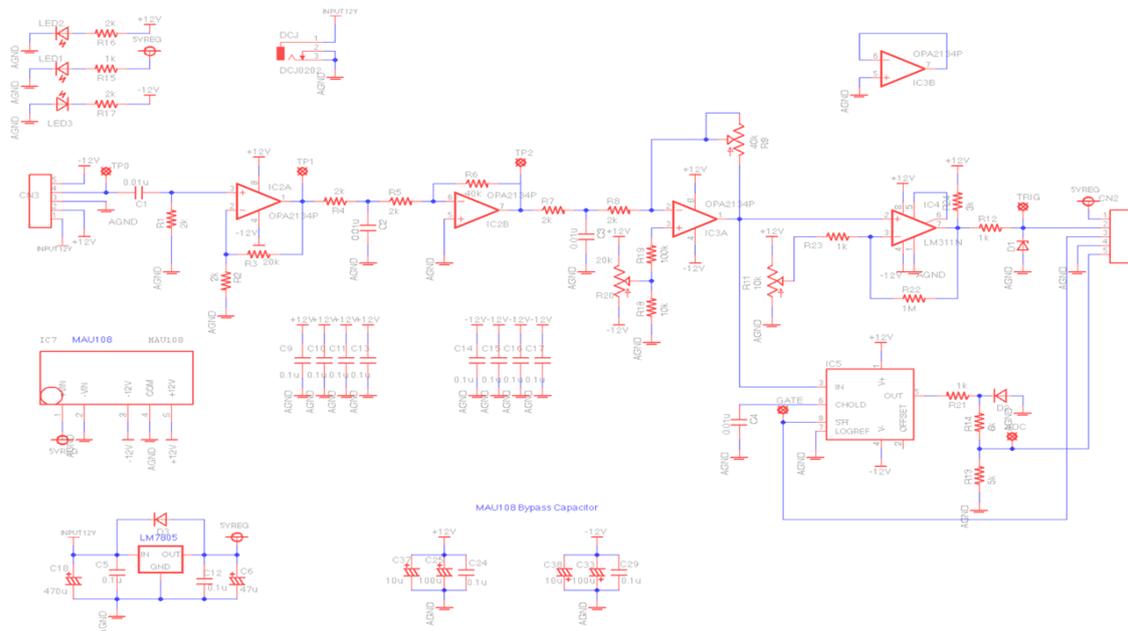


図 3.5 シェイパーアンプの回路図 [10]

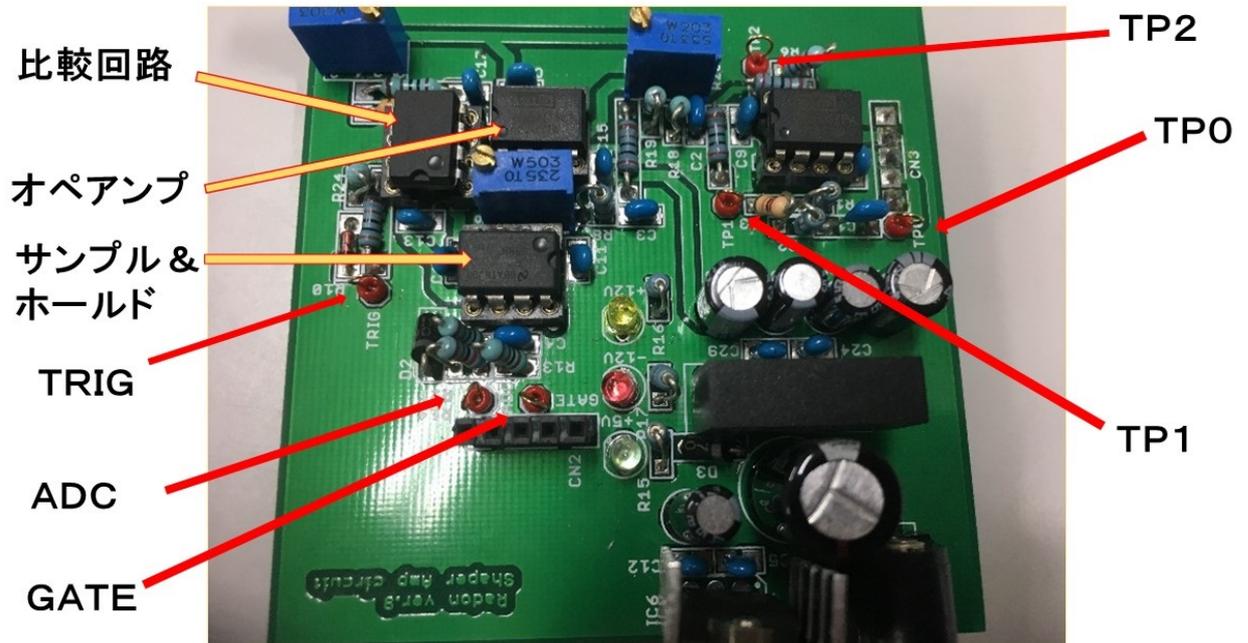


図 3.6 波形整形回路の配置

図 3.5 の CN3 はプリアンプとの接続部分であり、プリアンプから出力された信号 ($TP0$) はまずハイパスフィルターを通り低周波成分のノイズが減衰され、次に非反転増幅回路で増幅される ($TP1$)。その後ローパスフィルターを通り高周波成分のノイズが減衰され、次に反転増幅回路を通り信号の正負が逆転し増幅される ($TP2$)。その後もう一度ローパスフィルターを通り反転増幅回路を通る。こうして整形された信号は比較回路を通り、信号の電圧が閾値電圧より高ければ比較回路から Arduino へトリガー信号が送られる。図 3.5 の CN2 は Arduino との接続部分である。その後の流れはこの章の最初に記している通りである。後ろに () が記されている時の信号はオシロスコープで回路図の () 内に記されたところの信号を見ることで確認することができる。以下の四角で囲まれた部分にハイパスフィルター・ローパスフィルター、非反転増幅回路、反転増幅回路、比較回路、サンプル&ホールド回路の簡単な説明を記す。

ハイパスフィルターとローパスフィルター [10][11][12]

プリアンプから出力された信号に含まれる様々な周波数のノイズを取り除く。遮断周波数、時定数は使用した抵抗とコンデンサの値によって次のように決まる。

遮断周波数

$$f = \frac{1}{2\pi RC} \quad (3.4)$$

時定数

$$t = RC \quad (3.5)$$

遮断周波数とは出力信号が入力信号の半分になる周波数のことであり、 C と R のインピーダンスが等しくなる時の周波数である。

～ハイパスフィルター～

C のインピーダンスが $1/(2\pi fC)$ で f の増大とともに小さくなるので、高域通過型の回路となる。微分回路の一つである。

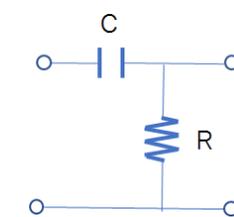


図 3.7 ハイパスフィルター

～ローパスフィルター～

高い周波数の成分はコンデンサ側に流れていくため、低域通過型の回路となる。積分回路の一つである。

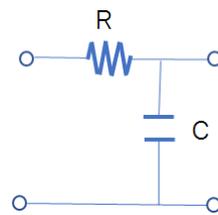


図 3.8 ローパスフィルター

～非反転増幅回路 [10][13]～

位相は同位相のまま、入力信号を増幅する。入力インピーダンスが極めて大きい。出力される信号の波高は使用した抵抗の値によって次のように決まる。

波高

$$V_{out} = \left(1 + \frac{R_2}{R_1}\right) V_{in} \quad (3.6)$$

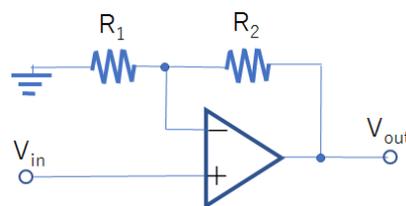


図 3.9 非反転増幅回路

～反転増幅回路 [10][13]～

位相を逆位相にして、入力信号を増幅する。出力される信号の波高は使用した抵抗の値によって次のように決まる。

波高

$$V_{out} = -\frac{R_2}{R_1} V_{in} \quad (3.7)$$

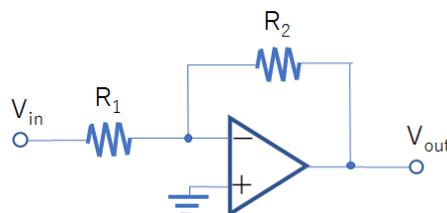


図 3.10 反転増幅回路

～比較回路 [10][13][14][15]～

閾値電圧より高い α 線の信号がくるとトリガー信号を出力する。

V_{in-} に閾値電圧が設定されており、入力信号の V_{in+} が閾値を上回れば正の最大電圧を出力する。

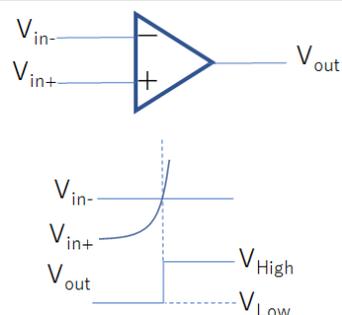


図 3.11 比較回路

～サンプル&ホールド回路 [10][13]～

ある期間だけ信号を一定値に保つホールド機能を持つ回路。

今回の実験では、保持する期間やどの値を保持するかは Arduino によって制御されている。Arduino から Gate 信号がくると、その時の入力電圧値を保持し、もう一度リセット用の Gate 信号がくるまで保持した入力電圧値を Arduino に送り続ける。

3.2.3 各信号の波形

以下の図はラドンガス測定時の各信号の波形をオシロスコープで確認したものである。図 3.12 は上から順に TP0,TP1,TP2,TP3 の波形を、図 3.13 は上から順にトリガー,TP3,ADC,Gate の波形を示している。ここで、TP3 はシェイパーアンプの二つ目の反転増幅回路を通った直後の信号である。

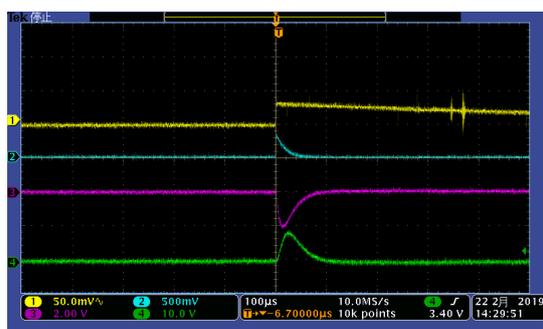


図 3.12 TP0,TP1,TP2,TP3 の波形

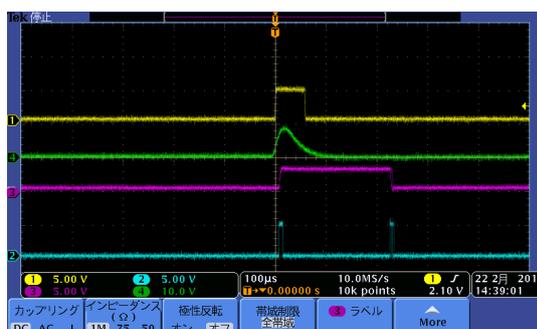


図 3.13 トリガー,TP3,Gate,ADC の波形

図 3.12 を見ると、TP0 の信号の値は約 30mV であり、TP1 の信号は約 330mV、TP2 の信号は約 2V、TP3 の信号は約 8V になっている。TP1 から TP2 への増幅率が約 6 倍であり、反転増幅回路の増幅度よりも低い増幅度となっているが、これは反転増幅回路の前にローパスフィルターがあり、その抵抗の部分で電圧降下を起こしているためだと考えられる。

図 3.13 を見ると TP3 の信号の比較回路の閾値を超えた部分でトリガー信号が出ており、トリガー信号が出力されたタイミングで Gate 信号が出て、Gate 信号に挟まれた部分で ADC 信号が出力されていることが分かる。ADC で保持される値は一つ目の Gate 信号の立ち上がり時の TP3 の信号の値であり、TP3 の信号のピーク時の値を保持するように一つ目の Gate 信

号の幅を決めているが、出力された ADC 信号の値は TP3 の信号のピーク時の値と比べて小さくなっている。これはサンプル&ホールド回路の出力から Arduino へ ADC 信号が送られる間に抵抗が挟まっており、この抵抗によってサンプル&ホールド回路の出力を Arduino の規格の 0~5V の範囲になるようにしているためである。

3.3 データ制御・収集部

3.3.1 Arduino

Arduino はハードウェアのマイコンボードとソフトウェアの統合開発環境 (IDE: Integrated Development Environment) の二つのことを指す。図 3.14 に実際に使用したボード、図 3.15 にソフトウェアを示す。オープンソースハードウェアでマイコンボードに組み込むためのソフトウェアは無償でネットからダウンロードして使用することができる。Arduino に電子部品などをつなげ、IDE にプログラムを書き込み、そのプログラムに沿って、電子部品を動かすことができ、シリアル通信によって PC との通信もできる。Arduino のボードは 6 チャンネルの 10bit AD コンバーターを搭載しており、0V~5V の入力電圧を 0 から 1023 の数値に変換することができるが、シリアル通信を使う際、送れるデータ量は 8bit なので、10bit のデータを一旦 8bit と 8bit に分け、データを PC に送信している。図 3.16 に Arduino の仕様、図 3.17 に Arduino との配線、図 3.18 に Arduino 行う波高分析手順を示す。



図 3.14 Arduino のボード



図 3.15 Arduino のソフトウェア (IDE)

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

図 3.16 Arduino Uno R3 の仕様

Arduino	波形整形回路
GND	GND
Analog 0番	ADC
Digital 2番	トリガー信号
Digital 7番	ゲート信号

図 3.17 Arduino との配線



図 3.18 Arduino が行う波高分析手順

3.3.2 processing

processing とは電子アートとビジュアルデザインのためのプログラム言語であり、統合開発環境 (Integrated Development Environment) を指す。このソフトウェアは無料でダウンロードすることができる。java をベースとしており、視覚的なフィードバックが即座にえられ、初心者にも扱いやすいプログラミング言語である。また、電子スケッチブックの基盤としても利用できる。図 3.19 は processing の IDE である。

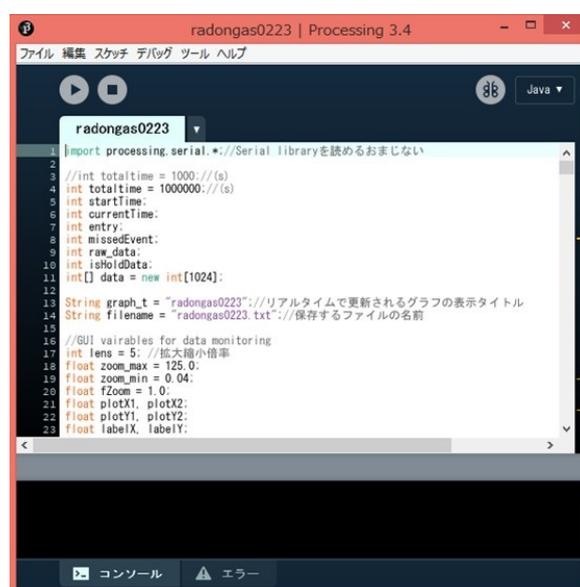


図 3.19 processing の IDE

データ保存

本研究では、processing を使い、Arduino からのデータを視覚化とデータの変換、保存をした。Processing は Arduino から送られてきた二つの 8bit のデータを受け取り、それら二つを 10bit に変換、波高データと時間情報の二つをファイルに書き込むという作業を、自分が指定した時間まで繰り返し、指定した時間が過ぎるとデータを保存終了する。Arduino から Processing にデータを送る際には、通信の信頼性を確保するために各 8bit のデータの後ろの 2bit にパリティビットを設定した。測定を行っている際には、図 3.21 のようにオンラインモニターを見ることができる。縦軸がカウント数で、横軸は 0V~5V を 1024ch に変換したものである。Processing のプログラムは付録 A に載せている。

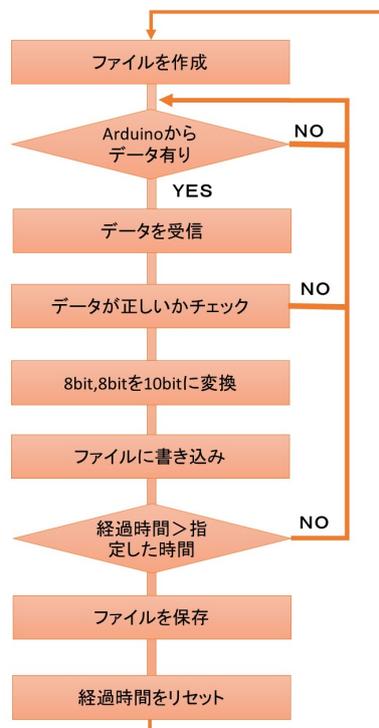


図 3.20 processing が行う波高手順

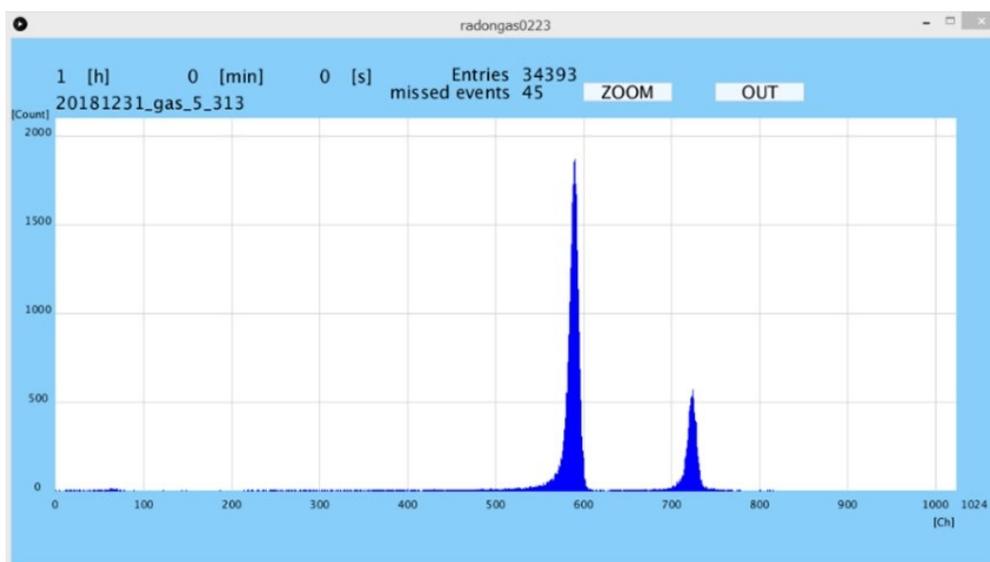


図 3.21 オンラインモニタリングの様子 (縦軸:カウント数 横軸:パルスの高さ)

第4章

実験

4.1 ^{241}Am 線源を用いた α 線測定

4.1.1 実験内容

ラドン検出器を使って、 α 線の測定をできるかを確認するために、またエネルギーとチャンネルの関係を調べるために、 ^{241}Am (5.4MeV) を用いて実験を行った。

4.1.2 実験方法

今回使用した ^{241}Am 線源は、線源から出てくる α 線の頻度が高いので、図 4.1 のように、穴をあけた紙を線源につけ、 α 線の頻度を低くした。また、エネルギーとチャンネルの関係を調べるために、 ^{241}Am 線源から出てくる α 線が、空気と相互作用し、エネルギー損失をするのを少なくするために、図 4.2 のように、出来るだけ ^{241}Am 線源とフォトダイオードを近づけ、この状態で5分間測定を行った。また、Arduino に書き込むプログラムを変えることにより、信号が来ている状態で、0MeV に対応する ch も調べることができる。この時にオシロスコープで見える波形の図を図 4.3 に示した。



図 4.1 穴の開けた紙をつけた ^{241}Am 線源

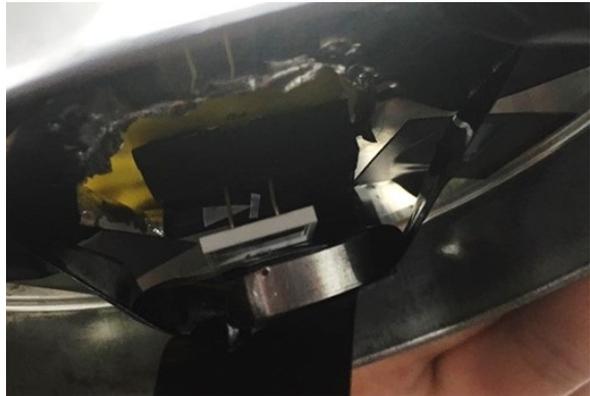


図 4.2 フォトダイオードに出来るだけ近づけた ^{241}Am 線源

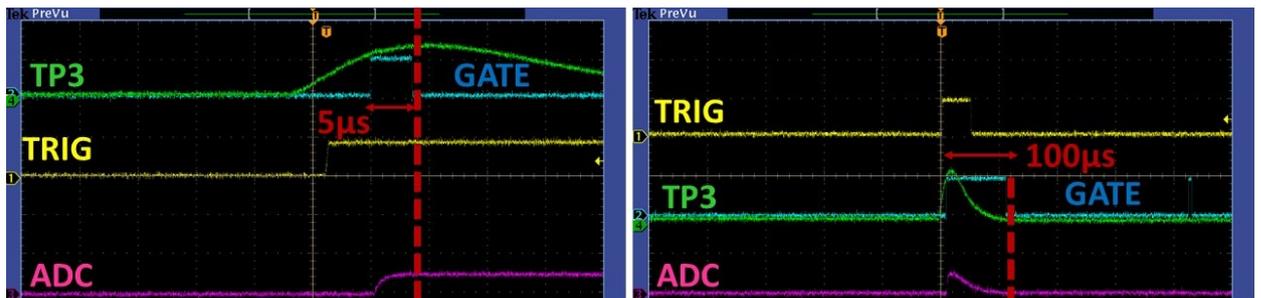


図 4.3 オシロスコープでみた波形

ラドン検出器では一つ目の GATE 信号の立ち下がり時に波形整形された後の信号 (TP3) の値を読み込んでいる。よって、一つ目の GATE 幅を $5\ \mu\text{s}$ に設定すると、TP 3 のピーク時の値を読み込み、一つ目の GATE 幅を $100\ \mu\text{s}$ に設定すると、0 点時の値を読み込むことができる。図 4.3 の左では Am の α 線のエネルギーを、図 4.3 の右では 0 エネルギーをそれぞれ測定時のオシロスコープの各テストピンでの出力画面を表している。

4.1.3 結果と考察

図 4.4 は、横軸が PulseHeight[ch]、縦軸がカウント数で、ピンクのピークが GATE 幅 100 μ s で実験を行った結果、青いピークが GATE 幅 5 μ s で実験を行った結果で、0 MeV に対応するのが 50.61ch 付近のピーク、 ^{241}Am 線源の α 線に対応するのが 526.2ch 付近のピークだということがわかった。これにより、 ^{241}Am 線源の α 線を検出できたことがわかる。

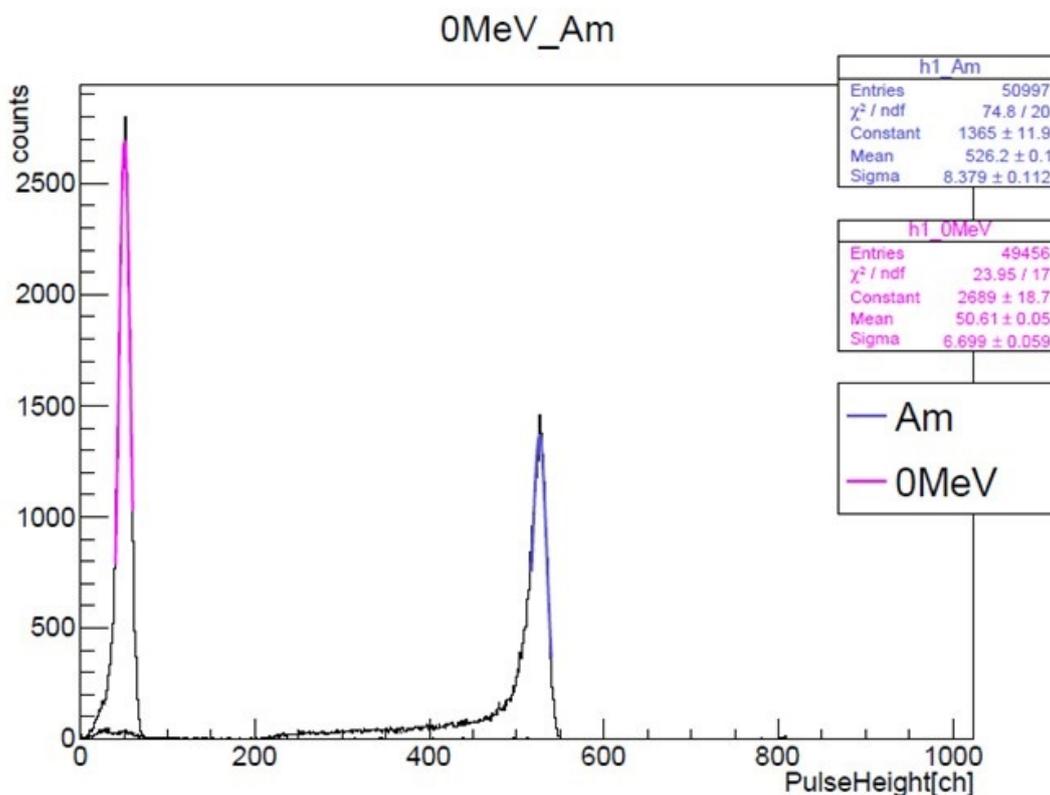


図 4.4 実験結果の波高分布

4.1.4 ベーテプロットホの式

^{241}Am の線源から出てくる α 線源が、密封線源の薄膜でいくらかエネルギー損失をしていると仮定し、ベーテプロットホの式を使いエネルギー損失の値を計算した。ベーテプロットホの式は次式で表される。

$$\frac{-dE}{dx} = 4\pi N_a r_e^2 m_e c^2 \rho \frac{Z}{A} \frac{z^2}{\beta^2} \left[\ln \frac{2m_e \gamma^2 v^2 W_{max}}{I^2} - 2\beta^2 \right] [\text{MeV/cm}] \quad (4.1)$$

$$\text{ここで、} 4\pi N_a r_e^2 m_e c^2 = 0.3070 [\text{MeV cm}^2/\text{g}] \quad (4.2)$$

とした。また

$$\begin{aligned}
 r_e &: \text{古典電子半径} = 2.817 \times 10^{-13} \text{ cm} \\
 m_e &: \text{電子質量} = 0.511 \text{ MeV}/c^2 \\
 N_a &: \text{アボガドロ数} = 6.022 \times 10^{23} \text{ mol}^{-1} \\
 I &: \text{物質の平均イオン化ポテンシャル} \\
 Z &: \text{物質の原子番号} \\
 A &: \text{物質の原子質量} \\
 \rho &: \text{物質の密度} \\
 z &: \text{入射粒子の電荷} \\
 \beta &= v/c: \text{入射粒子の速度} \\
 \gamma &= 1/\sqrt{1-\beta^2}: \text{入射粒子の速度} \\
 W_{(max)} &: \text{一回の衝突で電子が得る最大エネルギー}
 \end{aligned} \tag{4.3}$$

今回、吸収体を Au として、 $\rho = 19.32 \text{ [g/cm}^3]$ 、 $Z=79$ 、 $A=196.97$ 、 α 線の各種にもよるが $v=(1.5\sim 2.0)^7 \text{ [m/s]}$ で計算を行うと

$$\frac{-dE}{dx} = 8638.02 \text{ [MeV/cm]} \tag{4.4}$$

となった。密封線源の窓が厚さ $1\mu\text{ m}$ の時

$$\frac{-dE}{dx} = 8638.02 \text{ [MeV/cm]} \times 1 \times 10^{-4} \text{ [cm]} = 0.09 \text{ [MeV]} \tag{4.5}$$

これらの計算から、 ^{241}Am 線源から放出される α 線は、密封線源の膜で 0.09 [MeV] エネルギー損失をしていると仮定した。

4.1.5 ^{241}Am によるエネルギー較正直線

^{241}Am 線源を用いた実験結果を使い、エネルギーとチャンネルの関係において線形性があると仮定をし、エネルギー較正直線を引いた図を、図 4.5 に示した。この時、前節の計算から、 ^{241}Am 線源から放出される α 線のエネルギーの文献値は 5.4 [MeV] だが、 5.31 [MeV] とした。また、0 点に対応するチャンネルは 50.61 ch と測定したが、エネルギー較正直線を引く際、0 点を 0 ch に合わせて、直線を引いた。

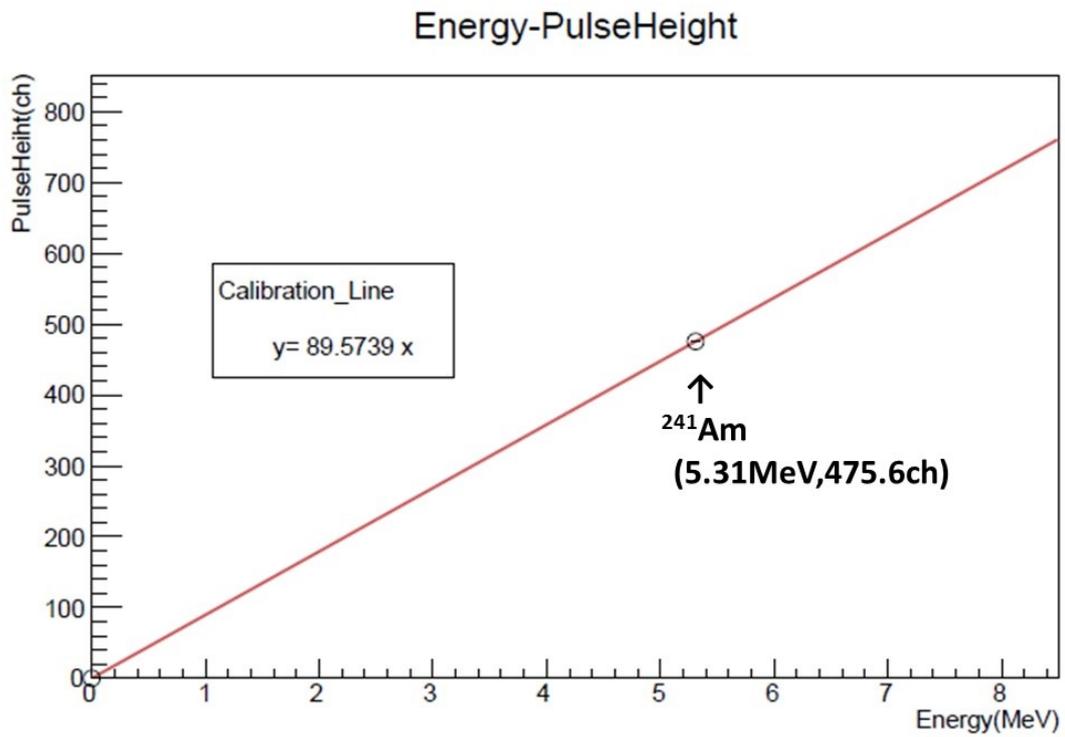


図 4.5 ²⁴¹Am によるエネルギー較正直線

表 4.1 エネルギーと ch の関係

	x(MeV)	y(ch)	x error	y error
0 点	0.0	0.0	0.0	0.1
²⁴¹ Am	5.31	475.6	0.0	0.1

上記から、1(MeV) に相当する ch は 89.6ch ということがわかる。

4.2 ラドンガス測定

ラドン検出器内に、天然のウランを含む鉱石 (Doll stone) を入れ、ラドンガスを充満させ、測定開始前に、Dollstone を取り出し実験をした。Doll stone とは微量の天然ウランを含む岩石・土砂を原料として、製造された横 100mm、縦 100mm、奥行きが 13mm の素焼きタイルで、天然のウランがおよそ 0.05 % ほど含まれており、一枚当たり 1 ~ 2 (Bq/h) のラドンが発生する。[16] このことから、この Dollstone は純粋なラドン石ではないので、測定開始前に取り出している。



図 4.6 Doll stone

4.2.1 ラドンガスの実験結果

図 4.7 のように、588.47ch 付近に一つ目のピーク、723.88ch 付近に二つ目のピークをみることができた。

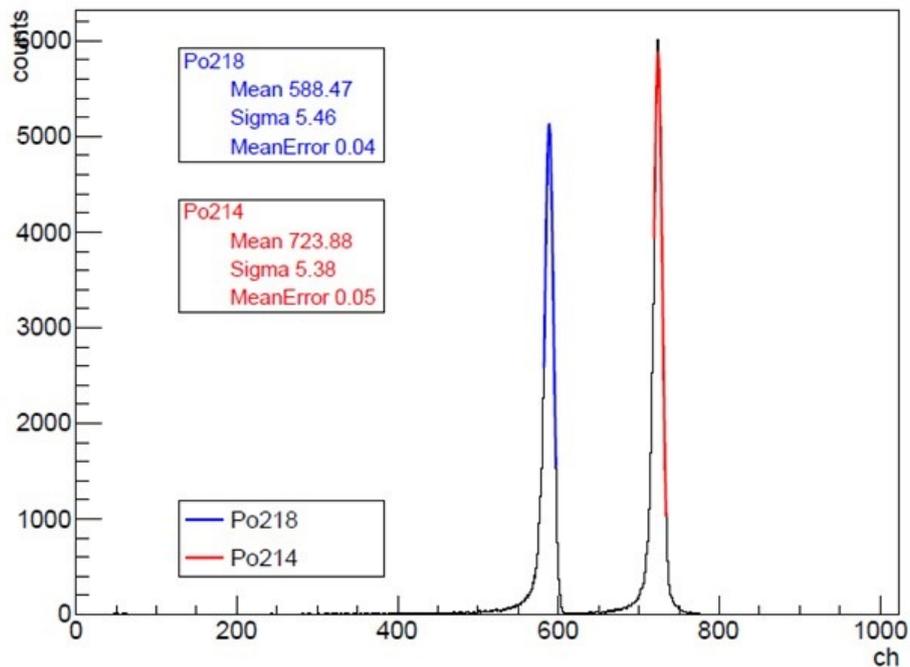


図 4.7 ラドンガスの波高分布

4.2.2 実験の考察

得られたピークを、較正直線から得られた値を使い、エネルギーに変換すると 588.47ch 付近に見えていたピークは、 6.0051 ± 0.0014 [MeV] となり、723.88ch 付近に見えていたピークは 7.5170 ± 0.0009 [MeV] となった。また、ラドンが属する崩壊系列の中でも一番存在比が高いウラン系列中の Po だと考え、得られたエネルギーに近い値を探すと、 ^{218}Po 、 ^{214}Po の文献値の値がそれぞれ、5.998 [MeV]、7.69 [MeV] なので、これら二つのピークがそれぞれ ^{218}Po 、 ^{214}Po だと断定した。

実際に、横軸の座標に、文献値のエネルギー、縦軸に実験結果の ch 数をあてはめ、 ^{218}Po 、 ^{214}Po の点を較正直線上にプロットをした図が、図 4.8 である。

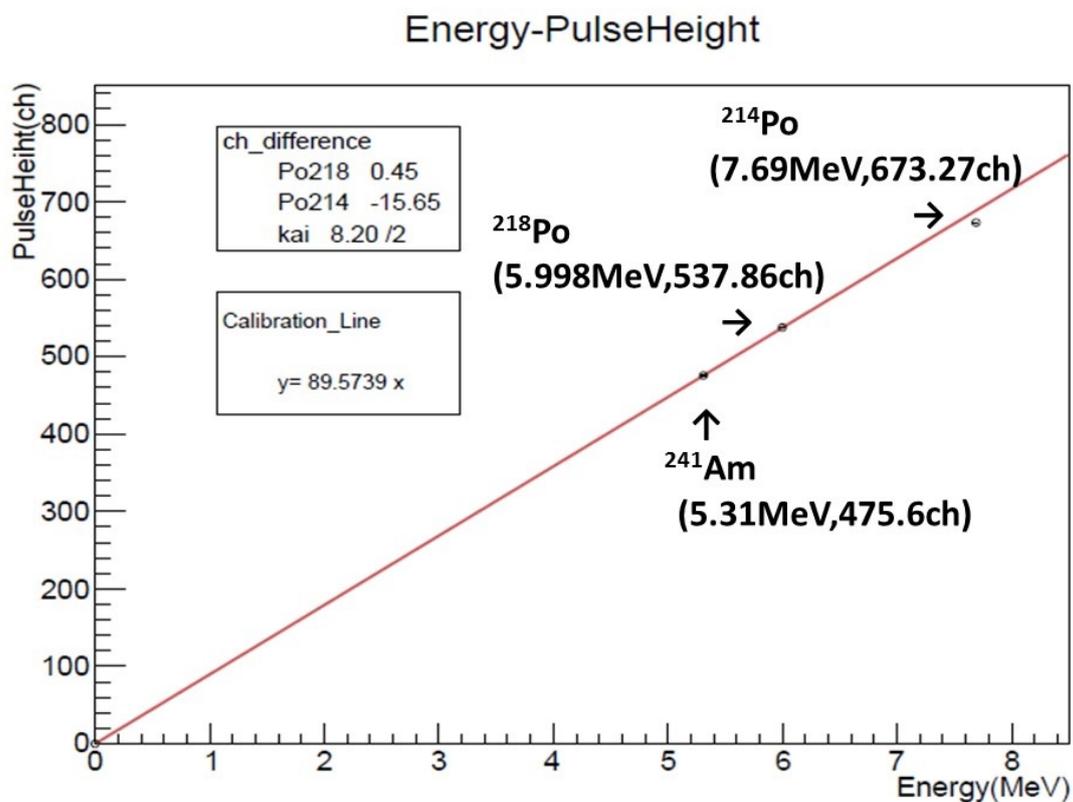


図 4.8 ^{241}Am によるエネルギー較正直線に ^{218}Po 、 ^{214}Po の点をプロットした

図 4.8 から、 ^{214}Po の点が直線から少しずれているのは、今回の実験では静電捕集法によって PD 表面に Po を吸着させることで、Po が α 崩壊したときの α 線のエネルギーを測定しているが、静電捕集法で PD 表面に吸着していた ^{218}Po が崩壊し、 $^{214}\text{Pb}^{214}\text{Bi}$ に崩壊しているときに、PD 表面に吸着していたものがはがれ、 ^{214}Po に崩壊してまた PD 表面にひきつけられるが、 ^{214}Po の半減期はととても短いため、PD 表面に向かっている段階で α 崩壊し、PD 表面との間の空気と相互作用し、エネルギー損失しているためだと考えられる。

4.3 半減期の導出

ラドンガスの測定で得られた α 線のエネルギーとそれを検出した時間のデータを用いることで半減期を求めることができ、今回は波高分布のヒストグラムの ^{218}Po と ^{214}Po のピークの Mean から 2σ 以内に入っているデータをそれぞれ ^{218}Po と ^{214}Po のデータとし、それらが単位時間当たりについていくつ検出されているかを調べることで半減期の導出をしている。 ^{222}Rn の崩壊系列における ^{218}Po と ^{214}Po は ^{222}Rn と永続平衡の関係になっているので、求めた半減期が ^{222}Rn の半減期とどの程度一致しているかがラドン検出器の性能評価となる。

^{222}Rn の崩壊系列における各核種はそれぞれ以下の微分方程式に従って連鎖的に崩壊する。

$$^{222}\text{Rn} : -dN_1 = -\lambda_1 N_1 dt \quad (4.6)$$

$$^{218}\text{Po} : -dN_2 = (\lambda_1 N_1 - \lambda_2 N_2) dt \quad (4.7)$$

$$^{214}\text{Pb} : -dN_3 = (\lambda_2 N_2 - \lambda_3 N_3) dt \quad (4.8)$$

$$^{214}\text{Bi} : -dN_4 = (\lambda_3 N_3 - \lambda_4 N_4) dt \quad (4.9)$$

$$^{214}\text{Po} : -dN_5 = (\lambda_4 N_4 - \lambda_5 N_5) dt \quad (4.10)$$

ここで、 N_1, N_2, N_3, N_4, N_5 はそれぞれ $^{222}\text{Rn}, ^{218}\text{Po}, ^{214}\text{Pb}, ^{214}\text{Bi}, ^{214}\text{Po}$ の原子数であり、 $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$ はそれぞれ $^{222}\text{Rn}, ^{218}\text{Po}, ^{214}\text{Pb}, ^{214}\text{Bi}, ^{214}\text{Po}$ の崩壊定数、 t は経過時間を表している。第 1.4 節 1.7 式より半減期と崩壊定数は $\lambda T_{1/2} = \log_e 2$ と表されるので、崩壊定数が分かれば半減期を求めることができる。 ^{222}Rn の半減期は ^{218}Po の半減期と比べて十分大きいので、 ^{222}Rn と ^{218}Po は永続平衡の関係となっており、 ^{218}Po の半減期よりも十分大きなスケールでは、 ^{218}Po の減少の仕方は ^{222}Rn の半減期に従って減少すると考えられる。同様に考えて、 ^{222}Rn の半減期は $^{214}\text{Pb}, ^{214}\text{Bi}, ^{214}\text{Po}$ の半減期と比べても十分大きいので、 ^{214}Po の半減期よりも十分大きなスケールでは、 ^{214}Po の減少の仕方は ^{222}Rn の半減期に従って減少すると考えられる。

式 4.6 の微分方程式を解いて ^{222}Rn の単位時間当たりの崩壊数を表す式は

$$\lambda_1 N_1(t) = N_1(0) \lambda_1 e^{-\lambda_1 t} \quad (4.11)$$

と求まるので、 ^{218}Po と ^{214}Po のデータ点を指数関数 $\exp(p_0 + p_1 \times x)$ で fit を行うことで、fit 線の傾きの大きさとして崩壊定数 λ が求められる。以下に先行実験の缶を用いてラドンガスを測定したときの波高分布を図 4.9、その時の測定データを fit した結果を図 4.10 に示す。ここで、図 4.10 の横軸は測定を開始してからの経過時間、縦軸はそれぞれのピークの Mean から 2σ 以内に入っているデータの 10 分ごとのカウント数である。また、ラドンガスは DOLLSTONE を缶の中に約 50 時間入れることで封入し、実験の測定は 8 時間行った。

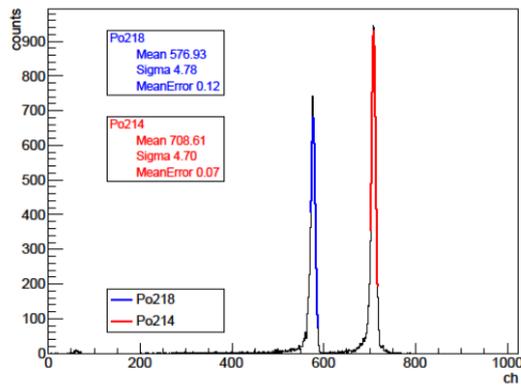


図 4.9 先行実験の缶を用いた時のラドンガスの波高分布

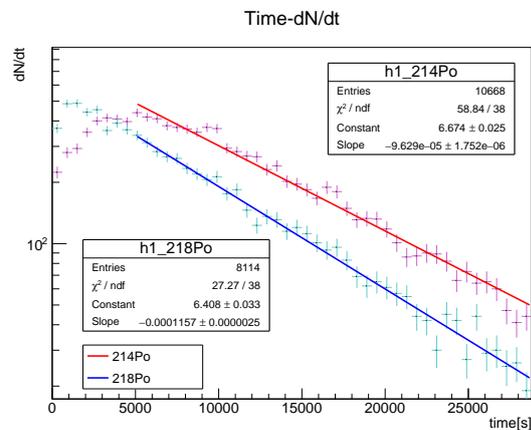


図 4.10 図 4.9 のデータを指数関数で fit した結果

fitにより得られた崩壊定数 λ と、それを用いて求めた半減期、 ^{222}Rn の半減期の文献値をまとめた表を以下の表4.2に示す。

表4.2 図4.10のfit結果

	^{218}Po のfit結果	^{214}Po のfit結果	文献値
崩壊定数 λ	$1.157 \times 10^{-4} \pm 2.5 \times 10^{-6}$	$9.629 \times 10^{-5} \pm 1.752 \times 10^{-6}$	2.09×10^{-6}
半減期 [day]	0.069 ± 0.001	0.083 ± 0.002	3.82

表4.2より、先行実験での缶を用いて測定した結果は ^{222}Rn の半減期と大きく離れた結果となっていることが分かる。その原因として、缶の密封度や体積、実験時の温度や湿度を考え、本研究では改良としてまず缶の密封度を高めることにした。

4.4 小型ラドン検出器の改良

4.4.1 改良方法

密封度を高める方法として、小型ラドン検出器に用いる缶の蓋の構造をより密封度の高い構造へと変更することにし、本研究では主に2種類の構造の蓋を作成した。一つ目は、缶の蓋にプリアンプの装着や、プリアンプとPDの接続部分を通すための穴を、先行実験でははさみなどを用いて開けていたのに対し電気ドリルを用いて開けることで缶の蓋に開く穴を先行実験の蓋よりも小さくした構造となっており、二つ目はプリアンプをテープを用いて装着するようにし、また、プリアンプとPDを接続するコネクタを缶の蓋に取り付けることで缶の蓋に開く穴を一つ目の構造の蓋よりも小さくした構造となっている。以下の図4.11、図4.12、図4.13にそれぞれの缶の蓋の外観を示す。写真ではそれぞれの蓋は同じ大きさに見えるが、図4.12は大缶の蓋であり、図4.13は小缶の蓋であるので、蓋の大きさは図4.12>図4.11>図4.13の順である。



図4.11 先行実験の缶の蓋

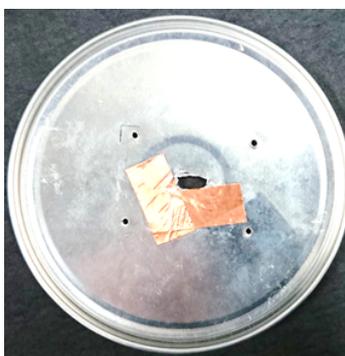


図4.12 一つ目の工夫の蓋



図4.13 二つ目の工夫の蓋

図4.11、図4.12の真ん中に開いている穴がプリアンプとPDの接続部分を通すための穴であり、その周りに開いている4つの穴がプリアンプを取り付けるためのねじ穴である。図4.11

の真ん中の穴の大きさは穴を長方形に見立てると縦が約 2.8cm 、横が約 3.6cm であり、その周りの4つの穴は直径約 5mm である。一方で、図 4.12 の真ん中の穴の大きさは穴を長方形に見立てると縦が約 7mm 、横が約 20mm であり、その周りの4つの穴は直径約 3mm である。よって、プリアンプを取り付けるための穴を開けるといふ点で先行実験の缶の蓋と一つ目の工夫をした蓋は同じ構造をしているが、穴を開けるときに電気ドリルを用いることで穴の大きさを小さくすることができたので、その分缶の密封度は高まっていると考えられる。先行実験の缶の蓋にプリアンプを取り付けた時の外観を以下の図 4.14、図 4.15 に示す。



図 4.14 プリアンプ装着時 (表)



図 4.15 プリアンプ装着時 (裏)

二つ目の工夫の蓋は先行実験での缶の蓋や一つ目の工夫の蓋とは全く違う構造をしており、密封度を高めるため、必要最低限の穴しか開けていない構造となっている。まず缶にあける穴は図 4.16 のように直径約 2mm の穴二つのみであり、そこにプリアンプと PD を接続するための図 4.17 のようなコネクタを取り付ける際にコネクタと缶の蓋で通電しないように穴に絶縁テープを貼っており、接着剤を用いてコネクタを取り付け、穴をふさいだ構造となっている。二つ目の工夫の蓋のコネクタ部分を拡大した最終的な外観を図 4.18 に示す。蓋に開ける穴は先行実験での缶の蓋や一つ目の工夫の蓋と比べてとても小さく、最終的に接着剤で穴をふさぐため、二つ目の工夫の蓋が構造的に一番密封度が高いと考えられる。



図 4.16 開ける穴の大きさ



図 4.17 使用するコネクタ



図 4.18 最終的な完成図 : 表 (左) 裏 (右)

二つ目の工夫の蓋は、蓋にあける穴を少なくするためプリアンプを取り付けるためのねじ穴を開けていないので、プリアンプの取り付けはテープを用いて行っており、先行実験での缶の蓋や一つ目の工夫の蓋では取り付けの際に使うねじの一つがプリアンプのグランドを缶と接続

する役割を果たしていたが、二つ目の工夫の蓋ではワニ口クリップを用いることでプリアンプのグラウンドを缶と接続している。プリアンプを取り付けた時の外観を図 4.19、図 4.20 に示す。

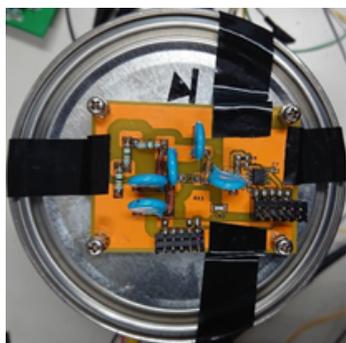


図 4.19 プリアンプの装着

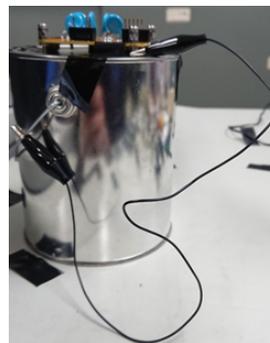


図 4.20 プリアンプのグラウンド接続

また、密封度を高めるため、図 4.21 のような市販のゴムパッキンを使用したり、パッキンの代わりとしてアルコール形シリコーン充てん材であるセメダイン株式会社のバスコーク N[17] を缶の蓋に塗るなどの工夫も行った。塗った時の外観を図 4.22 に示す。



図 4.21 市販のゴムパッキン



図 4.22 バスコーク N を塗った小缶の蓋

4.4.2 改良した検出器での半減期の導出

以下に先行実験での缶とほぼ同じ体積の小缶で前節の二つ目の工夫をし、パッキン代わりにバスコーク N を塗った図 4.22 の蓋を用いてラドンガスを測定したときの波高分布を図 4.23、その時の測定データを指数関数で fit した結果を図 4.24 に示す。ここで、図 4.24 の横軸は測定を開始してからの経過時間、縦軸はそれぞれのピークの Mean から 2σ 以内に入っているデータの 10 分ごとのカウント数である。また、ラドンガスは DOLLSTONE を缶の中に約 24 時間入れることで封入し、実験の測定は 240 時間行った。

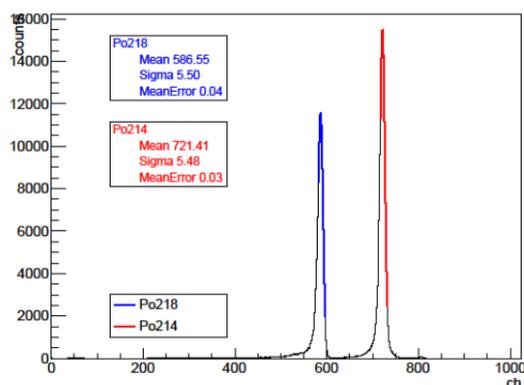


図 4.23 小缶を用いた時のラドンガスの波高分布

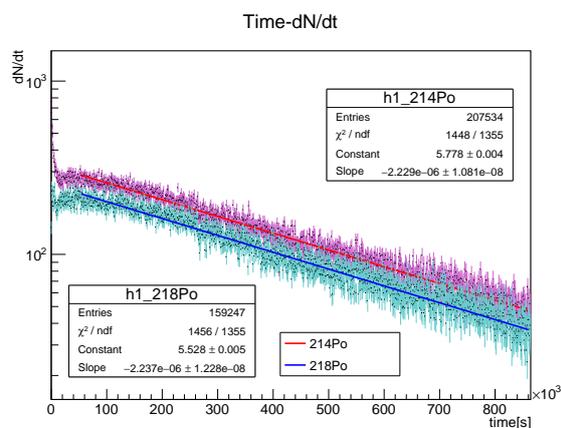


図 4.24 図 4.23 のデータを式指数関数で fit した結果

図 4.24 は先行実験での缶を用いた時よりも測定時間が長いカウント数をプロットする際の単位時間を同じにしたため、プロット数が非常に多くなりデータの大まかな情報が分かりにくいので、単位時間を 100 分にして図 4.23 のデータを指数関数で fit した図 4.25 を以下に示す。

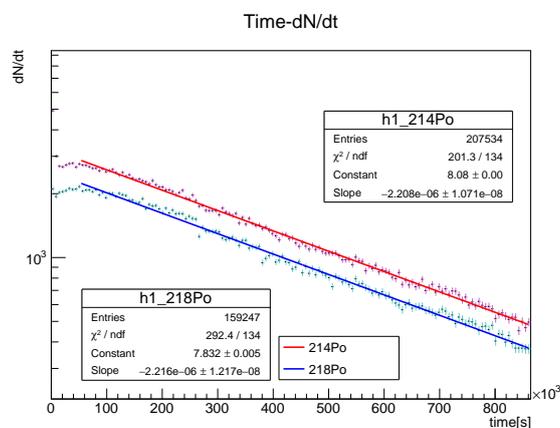


図 4.25 図 4.23 のデータを指数関数で fit した結果 (単位時間は 100 分)

図 4.25 より、プロットされたデータ点は多少のばらつきがあるものの、全体的に見ると fit はうまく行われていると考えられる。

fit により得られた崩壊定数 λ と、それを用いて求めた半減期、 ^{222}Rn の半減期の文献値をまとめた表を以下の表 4.3、表 4.4 に示す。

表 4.3 図 4.24 の fit 結果

	^{218}Po の fit 結果	^{214}Po の fit 結果	文献値
崩壊定数 λ	$2.237 \times 10^{-6} \pm 1.228 \times 10^{-8}$	$2.229 \times 10^{-6} \pm 1.081 \times 10^{-8}$	2.09×10^{-6}
半減期 [day]	3.587 ± 0.020	3.600 ± 0.017	3.82

表 4.4 図 4.25 の fit 結果

	^{218}Po の fit 結果	^{214}Po の fit 結果	文献値
崩壊定数 λ	$2.216 \times 10^{-6} \pm 1.217 \times 10^{-8}$	$2.208 \times 10^{-6} \pm 1.071 \times 10^{-8}$	2.09×10^{-6}
半減期 [day]	3.620 ± 0.020	3.633 ± 0.018	3.82

表 4.3 と表 4.4 を見比べると、求められた半減期に多少の違いがあるが、それぞれの誤差の範囲で重なる部分があるので、単位時間を変更しても求められる半減期に大きな違いは現れないことが分かった。また、表 4.3、表 4.4 より、 ^{222}Rn の半減期と誤差の範囲で一致しなかったものの、先行実験での缶を用いた時よりも求められた半減期が大幅に近づいたことが分かる。よって密封度を高めることで小型ラドン検出器の性能がよくなると考えられるが、先行実験での検出器を用いた実験と小缶を用いた実験ではラドンガスの封入時間、実験の測定時間が異なるため、これらの条件を同じにして密封度を変える実験を行い、密封度の小型ラドン検出器の性能にもたらす効果を調べることにした。

4.5 密封度の効果

ラドン検出部としては大缶を使用し、一つ目の工夫をした図 4.13 の蓋を用いた。プリアンプを取り付けるための 4 つのねじ穴のうち、プリアンプのグラウンドを缶と接続している部分のねじ穴以外のねじ穴をいくつか開けたままにすることで密封度を変更した。以下にラドンガスの封入時間を 24 時間、実験の測定時間を 24 時間として、密封度を変更したときのそれぞれの測定結果を示す。ここで、それぞれの時のラドンガスの波高分布のデータを fit した結果の横軸は測定を開始してからの経過時間、縦軸はそれぞれのピークの Mean から 2σ 以内に入っているデータの 10 分ごとのカウント数である。

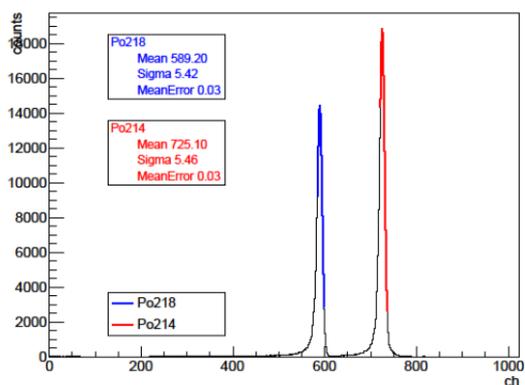


図 4.26 開けたねじ穴 0 個の時のラドンガスの波高分布

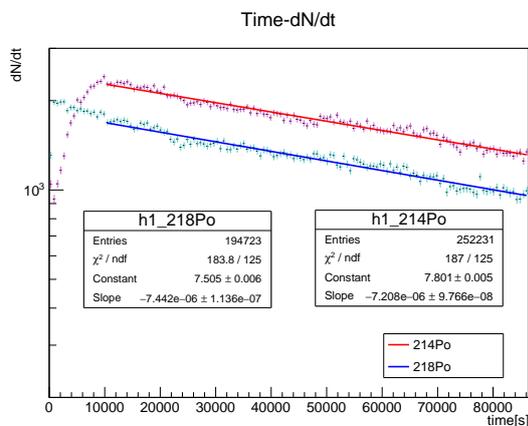


図 4.27 図 4.26 のデータを指数関数で fit した結果

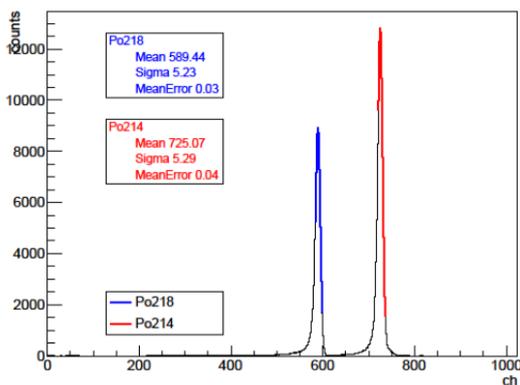


図 4.28 開けたねじ穴 1 個の時のラドンガスの波高分布

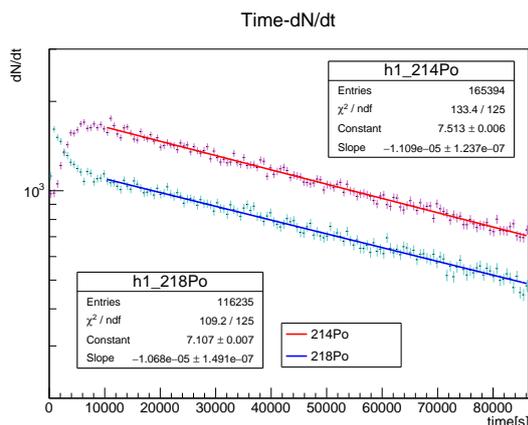


図 4.29 図 4.28 のデータを指数関数で fit した結果

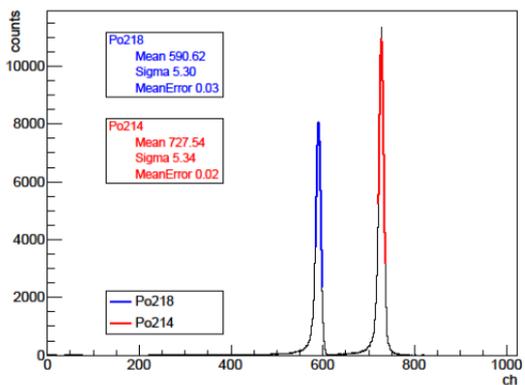


図 4.30 開けたねじ穴 2 個の時のラドンガスの波高分布

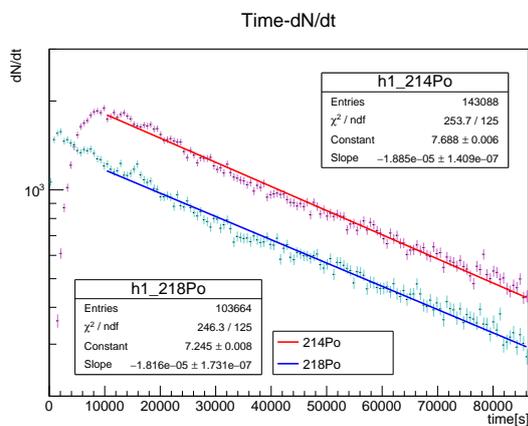


図 4.31 図 4.30 のデータを指数関数で fit した結果

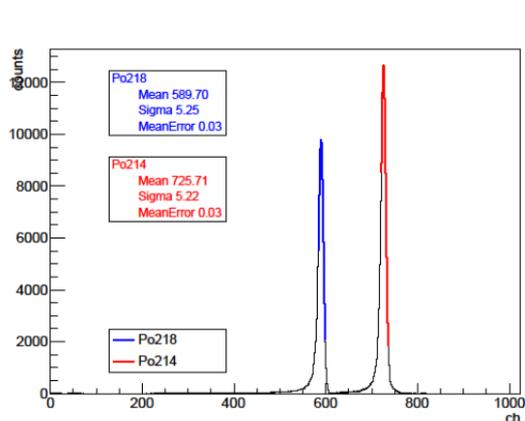


図 4.32 開けたねじ穴 3 個の時のラドンガスの波高分布

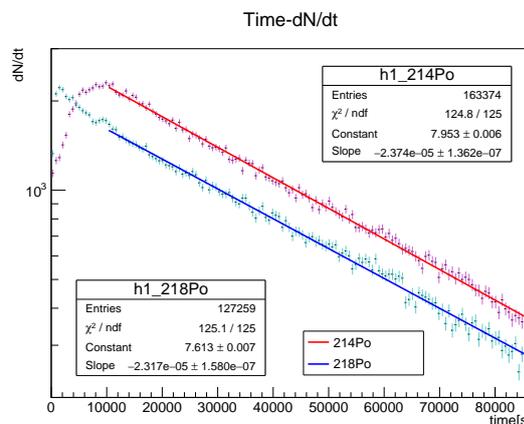


図 4.33 図 4.32 のデータを指数関数で fit した結果

fit により得られた崩壊定数 λ と、それを用いて求めた半減期、 ^{222}Rn の半減期の文献値をまとめた表を以下の表 4.5、表 4.6、表 4.7、表 4.8 に示す。

表 4.5 図 4.27 の fit 結果

	^{218}Po の fit 結果	^{214}Po の fit 結果	文献値
崩壊定数 λ	$7.442 \times 10^{-6} \pm 1.136 \times 10^{-7}$	$7.208 \times 10^{-6} \pm 9.766 \times 10^{-8}$	2.09×10^{-6}
半減期 [day]	1.078 ± 0.016	1.113 ± 0.015	3.82

表 4.6 図 4.29 の fit 結果

	^{218}Po の fit 結果	^{214}Po の fit 結果	文献値
崩壊定数 λ	$1.068 \times 10^{-5} \pm 1.491 \times 10^{-7}$	$1.109 \times 10^{-5} \pm 1.237 \times 10^{-7}$	2.09×10^{-6}
半減期 [day]	0.751 ± 0.010	0.723 ± 0.008	3.82

表 4.7 図 4.31 の fit 結果

	^{218}Po の fit 結果	^{214}Po の fit 結果	文献値
崩壊定数 λ	$1.816 \times 10^{-5} \pm 1.731 \times 10^{-7}$	$1.885 \times 10^{-5} \pm 1.409 \times 10^{-7}$	2.09×10^{-6}
半減期 [day]	0.442 ± 0.004	0.426 ± 0.003	3.82

表 4.8 図 4.33 の fit 結果

	^{218}Po の fit 結果	^{214}Po の fit 結果	文献値
崩壊定数 λ	$2.317 \times 10^{-5} \pm 1.580 \times 10^{-7}$	$2.374 \times 10^{-5} \pm 1.362 \times 10^{-7}$	2.09×10^{-6}
半減期 [day]	0.346 ± 0.002	0.338 ± 0.002	3.82

表 4.5、表 4.6、表 4.7、表 4.8 を見比べると、開けたねじ穴の数が多くなるにつれて測定結果から求められた半減期が、 ^{222}Rn の半減期の文献値から遠ざかっていることが分かる。よって、小型ラドン検出器の性能には検出器の密封度が重要であり、密封度を高めることで小型ラドン検出器の性能がよくなることが分かった。

第5章

まとめ

5.1 まとめ

ラドンとは身近な放射性物質である。静電捕集法で Po が放出する α 線を検出した。検出器を用いてラドンガスの測定、二つのピークを検出した。 ^{241}Am 線源を用いて、エネルギーと波高の関係を表す較正直線を引いた。較正直線から二つのピークが ^{218}Po と ^{214}Po ということがわかった。 ^{218}Po と ^{214}Po の単位時間当たりの崩壊数を調べて半減期を導出した。密封度を高める工夫し、 ^{222}Rn の半減期の文献値が 3.82 日のところ、 ^{218}Po の半減期は、 3.620 ± 0.020 日、 ^{214}Po の半減期は、 3.633 ± 0.018 日という測定結果を得た。密封度のみを変える実験を行い密封度が半減期に影響を与えることが分かった。

5.2 今後の課題・展望

今後の課題としては、本研究の実験により求められた半減期は ^{222}Rn の半減期の文献値と誤差の範囲では一致していないので、より性能の良い検出器にするために、より密封度の高い構造の缶を使用する、缶の蓋を閉めて上からテープでふさぐなど、密封度を高める工夫をすることが挙げられる。

また、本研究では時間的に行うことができなかったが、実験により求めた半減期が ^{222}Rn の半減期と大きく離れる原因の考察として挙げていた缶の体積、実験時の温度や湿度がどれくらい半減期に影響を及ぼすか実験をすることが挙げられる。まず、缶の体積については小缶、中缶、大缶と体積の違う缶が 3 つあるため、缶の蓋の構造、ラドンガスの封入時間等の条件をそろえて実験を行う方法が挙げられる。次に実験時の温度・湿度については、缶に穴を開けて温湿度計の外部センサーを缶に取り付け、実験を行う方法が挙げられる。温度の変更については実験室のエアコンを使用することや、検出器の外部にカイロや氷を置くことが挙げられる。ただし、ある一定の温度に保つことを考えると、極度な温度でない限り実験室のエアコンを使用するのが一番いいと考えられる。湿度の変更については、缶の中に乾燥材を入れることや、水を入れたコップを入れることなどが挙げられる。この際、PD 表面に水滴がつかないように気を付けなければならない。

今後の展望としては、大気中のラドンガスの濃度変動が地震予測につながるかもしれないと

いう研究 [18] があり、その研究に役立てるため全国各地に設置できるように検出器の小型化を行うことが挙げられる。また、大気中のラドンガスの濃度変動を測定するためのシステムの構築が挙げられる。例えば、大気を封入するための缶を2台用意し、片方の缶のラドンガス測定を行っている間に、もう片方の缶は大気を封入するようにするなどである。

謝辞

本研究に取り組むにあたり、多くの方々にお世話になりましたことをこの場を借りて深くお礼申し上げます。

指導教官である下村真弥助教には、本研究について右も左も分かっていなかった私たちに本研究についての資料や研究でつまずいたときにご助言をいただき、お忙しい中産休育休に入られるまで卒業研究報告の時間を毎週設けてくださり、産休育休に入られてからも進捗を気にかけてくださり、多くの面でご指導いただきましたことを深く感謝するとともに御礼申し上げます。また、蜂谷崇助教には、下村真弥助教が産休育休に入られてからの進捗報告を聞いてくださり、またプログラムなど多くのことに関してご助言いただき、深く感謝しております。

林井久樹教授には、PDのアノードとカソードの見分け方を講義を交えて教えていただくなど様々なことにご助言いただき、大変お世話になりました。宮林謙吉教授には、実験をしている様子を気にかけては色々にご助言いただき、大変感謝しております。

また、筑波大学の方々、研究室の港陽子先輩や鈴木彩香先輩には、先行研究として大変多くのことを学ばせていただき、深く感謝するとともに御礼申し上げます。研究室のほかの方々にも、様々なことに関するご助言や励ましの言葉をいただき、多くの面で支えていただいたことを深く感謝しております。放射線物理学研究室には、本研究で使用する ^{241}Am 密封線源を何度もお貸しくださり、誠にありがとうございました。

最後に、本研究においてご支援賜りましたすべての方々に深謝いたしますとともに、改めて深く御礼申し上げます。

付録 A

プログラム

A.1 Arduino のプログラム

Listing A.1 Arduino

```

1 //S/の出力H(ADC)はの番ピン Analog0
2 //信号はの7番ピン GateDigital
3 //信号はの番ピン TrigDigital2
4 //グラウンドは共通にする
5
6 const byte ADC_PIN = 0;
7 const byte TRG = 0; // 0: pin-2 1: pin-3
8 const byte bitmask = B00011111;
9
10 volatile int num; //データを入れる箱をつくる
11 int nloop = 100;
12
13 void setup(){
14     pinMode(7, OUTPUT); //ピン GATE
15     digitalWrite(7, LOW); //最初ゼロの値を与える
16     attachInterrupt(TRG, GATE, RISING);
17     //がしたら関数を実行 TRGRisingGATE
18     Serial.begin(9600);
19
20     //speed test
21     int time = millis();
22     for( int i=0; i<nloop; i++){
23         GATE();
24     }
25     Serial.println("//Speed Test");
26     Serial.print("loop of");
27     Serial.println(nloop);
28     time = millis() - time;
29     Serial.print("Time:");
30     Serial.print(time);
31     Serial.println("ms");
32 }
33
34 void GATE(){
35     detachInterrupt(TRG);
36     //not to interrupt during this function
37
38     //信号 Gate
39     digitalWrite(7, HIGH);
40     delayMicroseconds(8); //単位はusec
41     digitalWrite(7, LOW);
42
43     num = analogRead(ADC_PIN);
44     //アナログピンの値を読み込む
45
46     delayMicroseconds(200); //時間をおく
47
48     //リセット用の信号 Gate
49     digitalWrite(7, HIGH);
50     delayMicroseconds(3);
51     digitalWrite(7, LOW);
52
53     senddata( num );
54     attachInterrupt(TRG, GATE, RISING);
55 }
56
57 byte getParity( byte val ){
58     byte x;
59     x = val >> 4;
60     val = val ^ x;
61     x = val >> 2;
62     val = val ^ x;
63     x = val >> 1;
64     val = val ^ x;
65
66     return ( val & 1 );
67 }
68
69 void senddata( int val ){
70     //convert data
71     byte lowVal = ( val & bitmask );
72     byte higVal = ( ( val >> 5 ) & bitmask );
73
74     //horizontal parity check
75     byte lowParity = getParity( lowVal );
76     byte higParity = getParity( higVal );
77
78     //vertical parity check
79     byte andParity = getParity( higVal &
80         lowVal );
81     byte orParity = getParity( higVal |
82         lowVal );
83
84     //make packet
85     byte higBit = B10000000;
86     byte lowBit = B00000000;
87     higBit += ( higVal << 2 );
88     lowBit += ( lowVal << 2 );
89     higBit += ( andParity << 1 );
90     lowBit += ( orParity << 1 );
91     higBit += higParity;
92     lowBit += lowParity;

```

```

96     Serial.write( higBit );
97     Serial.write( lowBit );
98
99
100    //byte tmp[2];
101    //tmp[0] = higBit;
102    //tmp[1] = lowBit;
103    //Serial.write( tmp, 2 );
104 }
105
106
107 void loop(){
108     //delay(3000);
109     //senddata( 750 );
110 }
111
112 \normalsize

```

A.2 processing のプログラム

Listing A.2 processing

```

1  \tiny
2  import processing.serial.*; //Serial を読め
3  るおまじない library
4
5  //int totaltime = 1000; //(s)
6  int totaltime = 1000000; //(s)
7  int startTime;
8  int currentTime;
9  int entry;
10 int missedEvent;
11 int raw_data;
12 int isHoldData;
13 int[] data = new int[1024];
14
15 String graph_t = "radongas0223"; //リアルタイ
16 ムで更新されるグラフの表示タイトル
17 String filename = "radongas0223.txt"; //保
18 存するファイルの名前
19
20 //GUI vairables for data monitoring
21 int lens = 5; //拡大縮小倍率
22 float zoom_max = 125.0;
23 float zoom_min = 0.04;
24 float fZoom = 1.0;
25 float plotX1, plotX2;
26 float plotY1, plotY2;
27 float labelX, labelY;
28 float tmp_y;
29 boolean press=false;
30 String words;
31
32 PrintWriter outFile; //保存用の変数を用意
33 Serial usbPort;
34 int port_id;
35 int nport;
36 boolean selectedPort;
37
38 String user_filename;
39 char letter;
40
41 //for serial communication
42 class packet {
43     int digit;
44     int data;
45     int check;
46     int parity;
47     packet(){}
48 };
49 packet pkt1 = new packet();
50 packet pkt2 = new packet();
51
52 int mask_digit = unbinary("1000000");
53
54 int mask_data = unbinary("01111100");
55 int mask_check = unbinary("00000010");
56 int mask_parity = unbinary("00000001");
57
58 //初期設定
59 void setup(){
60     size(1124,600);
61
62     //ファイル生成
63     outFile = createWriter( filename ); //
64     ファイルを生
65     成
66     outFile.print("TIME(s)"); //1行ごとの書き
67     込み
68     outFile.print(TAB);
69     outFile.println("PulseHeight(ch)");
70
71     //canvas area
72     plotX1 = 50;
73     plotX2 = width - plotX1;
74     plotY1 = 90;
75     plotY2 = height - plotY1;
76
77     //initialization
78     entry = 0;
79     isHoldData = 0;
80     missedEvent = 0;
81     selectedPort = false;
82 }
83
84 //ループ関数
85 void draw() {
86     currentTime = (millis() - startTime)
87     /1000;
88     if( currentTime >= totaltime ){
89         outFile.flush(); //データの保存
90         outFile.close(); //ファイルを閉じる
91         exit(); // Stops the program
92     }
93
94     //Online monitoring
95     makeCanvas();
96     makeButton();
97     drawTitle();
98     drawTime(currentTime);
99     drawAxis();
100    drawHistogram();
101    strokeWeight(1);
102
103    if( !selectedPort ) setSerialPort
104    ();
105 }

```

```

106 void analyzePacket( int val, packet p ){ 177
107     p.digit = ( val & mask_digit ) >> 7; 178
108     p.data = ( val & mask_data ) >> 2; 179
109     p.check = ( val & mask_check ) >> 1; 180
110     p.parity = ( val & mask_parity ); 181
111 } 182
112
113 int getParity( int val ){ 183
114     int x; 184
115     x = val >> 4; 185
116     val = val^x; 186
117     x = val >> 2; 187
118     val = val^x; 188
119     x = val >> 1; 189
120     val = val^x; 190
121     return ( val & 1 ); 191
122 } 192
123
124 int getData( packet p1, packet p2 ){ 193
125     int p_sum1 = 0; 194
126     int p_sum2 = 0; 195
127
128     //horizontal parity check 196
129     if( getParity(p1.data)!=p1.parity ) 197
130         return -1; 198
131     if( getParity(p2.data)!=p2.parity ) 199
132         return -1; 200
133
134     //vertical parity check 201
135     int andParity = getParity( p1.data & p2 206
136         .data ); 207
137     int orParity = getParity( p1.data | p2 208
138         .data ); 209
139     if( andParity!=p1.check ) return -1; 210
140     if( orParity!=p2.check ) return -1; 211
141
142     //convert raw data to ADC value 212
143     int fValue = ( p1.data << 5 ); 213
144     fValue += p2.data; 214
145     if( fValue<0 || fValue>1023 ) return 215
146         -1; 216
147     return fValue; 217
148 } 218
149
150 void serialEvent(Serial usbPort){ 219
151     if( usbPort.available() > 1 ){ 220
152         //から送られてきたデータが以上ならArduino1 221
153
154         if( isHoldData!=1 ){ 222
155             raw_data = usbPort.read(); 223
156             //からくるデータを読み込むusbPort 224
157             analyzePacket( raw_data, pkt1 ); 225
158             }else{ 226
159                 pkt1 = pkt2; 227
160             } 228
161
162             //check if pkt1 is higher bit 229
163             if( pkt1.digit!=1 ){ 230
164                 missedEvent++; 231
165                 return; 232
166             }else{ 233
167                 raw_data = usbPort.read(); 234
168                 analyzePacket( raw_data, pkt2 ); 235
169             } 236
170
171             //check if pkt2 is lower bit 237
172             if( pkt2.digit==0 ){ 238
173                 int value = getData( pkt1, pkt2 ) 239
174                 ; 240
175                 if( value== -1 ) missedEvent++; 241
176                 else saveData( value ) 242
177                 ; 243
178                 isHoldData = 0; 244
179             }else{ 245
180                 isHoldData = 1; 246
181             } 247
182             }//available 248
183     } 249
184
185     void saveData( int adc ){ 250
186         entry++; 251
187         data[adc]++; 252
188
189         int passedtime = millis() - startTime; 253
190         int new_passed_time = floor(passedtime 254
191             /1000); 255
192
193         outFile.print(new_passed_time); //1行ごとの 256
194         書き込み 257
195         outFile.print(TAB); 258
196         outFile.println(adc); 259
197         outFile.flush(); //データの保存 260
198     } 261
199
200     //Functions for Online monitoring 262
201     //----- 263
202     void makeCanvas(){ 264
203         background(#87CEFA); 265
204         fill(255); 266
205         rectMode(CORNERS); 267
206         noStroke(); 268
207         rect(plotX1, plotY1, plotX2, plotY2); 269
208         //plot area 270
209     } 271
272
273     void drawTitle() { 272
274         fill(0); 273
275         textSize(20); 274
276         textAlign(LEFT); 275
277         text(graph_t, plotX1, plotY1 - 10); 276
278
279         text("Entries", 500, 48); 277
280         text(entry, 580,48); 278
281         text("missed events", 430, 68); 279
282         text(missedEvent, 580,68); 280
283     } 281
284
285     void drawTime( int in_time ){ 282
286         fill(0); 283
287         textSize(20); 284
288         textAlign(LEFT); 285
289         if (in_time < 60) { 286
290             text(in_time, 50, 50); 287
291             text(" [s]", 80, 50); 288
292         } 289
293         else if (in_time >= 60 && in_time < 290
294             3600) { 291
295             text(in_time/60, 50, 50); 292
296             text(" [min]", 80, 50); 293
297             text(in_time%60, 200, 50); 294
298         } 295
299     } 296
300

```

```

247     text("_[s]", 230, 50);           313
248 }                                   314
249 else if (in_time >= 3600) {         315
250     text(in_time/3600, 50, 50);
251     text("_[h]", 80, 50);           316
252     text((in_time%3600)/60, 200, 50); 317
253     text("_[min]", 230, 50);
254     text((in_time%3600)%60, 350, 50); 318
255     text("_[s]", 380, 50);           319
256 }                                   320
257 }                                   321
258 }                                   322
259                                   323
260 void drawAxis(){                    324
261                                   325
262 //Set axis labels and grid lines
263 textAlign(CENTER);
264 for (int k = 1; k <= 10; k++) {     326
265     stroke(#D3D3D3);                327
266     line(50+k*100, 510, 50+k*100, 90); // 328
267     //vertical lines
268     if (k < 5) {                    329
269         line(50, 100*k+10, 1074, 100*k 330
270             +10); //horizontal lines
271         textSize(12);
272         if ( fZoom<zoom_max ) text((    331
273             int)((500-k*100)/fZoom),    332
274             30, 100*k+10);             333
275     } else text                       334
276     ((5-k), 30, 100*k+10);           335
277 // y-label
278     text(k*100, 50+k*100, 530); // x- 336
279     label                             337
280 }                                     338
281 // y-label
282     text(k*100, 50+k*100, 530); // x- 339
283     label                             340
284 }                                     341
285                                   342
286 text(0, 30, 510); // y=0
287 text("0", 50, 530); // x=0
288 text("[Ch]", 1060, 550);
289 text("[Count]", 20, 90);
290 text("1024", 1094, 530);
291 }
292
293 void drawHistogram(){
294 //Draw bin contents
295 for (int j = 0; j < 1024; j++) {
296     if (data[j]!=0){
297         if( fZoom<zoom_max ) tmp_y =
298             fZoom * data[j];
299         else tmp_y =
300             = 100 * data[j];
301         //draw line of content at j-
302         bin
303         if (tmp_y < 420) {
304             stroke(#0000FF);
305             line(j+plotX1, plotY2, j+plotX1,
306                 plotY2-tmp_y);
307         } else {
308             stroke(#0000FF);
309             line(j+50, 510, j+50, 90);
310         }
311     }
312 }
313 }
314
315 void makeButton(){
316     fill(0);
317
318     textSize(20);
319     textAlign(LEFT);
320     if (mouseX>650 && mouseX<750 &&
321         mouseY>50 && mouseY<70){
322         //ボタンの矩形範囲にマウスがある 時
323         fill(#000000); //ボタン塗り色を#にす
324             るF0F8FF
325         stroke(#696969);
326     }
327     else{
328         fill(#F0F8FF); //ボタン塗り色を#にす
329             るF0F8FF
330         stroke(#DCDCDC);
331     }
332     rect(750, 50, 650, 70);
333     if (mouseX>650 && mouseX<750 &&
334         mouseY>50 && mouseY<70){
335         //ボタンの矩形範囲にマウスがある 時
336         fill(#F8F8FF); //色をmoji#にす
337             るF0F8FF
338     }
339     else{
340         fill(#000000); //色をmoji#にす
341             るF0F8FF
342     }
343     text("ZOOM", 670, 68);
344     if (mouseX>800 && mouseX<900 &&
345         mouseY>50 && mouseY<70){
346         //ボタンの矩形範囲にマウスがある 時
347         fill(#000000); //ボタン塗り色を#にす
348             るF0F8FF
349         stroke(#696969);
350     }
351     else{
352         fill(#F0F8FF); //ボタン塗り色を#にす
353             るF0F8FF
354         stroke(#DCDCDC);
355     }
356     rect(800, 50, 900, 70);
357     if (mouseX>800 && mouseX<900 &&
358         mouseY>50 && mouseY<70){
359         //ボタンの矩形範囲にマウスがある 時
360         fill(#F8F8FF); //色をmoji#にす
361             るF0F8FF
362     }
363     else{
364         fill(#000000); //色をmoji#にす
365             るF0F8FF
366     }
367     text("OUT", 830, 68);
368 }
369
370 void mousePressed(){ //マウスが押されたら発動する
371     if(mouseX>650 && mouseX<750 && mouseY
372         >50 && mouseY<70){
373         //ボタンの矩形範囲にマウスがある 時
374         fZoom = (int)( fZoom * lens * 100
375             + 0.5 ) / 100.0 ;
376     }
377     else if(mouseX>800 && mouseX<900 &&
378         mouseY>50 && mouseY<70){
379         //ボタンの矩形範囲にマウスが ある時
380         fZoom = (int)( fZoom / lens * 100
381             + 0.5 ) / 100.0 ;
382     }
383     if( fZoom < zoom_min ) fZoom =
384         zoom_min;
385     if( fZoom > zoom_max ) fZoom =
386         zoom_max;
387 }
388
389 void setSerialPort(){
390     background(0);

```

```

373 fill(255);
374 textAlign(LEFT);
375 textSize(22);
376
377 //シリアル通信のための設定
378 String port[] = Serial.list();
379 println(port);
380 nport = port.length;
381 text("Select a port number in your
382 keyboard.", 20, 40);
383 for( int i=0; i<nport; i++){
384 text( i, 40, 30*i+80 );
385 text( port[i], 60, 30*i+80 );
386 }
387 selectedPort = false;
388 noLoop();
389 }
390
391 void keyPressed(){
392
393 if( selectedPort ) return;
394
395 int val = int( key );
396 if( val>47 && val<57 ){
397
398 val -= 48;
399 if( val<nport ){
400 port_id = val;
401 print(val);
402 println(" is pressed.");
403 selectedPort = true;
404 usbPort = new Serial( this, Serial.
405 list()[port_id], 9600 );
406 //usbPort = new Serial( this, "/dev
407 /cu.usbserial-A9016A67", 9600 );
408 startTime = millis();
409 loop();
410 }
411 }
412 void keyPressed_F() {
413 // The variable "key" always contains
414 the value
415 // of the most recent key pressed.
416 if ((key >= 'A' && key <= 'z') || key
417 == '.') {
418 letter = key;
419 words = words + key;
420 // Write the letter to the
421 console
422 println(key);
423 }
424 }
425 void setFileName(){
426
427 background(0);
428 fill(255);
429 textAlign(LEFT);
430 textSize(22);
431
432 //シリアル通信のための設定
433 String port[] = Serial.list();
434 println(port);
435 nport = port.length;
436 text(" Input the file name in your
437 keyboard.", 20, 40);
438 text(" Save as", 20, 40);
439 text("The String is" + words.length
440 () + " characters long", 50, 90);
441
442 selectedPort = false;
443 noLoop();
444 }

```

参考文献

- [1] 「環境省「放射線による健康影響等に関する統一的な基礎資料（平成26年度版）」
<https://www.env.go.jp/chemi/rhm/kisoshiryo/attach/201510mat1s-01-6.pdf>
- [2] 「放射線 α β γ 」
<http://contest.japias.jp/tqj14/140054/syurui.html>
- [3] 「放射線による健康影響等に関する統一的な基礎資料（平成29年度版）」
<https://www.env.go.jp/chemi/rhm/h29kisoshiryo/h29kiso-01-03-08.html>
- [4] 「ブリタニカ国際大百科事典 小項目辞典 より α 崩壊について」
<https://kotobank.jp/word/P%E5%9E%8B%E5%8D%8A%E5%B0%8E%E4%BD%93-2076517%E7.B2.BE.E9.81.B8.E7.89.88.20.E6.97.A5.E6.9C.AC.E5.9B.BD.E8.AA.9E.E5.A4.A7.E8.BE.9E.E5.85.B8>
- [5] 「物理メモ」
<https://butsurimemo.com/create-new-elements/>
- [6] 「素粒子・原子核物理入門 B. ポッフ/K. リーツ/C. ショルツ/F. サッチャ柴田利明 訳」
- [7] 「浜松ホトニクス公式ページより、フォトダイオード」
https://www.hamamatsu.com/resources/pdf/ssd/s3590-08_etc_kpin1052j.pdf \\ https://www.hamamatsu.com/resources/pdf/ssd/02_handbook.pdf
- [8] 「第1種放射線取扱主任者試験 マスター・ノート 3rd edition 2015年12月25日 第3版第1刷発行 福士政広編集」
- [9] 「放射性崩壊系列の数学 - FN の高校物理（分野別目次）」2019/1/23
http://fnorio.com/0079radioactive_decay_series1/radioactive_decay_series1.htm
- [10] 第9回 大学生のための素粒子・原子核、物質・生命スクール サマーチャレンジ
 演習課題 P03 ミニ素粒子・原子核実験 ～ラドン検出器～
 「演習課題のご紹介」2019/1/28
https://www2.kek.jp/ksc/9th_2015/enshu.html 講義資料 sc2015_eleki.pdf
- [11] 川又 晃, ”エレクトロニクス基礎回路講座（2）パルス基礎回路”, 日刊工業新聞社, pp.38-43. 昭和40年11月10日 5版発行
- [12] 「よく分かる！ コンデンサの仕組みと働き（2）：電子回路で多様な活躍、交流を通すコ

- ンデンサの機能 (2/3) - EDN Japan」 2019/1/31
http://ednjapan.com/edn/articles/1205/23/news067_2.html
- [13] 片野義雄、磯部直吉、富田英雄、前田明志、本間和明、大庭勝實、宮下収、中村尚五、飯田祥二、西方正司、羽根吉寿正、川島忠雄”基礎 電気・電子工学”、東京電機大学出版局、pp.145-151. pp.187-193. 昭和63年3月20日 第1版2刷発行
- [14] 「コンパレータ（比較器）-電子工作で覚える！電子回路」 2019/2/1
http://www.kairo-nyumon.com/practice_comparator.html
- [15] 「コンパレータ回路の基本を勉強しよう、ポイントはこれだ!!、比較器の基本原理、切り替わり動作、コンパレータにおける注意点、内部回路の簡単な選定方法、とは、？、など」 2019/2/1
http://kaironohanashi.main.jp/html/advanced_exp/1_11_comp_basic.html
- [16] 「doll stone について - ホルミシスルームあたごさん」 2019/2/13 <http://atagosan.com/hr/dollstone.html>
- [17] 「浴室目地補修材バスコークシリーズ | 補修 | セメダイン株式会社」 2019/3/14
<https://www.cemedine.co.jp/home/repair/bathcalk/index.html>
- [18] 「地震に先行する大気中ラドン濃度変動に関する観測：文部科学省」 2019/3/18
http://www.mext.go.jp/b_menu/shingi/gijyutu/gijyutu6/sonota/attach/1348276.htm
- [19] 中島朋（2012）修士論文「ラドン検出器を用いた放射線教育—教材開発と実践—」
- [20] 港陽子、鈴木彩香（2017）卒業論文「小型ラドン検出器の開発」