

μ 粒子の寿命と電子エネルギースペクトラムの測定

北内 久美 太地 彩 武田 明莉

2015 年 3 月

目次

1	目的	4
2	宇宙線・ μ 粒子	4
2.1	宇宙線とは	4
2.2	μ 粒子とは	4
3	放射線測定の実理	6
3.1	μ 粒子の寿命測定	6
3.2	電子エネルギースペクトラム	9
3.3	制動放射	12
3.4	電離損失	13
3.5	シンチレーションカウンター	15
3.6	プラスチックシンチレーターのエネルギー損失	16
4	測定方法	17
4.1	計測全体の流れ	17
4.2	μ 粒子と電子の波形の見分け方	18
5	セットアップ	19
5.1	シンチレーションカウンターの配置	19
5.2	エレクトロニクスのセットアップ	22
5.3	CAMAC(Computer Aided Measurements And Control)	24
5.4	各 CAMAC のモジュールについてのセットアップ	26
5.5	エレクトロニクスの設定	29
5.6	データ収集プログラム	30
6	データ解析	34
6.1	μ 粒子の寿命測定	34
6.2	電子のエネルギースペクトラムの計測	41
7	考察	48
7.1	寿命測定 of 考察	48
7.2	電子エネルギースペクトラム of 測定 of 考察	50
8	まとめ	55
8.1	実験のまとめ	55
8.2	謝辞	55
8.3	参考文献	55
付録 A	ele.c (CAMAC からのデータ収集)	56
付録 B	mu-e.kumac (paw によるヒストグラム作成)	61
付録 C	readmue.f (ヒストグラムの選別条件)	62
付録 D	Geant4 の構造の概要	65

付録 E	exampleN03.cc	67
付録 F	EventAction.hh	70
付録 G	EvevtAction.cc	72
付録 H	SteppingAction.cc	75

μ^- 崩壊のファインマン図を図2に示す。ここで W^- は W ボソンである。

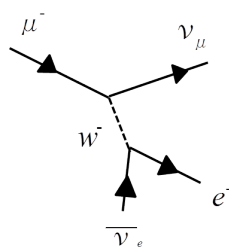


図2 ファインマン図

3 放射線測定の実験原理

放射線を私たちは普段直接感じることはできない。放射線検出には様々な手段が一般に用いられているが、本実験では、シンチレーター内の荷電粒子の電離作用と発光現象を用いて放射線を検出する。微弱なシンチレーション光を検出するため、シンチレーション光を光電子増倍管で増幅し、ADC でアナログ信号をデジタル信号に変換した。

今回の実験では、 μ 粒子の寿命測定と、 μ 粒子崩壊後放出された電子のエネルギー分布を測定した。以下でそれぞれの測定の原理について説明する。

3.1 μ 粒子の寿命測定

μ 粒子の寿命の逆数 τ_μ^{-1} は

$$\tau_\mu^{-1} = \frac{G_F^2 m_\mu^5}{192\pi^3} F\left(\frac{m_e^2}{m_\mu^2}\right) \left(1 + \frac{3}{5} \frac{m_\mu^2}{M_W^2}\right) \left[1 + \left(\frac{25}{8} - \frac{\pi^2}{2}\right) \frac{\alpha(m_\mu)}{\pi} + C_2 \frac{\alpha^2(m_\mu)}{\pi^2}\right] \quad (1)$$

と、表わせる。また

$$F(x) = 1 - 8x + 8x^3 - x^4 - 12x^2 \ln x \quad (2)$$

$$C_2 = \frac{156815}{5184} - \frac{518}{81}\pi^2 - \frac{895}{36}\zeta(3) + \frac{67}{720}\pi^4 + \frac{53}{6}\pi^2 \ln(2) \quad (3)$$

$$\alpha(m_\mu)^{-1} = \alpha^{-1} - \frac{2}{3\pi} \ln\left(\frac{m_\mu}{m_e}\right) + \frac{1}{6\pi} \approx 136 \quad (4)$$

$$G_F = 1.16637 \times 10^{-5} \text{GeV}^{-2} \quad (5)$$

である。ここで、 m_e は電子の質量、 m_μ は μ 粒子の質量、 M_W は W ボソンの質量である。

今回は実験により μ 粒子の寿命を測定する。 μ 粒子の寿命測定を行うにあたって、放射性崩壊の時に成り立つ指数関数法則を用いると、以下のように導かれる。

今、それぞれの粒子が単位時間に崩壊する確率を λ とする。独立な粒子の集合を考えると、短い時間 dt の間に崩壊する数 dN は次式で与えられる。

$$dN = -\lambda N(t) dt \quad (6)$$

ここで、 $N(t)$ はある時間 t に存在する粒子の数である。式 (6) を積分する。

$$\int_{N_0}^N \frac{dN}{N} = -\lambda \int_{t_0}^t dt \quad (7)$$

$$[\ln N]_{N_0}^N = -\lambda [t]_{t_0}^t \quad (8)$$

$$\ln \frac{N}{N_0} = -\lambda(t - t_0) \quad (9)$$

$t=0$ の時、式 (9) は次式に書き直せる。

$$N = N_0 e^{-\lambda t} \quad (10)$$

これを放射性崩壊の指数関数法則と呼ぶ。ここで $N(t)$ は時刻 t に崩壊せずに残っている粒子の数である。時間 t と $t + dt$ の間の無限小に小さい dt に崩壊する粒子の数は式 (6) で表わされる。 N_0 個すべての粒子の生存時間の積分和 L は

$$L = \int_0^\infty t N(t) \lambda dt = \int_0^\infty t N_0 e^{-\lambda t} \lambda dt = \left[-\frac{1}{\lambda} N_0 e^{-\lambda t}\right]_0^\infty = \frac{N_0}{\lambda} \quad (11)$$

以上より平均生存時間 L/N_0 (平均寿命 τ) は (12) 式で与えられる。

$$\frac{L}{N_0} = \tau = \frac{1}{\lambda} \Rightarrow \lambda = \frac{1}{\tau} \quad (12)$$

これを式 (10) に代入すると

$$N = N_0 e^{-\frac{t}{\tau}} \quad (13)$$

が得られる。

式 (13) を図 3 に示す。

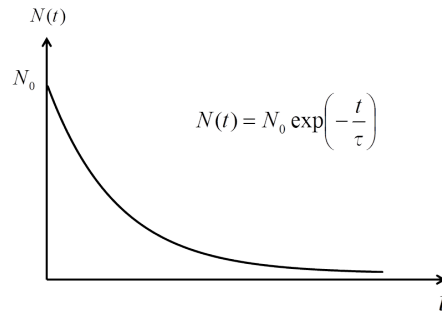


図 3 放射性崩壊の図

また、式 (13) を t で微分すると、以下で表わされる単位時間当たりの崩壊数の式が得られる。

$$\frac{dN}{dt} = -\frac{N_0}{\tau} \exp\left(-\frac{t}{\tau}\right) \quad (14)$$

ここで dN/dt は $t \rightarrow t+dt$ の間に N 個の粒子が $N+dN$ 個に減少したという意味である。すなわち dN は負である。 $t+dt$ の間に崩壊した μ 粒子の個数 $dN_{\text{崩壊}}$ は $dN_{\text{崩壊}} = -dN$ なので

$$\frac{dN_{\text{崩壊}}}{dt} = \frac{N_0}{\tau} \exp\left(-\frac{t}{\tau}\right) \quad (15)$$

となる。 μ 粒子がシンチレーター内に入射してから、電子が放出されるまでの時間を TDC で測定し、得られる TDC 分布は式 (15) の関数となる。これより平均寿命 τ を算出する。式 (15) を図 4 に、つまり得られる TDC 分布の概形を示す。

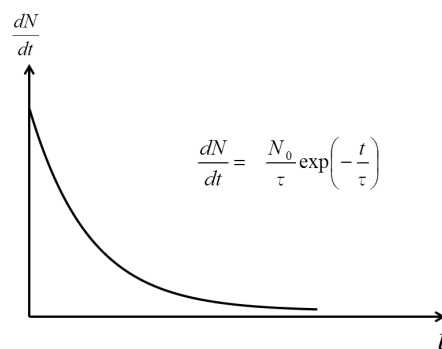


図 4 TDC 分布

3.2 電子エネルギースペクトラム

μ 粒子の崩壊によって生成される、電子のエネルギー分布について調べる。 μ 粒子の崩壊のファインマン図と、崩壊式の対応は以下ようになる。

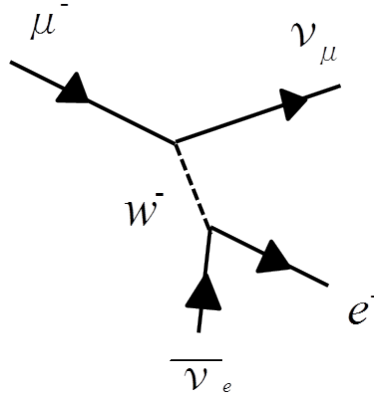


図5 ファインマン図

$$\mu^-(P_1) \rightarrow e^-(P_4) + \bar{\nu}_e(P_2) + \nu_\mu(P_3)$$

今回の実験で用いた電子のエネルギー分布は以下の式で導かれる。振幅 M は

$$M = \frac{g_w^2}{8(M_w c)^2} [u(3)\gamma^\mu(1 - \gamma^5)u(1)][u(4)\gamma_\mu(1 - \gamma^5)u(2)]$$

このように表わせる。

ここで、 M は崩壊振幅、 g_w は結合定数、 $\gamma^\mu \equiv (\gamma^0, \vec{\gamma})$ 、 $\gamma \equiv i\gamma^0\gamma^1\gamma^2\gamma^3$ 、 $u(i)$ ($i=1,2,3,4$) はスピノール (ディラック方程式の解) である。

よって

$$\langle [M^2] \rangle = 2\left(\frac{g_w}{M_w c}\right)^4 (P_1 \cdot P_2)(P_3 \cdot P_4) \quad (16)$$

P_1, P_2, P_3, P_4 はそれぞれ、 μ 粒子、電子、 μ ニュートリノ、電子ニュートリノの4元運動量である。 μ 粒子の静止系では、 $P_1 = (m_\mu c, \vec{0})$ となり、 P_1 と P_2 の内積 $P_1 \cdot P_2$ は

$$P_1 \cdot P_2 = m_\mu E_2 \quad (17)$$

となる。

また、エネルギー運動量保存則より

$$(P_3 + P_4)^2 = M_\mu^2 c^2 - 2P_1 \cdot P_2$$

よって

$$P_3 \cdot P_4 = \frac{(m_\mu^2 - m_e^2)c^2}{2} - m_\mu E_2 \quad (18)$$

となる。

式 (16) に式 (17) と式 (18) を代入すると、振幅 M は

$$M = \left(\frac{g_w}{M_w c}\right)^4 m_\mu^2 E_2 (m_\mu c^2 - 2E_2) \quad (19)$$

この振幅 M より、崩壊幅は次の式で与えられる。

$$d\Gamma = \frac{\langle [M^2] \rangle c}{(4\pi)^4 h m_\mu} dE_2 \frac{d^3 \vec{p}_4}{E_4^2} \quad (20)$$

式 (20) に式 (19) を代入し、積分する。

$$d\Gamma = \left(\frac{g_w}{4\pi M_w c}\right)^4 \frac{m_\mu c}{h} \left(\frac{m_\mu c^2}{2} - \frac{2}{3} E_4\right) d^3 \vec{p}_4 \quad (21)$$

また、

$$d^3 \vec{p}_4 = 4\pi \left(\frac{E_4}{c}\right)^2 \frac{dE_4}{c} \quad (22)$$

よって、式 (21) に式 (22) を代入すると

$$\frac{d\Gamma}{dE} = \left(\frac{g_w}{M_w c}\right)^4 \frac{m_\mu^2 E^2}{2h(4\pi)^3} \left(1 - \frac{4E}{3m_\mu c^2}\right) \quad (23)$$

が得られる。式 (23) は崩壊時に生成される電子の、エネルギー分布の式を与える。電子のエネルギーが最小になるのは、 μ 粒子の静止系で、電子が静止して作られる場合である。また、最大になるのは、2 個のニュートリノの運動量がともに電子の運動量の 180° 方向を向き、電子に最大の反跳を与える時である。(図 6) μ 粒子の静止系における電子のエネルギー分布は図 7 のようになり、電子が大きなエネルギーを持って崩壊する割合が高い。ここでは電子の質量を無視した。

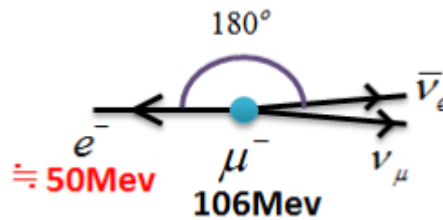


図 6 電子のエネルギーが最大になるときの μ 粒子の静止系における電子と 2 個のニュートリノの運動量ベクトル

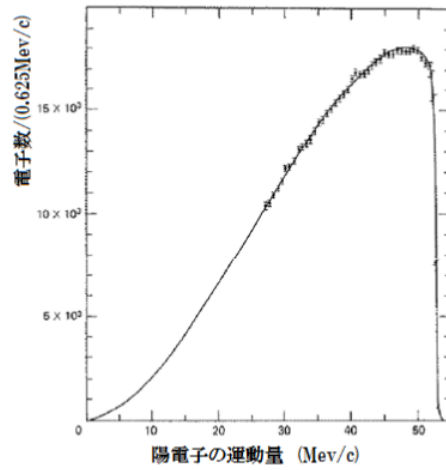


図7 μ 粒子崩壊時の電子のエネルギー分布

実際の実験ではシンチレーター中で静止した μ 粒子は元素核に束縛されてミュオニック原子を形成する。実験結果の考察ではこの効果を含める必要がある。詳しい議論は第7章で行った。また、プラスチックシンチレーター中に生成された電子はエネルギー損失とともに、制動放射で光子を放出してエネルギーを失う。観測する電子のエネルギー分布の考察にはこの効果の検討も重要である。次で制動放射について説明する。

3.3 制動放射

電子が物質中で他の荷電粒子と衝突した際に原子の励起やイオン化を行わない弾性衝突であっても、電子は質量が小さいので、原子核の作る電場によって速度の時間変化、方向の変化をかなり受け、加速度運動する。このとき、加速度の二乗に比例するエネルギーを電磁波として放出する。加速度の大きさは物質内の荷電粒子の電荷 Z に比例し、入射電子の質量に反比例するので、放出するエネルギーは $(Z/me)^2$ に比例する。したがって、物質内で Z の大きい原子核の存在が最も制動放射に効く。また、制動放射によるエネルギー損失は入射電子のエネルギーに比例して増していく。そのため、この損失は電子が高速になってから有効となる。詳しい計算によると、単位長さあたりに制動放射で失われる電子のエネルギーは次式で与えられる。

$$\frac{dE}{dx} = \frac{NEZ^2r_e^2}{137} \left(4 \ln \frac{183}{z^{\frac{1}{3}}} + \frac{2}{9} \right) \quad (24)$$

ここで、 N は物質の単位体積中の原子数 ($N = \frac{N_0\rho}{A}$)、 E は入射電子のエネルギー、 r_e は電子の古典半径 ($= \frac{e_0^2}{m_e c^2} = 2.81 \times 10^{-13}(cm)$)、 A は物質の原子量、 N_0 はアボガドロ数 ($= 6.0210223 \times 10^{23}$)、 ρ は物質の密度である。制動放射でエネルギーを失って、電子の始めのエネルギーの $\frac{1}{e}$ になるまでに走る物質層の長さを放射長という。式 (24) より

$$-\frac{dE}{E} = \frac{dx}{x_0} \quad (25)$$

となるので、放射長 x_0 を求めることが出来る。

また、放射長は電子が一回制動放射で γ を放出する際に進む距離であり、プラスチックシンチレーターの放射長は $43.79(g/cm^2)$ である。これは、電子エネルギー分布を測定する際、電子が測定用のシンチレーター内を走る際にどれだけのエネルギーが制動放射で失われるか評価するのに重要である。

3.4 電離損失

荷電粒子が物質中を通過すると、入射荷電粒子と物質を構成する原子との相互作用によって、原子が電子と陽イオンに分離される。これを電離という。また、電離作用を起こさずに、原子や分子がエネルギーの高い状態になることもある。これを励起、その状態を励起状態と呼ぶ。入射荷電粒子が物質を通過するとき、物質中の電子と衝突し、電離や励起を繰り返しながらエネルギーの一部を失う。これを電離損失という。電離損失によって荷電粒子が失うエネルギーは Bethe-Boch の式で表わされる。

$$\frac{dE}{dx} = 4\pi N_a r_e^2 m_e c^2 \rho \frac{Z z^2}{A \beta^2} \left[\ln\left(\frac{2m_e \gamma^2 v^2 W_{max}}{I^2}\right) - 2\beta^2 - \delta - 2\frac{C}{Z} \right] \quad (\text{Mev/g/cm}^2) \quad (26)$$

ここで、 r_e は電子古典半径、 N_a はアボガド数、 I は電離ポテンシャル、 Z は物質の原子番号、 A は物質の原子量、 $\beta = \frac{v}{c}$ 、 $\gamma = \frac{1}{\sqrt{1-\beta^2}}$ 、 v は入射粒子の速度、 z は入射粒子の電荷、 m_e は電子の質量、 W_{max} は single collision で最大エネルギー移行、 ρ は吸収物質の密度、 δ は密度補正、 C は殻補正である。

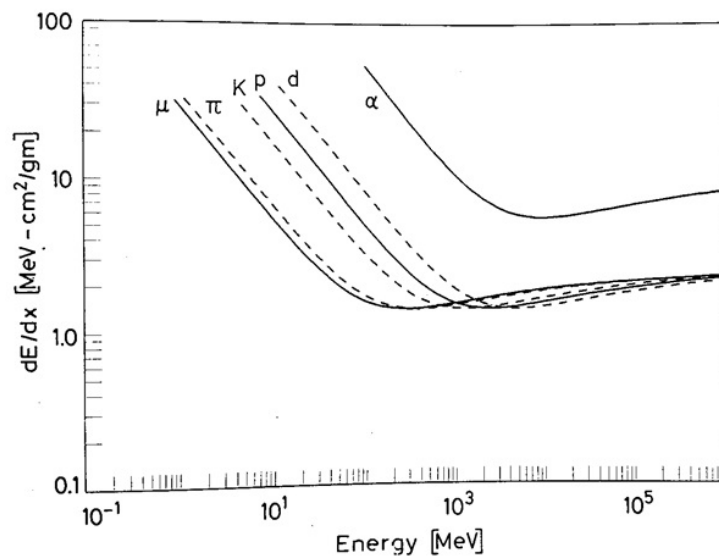


図8 電離損失

電離損失 $\beta = \frac{v}{c}$ は入射荷電粒子の質量には依存しないが、速度 $v = \beta c$ には依存する。入射荷電粒子のエネルギーが低く、 β が小さいとき、Bethe-Bloch の式は

$$\frac{dE}{dx} \propto \frac{1}{\beta^2} \quad (27)$$

となる。入射粒子の運動量が大きくなると、電離損失は $\frac{1}{\beta^2}$ に従って急激に減少し、最小値に達する。この領域での電離を Minimum ionization という。電荷が同じ粒子ならば、粒子の質量が 3 倍になるところで、最小値はほぼ同じ値をとる。

入射荷電粒子のエネルギーが高くなると、 $\beta^2 \simeq 1$ となり、式 (26) の \log の中の項が効くのでエネルギー損失は $\log \gamma$ でエネルギーが増加するにつれて上昇する。

$$\frac{dE}{dx} \propto \log\left[\frac{1}{1-\beta^2}\right] \quad (28)$$

$$\gamma = \frac{1}{1-\beta^2} = \frac{E}{m} \quad (29)$$

シンチレータ内の電離・励起状態にある原子・分子が基底状態に遷移するとき、二つの状態のエネルギーの差を光として放出する。この発光をシンチレーションという。今回の実験では、シンチレーション光を用いて、入射荷電粒子の電離損失により失うエネルギーを測定する。

3.5 シンチレーションカウンター

シンチレーションカウンターは放射線検出器の一種である。シンチレーターと光電子増倍管を組み合わせ、シンチレーター内で発生した光を光電子増倍管で電気信号に変換することによって放射線検出器として働く。以下でシンチレーターと光電子増倍管について詳しく述べる。

3.5.1 プラスチックシンチレーター

シンチレーターとは荷電粒子が通過する時にその粒子が電離損失によって失うエネルギーを光エネルギーに変換する機器であり、シンチレーターに荷電粒子が入射するとシンチレーション光が放出される。シンチレーターには大きく分けて有機シンチレーターと無機シンチレーターがあり応答速度に大きな違いがある。有機シンチレーターは無機シンチレーターより光の減衰時間がとても短い(約 nsec) ので、時間分解能が高い。しかしシンチレーションの光が弱く、高いエネルギー分解能が得られない。有機シンチレーターは、アントラセン (anthracene) やスチルベン (stilbene) などの環式化合物からできている有機結晶シンチレーター、ターフェニル (terphenyl) とポリスチレン (poly-styrene) との固溶体でできているプラスチックシンチレーター、p-ターフェニル (p-terphenyl) ・ジフェニルオキサゾール (di-phenyloxazol) などをキシレン (xylene) ・トルエン (toluene) などの溶媒に適量溶かした有機液体シンチレーターに大きく分けられる。今回は有機シンチレーターのプラスチックシンチレーターを用いる。

3.5.2 光電子増倍管

光電子増倍管とは光電効果を利用して光エネルギーを電気エネルギーに変換し電子を増倍させるものである。その模式図を図9に示す。光電面に光が入射すると光電効果により電子が飛び出す。光電面と各陰極の間には電圧がかかっているため、その電子は真空管中で加速される。加速された電子はダイノードに衝突し、一次電子より多くの二次電子を放出する。これを繰り返すことで電子を増幅する。その増幅量は入射光量に比例する。増幅された電子は陽極に集まり電気信号として出力される。

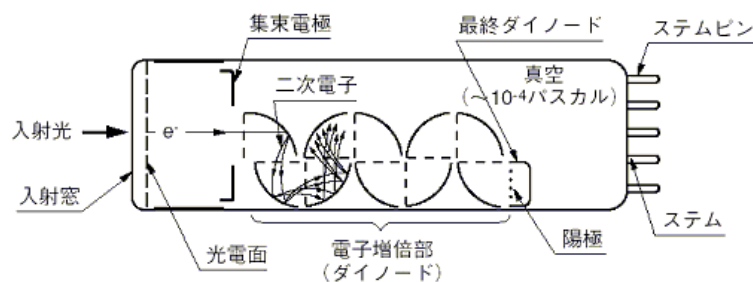


図9 光電子増倍管

3.6 プラスチックシンチレーターのエネルギー損失

プラスチックシンチレーターのエネルギー損失は以下の式で求められる。

$$\Delta E_{(MeV)} = \frac{dE}{dx \left(\frac{MeV}{g/cm^2} \right)} \times \rho_{(g/cm^2)} \times t_{(cm)} \quad (30)$$

ここで、 $\frac{dE}{dx}|_{min}$ は minimum ionization の粒子に対する厚さ $1g/cm^2$ あたりのエネルギー損失、 ρ は物質の密度、 t は物質の厚さである。それぞれの値は $\frac{dE}{dx} = 1.936 \left(\frac{MeV}{g/cm^2} \right)$ 、 $\rho = 1.06(g/cm^2)$ とする。

これより、プラスチックシンチレーターの $1cm$ あたりのエネルギー損失は以下のように求められる。

$$\Delta E = 1.936 \times 1.06 \times 1 = 2.05 \quad (MeV/cm) \quad (31)$$

今回の実験で用いる測定用のメインシンチレーターの厚みは $26cm$ なのでメインシンチレーターを通過した時のエネルギー損失は

$$2.05 \times 26 = 53.3 \quad (MeV) \quad (32)$$

となる。この値は、ADCのエネルギー較正を行う際に使用する。

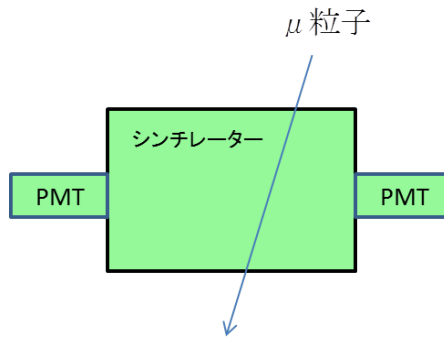


図 10 μ 粒子がシンチレーターを通過する様子

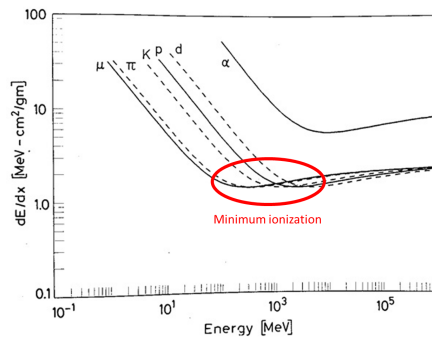


図 11 電離損失の minimum ionization 部

4 測定方法

4.1 計測全体の流れ

放射線計測に用いる装置の概要を図 12 に示す。

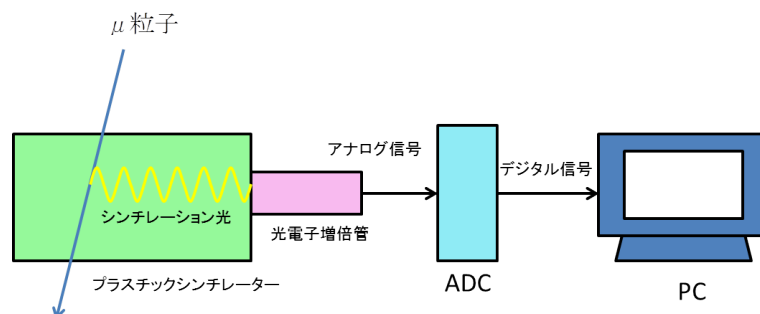


図 12 放射線計測の装置の概要

荷電粒子がシンチレーター内を通ることにより光を放出する。その光は光電子増倍管の光電面に当たると、光電効果により電子を発生する。光電子増倍管でその電子を増幅させ、ADC を用いてデジタル信号に変換し、そのデータを PC 上のプログラムを用いて解析を行うことが、放射線計測の全体の流れとなる。

4.2 μ 粒子と電子の波形の見分け方

本実験では、シンチレーター内で静止した μ 粒子を用いて、 μ 粒子の寿命と電子のエネルギー分布を測定することが課題である。シンチレーターを通過した μ 粒子の信号と、シンチレーター内で静止した後崩壊した μ 粒子の信号には以下で述べるように観測される信号に大きな違いがある。

- μ 粒子が測定用のシンチレーターを通過するときの信号は図 14 に示すように、1 つ山の信号が現れる。

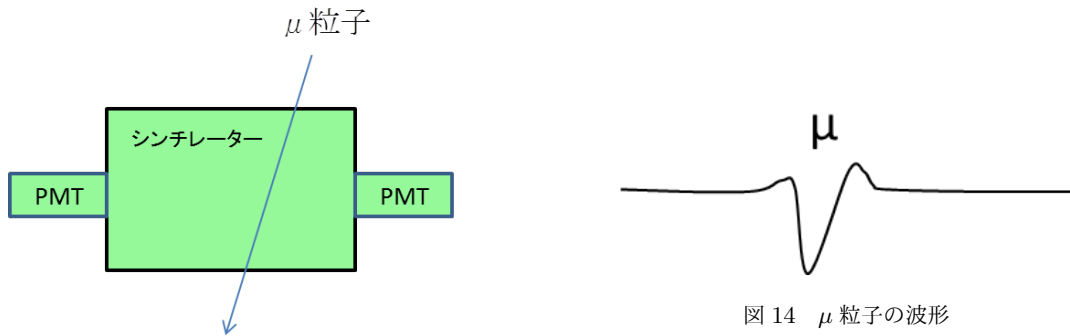


図 13 μ 粒子とシンチレータの様子

- μ 粒子が測定用のシンチレーター内で静止し、その後しばらくしてから崩壊して電子を放出するときの信号は図 16 に示すように、2 つ山の信号が現れる。1 つ目の山は μ 粒子の信号であり、2 つ目の山は電子の信号である。

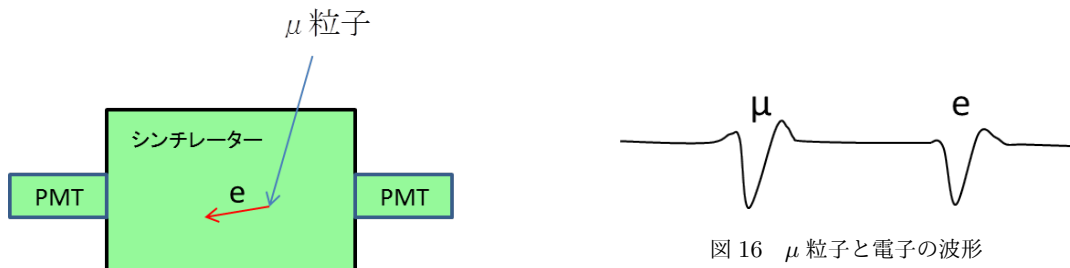


図 15 μ 粒子と電子とシンチレータの様子

μ 粒子の信号が出てきてから電子の信号が出てくるまでの間の平均時間を測ることで、 μ 粒子の寿命が測定できる。

5 セットアップ

5.1 シンチレーションカウンターの配置

本実験で用いたシンチレーションカウンターの配置図を図 17~19 に示す。使用したカウンターは以下の通りである。

- ・大型プラスチックシンチレーター (580mm × 260mm × 250mm) 1 個
- ・プラスチックシンチレーター (582mm × 260mm × 10mm) 2 個
- ・プラスチックシンチレーター (250mm × 131mm × 10mm……直径 60mm の半円の穴がある) 4 個
- ・プラスチックシンチレーター (280mm × 261mm × 10mm) 4 個
- ・ライトガイド 10 個
- ・光電子増倍管 12 個

以下でシンチレーションカウンターの配置図を詳しく説明する。シンチレーションカウンターの全体の配置図と分解した図を図 17 と図 18、図 19 に示す。中心にある 1 番大きなシンチレーター (メインシンチレーターとする) には、ライトガイドなしで 2 つの光電子増倍管 (S1、S2) を直接取り付けました。図中では、分かりやすくするために S1、S2 以外の光電子増倍管は描いていない。

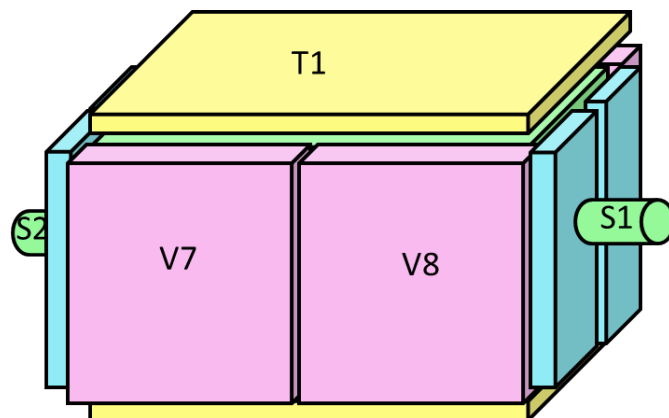


図 17 全体図

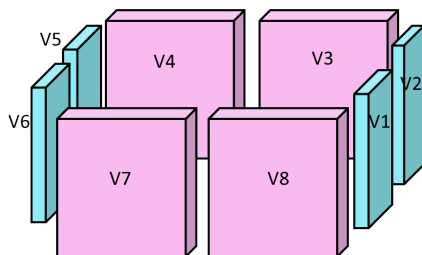


図 18 分解図 1

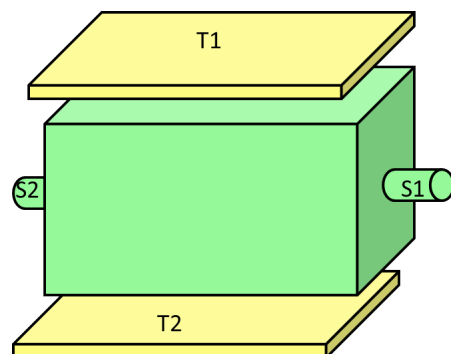


図 19 分解図 2

5.1.1 トリガーカウンター

μ 粒子は宇宙から絶え間なく降り注いでおり、シンチレーターにはあらゆる方向から μ 粒子が入射してくる。しかし、シンチレーターは μ 粒子の信号以外にもノイズや電子などの他の粒子にも反応する可能性があるため μ 粒子の信号と他のノイズの信号を区別する必要がある。そこで、いくつかのカウンターを用いて、信号がきたことを同定する必要がある。これをトリガーという。今回の実験ではメインシンチレーターの上に置いたシンチレーションカウンター T1 と S1 をトリガーカウンターとして使用し、T1、S1 のコインシデンスをトリガー信号として用いた。

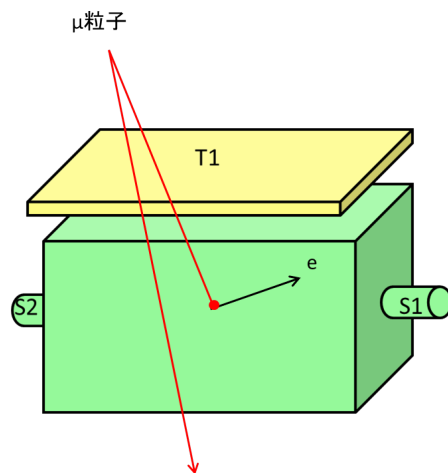


図 20 トリガーカウンターの模式図

図中の、S1、T1 のコインシデンスが ON のとき μ 粒子が入射した条件となる。以下で述べるように、この信号は ADC の Gate 信号や TDC の start 信号に用いられる。

5.1.2 ベトーカーカウンター

今回の実験では、メインシンチレーター内で静止した μ 粒子を識別し、かつ崩壊した電子がメインシンチレーター中で止まった事象を計測したい。 μ 粒子の崩壊後、電子はシンチレーター内をエネルギーを放出しながら通過する。その多くはエネルギーを持ったままシンチレーターから飛び出す。電子のエネルギースペクトラムを計測するために必要なイベントは電子がシンチレーター内で全エネルギーを放出したイベントのみである。そのため、メインシンチレーターから飛び出してしまった電子を測定し、そのイベントについてはデータ処理の段階で取り除く必要がある。その作業をするのがベトーカーカウンターである。

今回の実験では T1、T2、V1~V8 の 10 個のカウンターをベトーカーカウンターとし、その電子のタイミングの ADC の値によって電子がメインシンチレーターの外に飛び出したかを判断した (図 21)。また、メインシンチレーターの下に置いたシンチレーションカウンター T2 はメインシンチレーター内を通過した μ 粒子を識別する目的にも使用した。

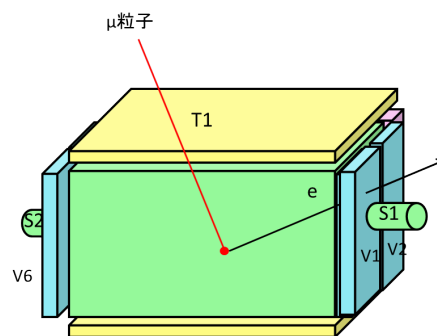


図 21 ベトーカーカウンターの模式図

図中の T1、T2、V1~V8 は電子が来たかどうかを判断する。

5.2 エレクトロニクスへのセットアップ

本実験で用いたエレクトロニクスの全体図を図 22 に示した。以下、各モジュールの役割について順に説明する。

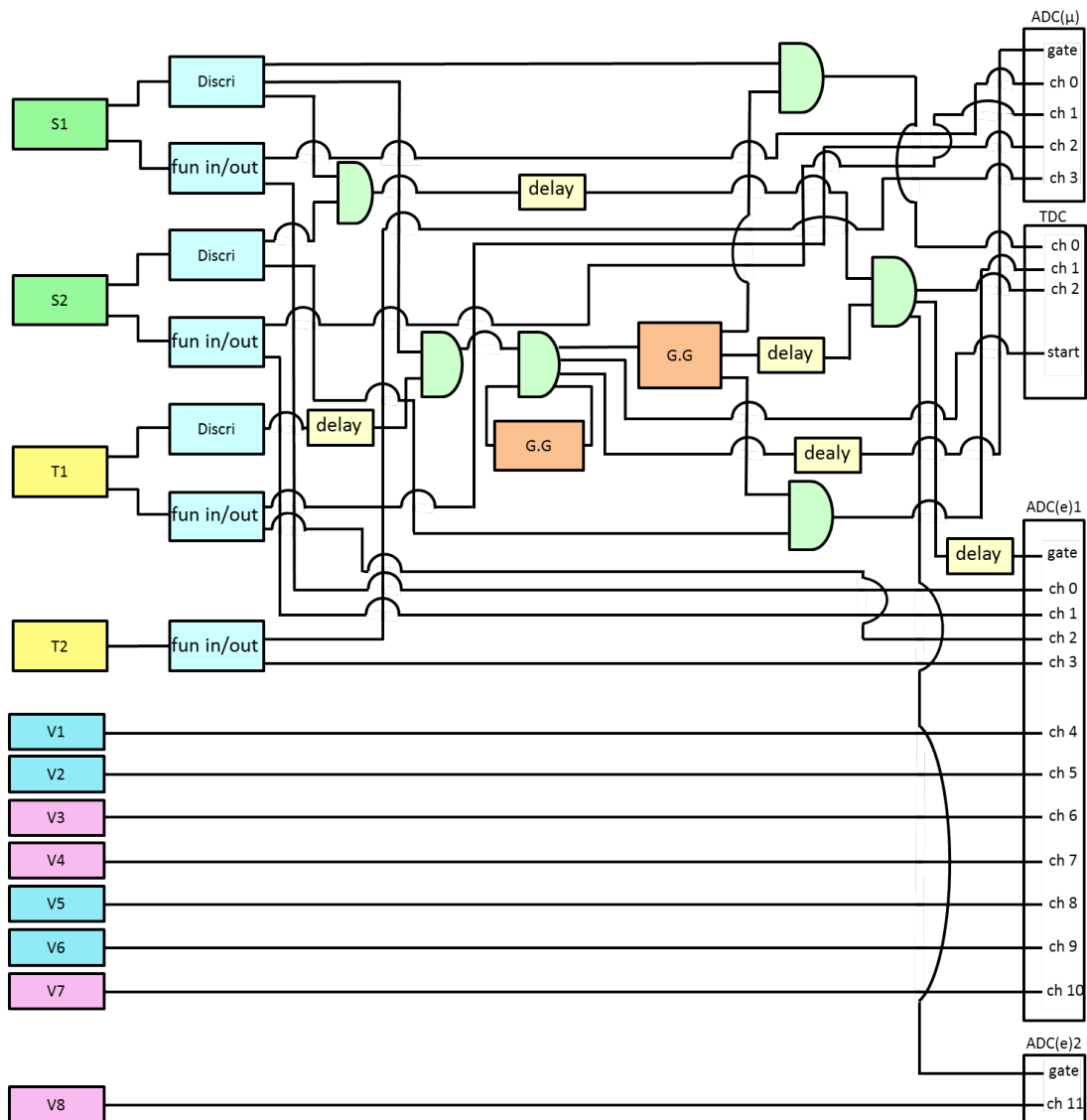


図 22 エレクトロニクスの全体図

5.2.1 NIM(Nuclear Instrument Module)

NIMとは「放射線測定モジュール標準規格 TID-20893」に準拠した標準規格のことであり、信号レベル・コネクタ形状・電源・サイズなどについて規定されている。この規格に準拠した回路は NIM モジュールといわれ、今回の実験で以下の NIM モジュールを使用した。

使用した NIM モジュールについて簡単に説明する。

- ディスクリミネーター (Discriminator)

あらかじめ設定していた threshold (しきい値) よりも大きなシグナルが入力されたとき、その threshold を超えた時刻を起点として方形 (パルス) 波を出力するモジュール。出力パルスは-0.7V と NIM 規格で決められている。今回の実験では threshold 以下のノイズを除去するために用いた。

- コインシデンス (Coincidence)

複数のパルス (最大 4 つ) が同時に入力され、デジタル信号が重なったとき、重なった時刻を起点として、ある一定の時間幅を持ったパルスを出力する (複数の信号の AND をとる) モジュール。

- ゲート・ジェネレーター (Gate and delay Generator)

波形を成形するモジュールの出力信号のパルス幅などを調節でき、Delay 機能も付いているモジュール。

- ファンイン ファンアウト (Fan-in Fan-out)

最大 4 つまでの入力信号についてアナログ和をとり、その結果のアナログ信号を最大 4 ch まで複製して送り出すモジュール。今回の実験では T1、T2、S1、S2 それぞれからの信号を複製するために使用した。

5.3 CAMAC(Computer Aided Measurements And Control)

CAMAC とはモジュール化されたデータを処理するシステム規格のことである。世界中のほとんどの素粒子・原子核研究室やたくさんの工場現場で使用されている。これは U.S の NIM とヨーロッパの ESONE 委員会のジョイントで提供された。

CAMAC は計算機器周辺のデジタル化された情報の処理を機能ごとにモジュール化して行えるようにできている。すなわち、実験装置など外からの情報はプラグイン・ユニットまたはモジュールの画面からパネルからコネクタを通して入り込む。この情報はプラグインの中で処理されると、裏面のプリント基盤エッジを利用したコネクタでクレートと呼ばれるプラグインを収容する箱の裏側の配線（データウェイ）とつながれる。このデータウェイはクレートコントローラが制御するが、たいていはクレートコントローラ自身が計算機の指示に従って制御するようになっている。クレートコントローラはたいてい小型計算機と CAMAC のインターフェースを兼ねる。プラグイン・ユニットは回路配線に使われたプリント基盤自身のエッジが 86 ピンのコネクタとなって飛び出しており、クレートに差し込むと自動的にクレート裏側のコネクタを通して CAMAC のデータウェイと接続され、電源やデータの受け渡し、制御信号の受け渡しがされるようになっている。CAMAC の規則は、アドレス指定であり、C、N、A、F、の 4 つの数字で指定する。

C:crate controller の数。今、crate は一つだけ使っているのので、C=1 とする。

N:station number

A:sub address (モジュール内のアドレス)

F:function

代表的な Function を以下に示す。

F(0),F(2);Read Data

F(8);Test LAM

F(9);Clear LAM

F(24);Disable LAM

F(26);Enable LAM

ここで LAM とは Look At Me の略で、モジュールからデータを読み出し可となったことをコンピューターに知らせる信号のことである。

次に使用した CAMAC のモジュール、ADC と TDC について説明する。

5.3.1 ADC(Analog to Digital Converter)

ADCとは、アナログ電気信号をデジタル電気信号に変換する電子回路のことである。今回はQモードを使用し、プラスチックシンチレーター内で崩壊した μ 粒子と電子からの信号を数値化して解析を行った。

Qモードとは荷電積分型ADCでGateパルスが持続している間の信号を積分した値を測定するモードである。測定される電荷の量は

$$Q = \int_{dt} i dt$$

となる。ここで*i*は信号の電流である。

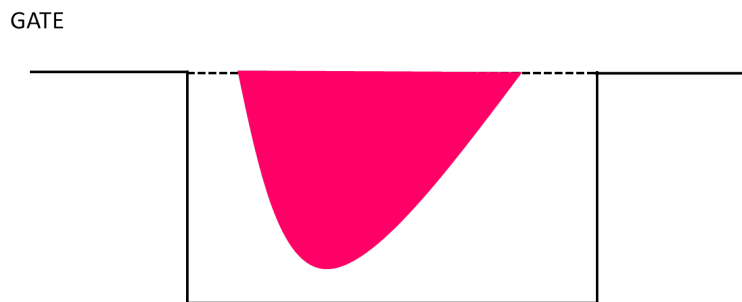


図 23 ADC のしくみ

5.3.2 TDC(Time to Digital Converter)

TDCとは、スタート信号の入力で内部のクロックをスタートさせ、ストップ信号の入力時のクロックの出力数をカウントする電子回路のことで、図24のようにスタート信号からストップ信号までの時間を測定する。なお今回実験で使用したTDCはREPIC社のRpc-060型(25psec/count select)で最大118nsまで測定できる。

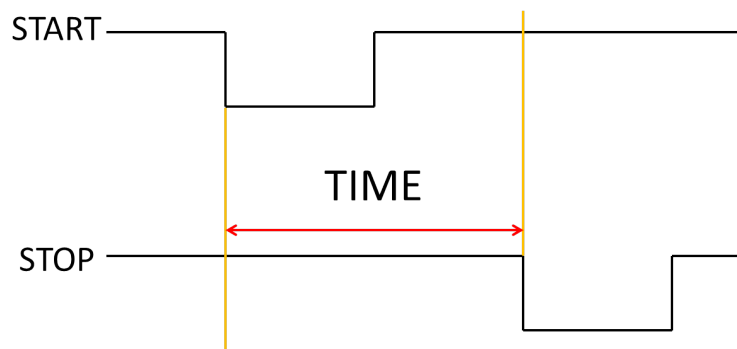


図 24 TDC のしくみ

5.4 各 CAMAC のモジュールについてのセットアップ

本実験では、次の3つの量の測定が必要であり、以下に示す。

- (1) μ 粒子のエネルギー測定
- (2) μ 粒子が入射してから電子が放出されるまでの時間の測定 (μ 粒子の寿命測定)
- (3) 電子エネルギースペクトラムの測定

全セットアップを図 22 に示したが、これでは詳細が分からないため、各測定についてどのようなセットアップになっているかを以下に示す。

5.4.1 μ 粒子のエネルギー測定 (ADC(μ)) のセットアップ)

μ 粒子のエネルギーを測定する ADC を ADC (μ) とする。ADC (μ) のセットアップ図を図 25 に示す。

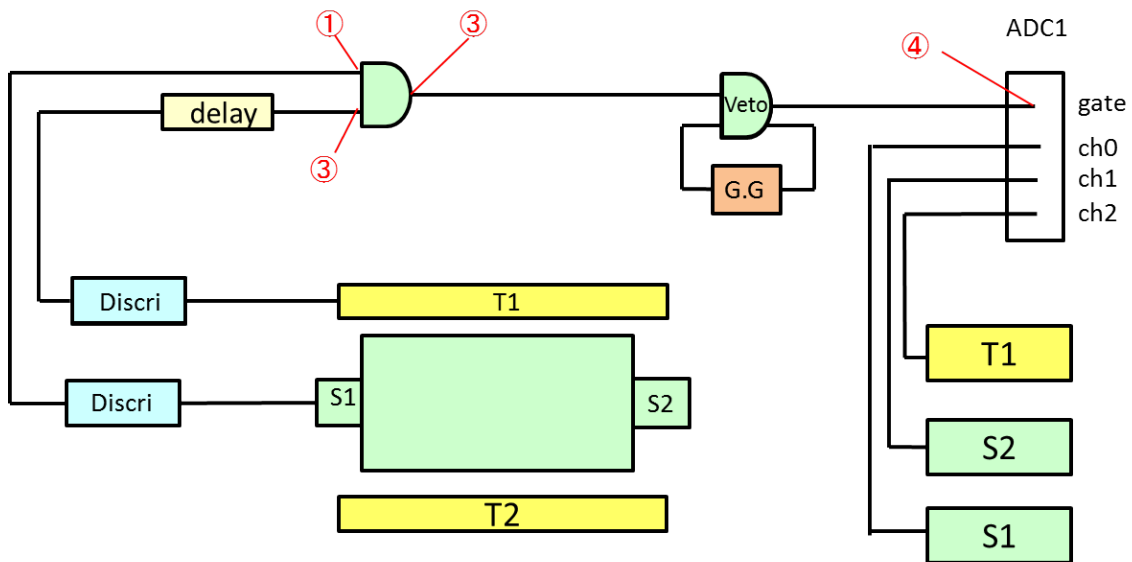


図 25 ADC(μ) のセットアップ

μ 粒子が入射したと判定する条件は T1 と S1 のコインシデンスが ON のときとする。このコインシデンスで gate をつくり、 μ 粒子のエネルギーを測定した。

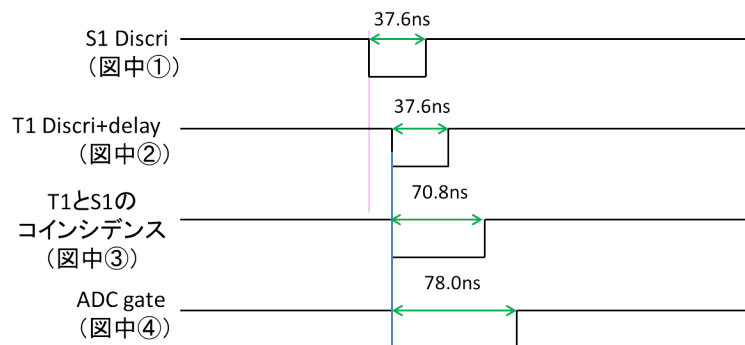


図 26 ADC(μ) のセットアップにおける各モジュールの入力または出力信号

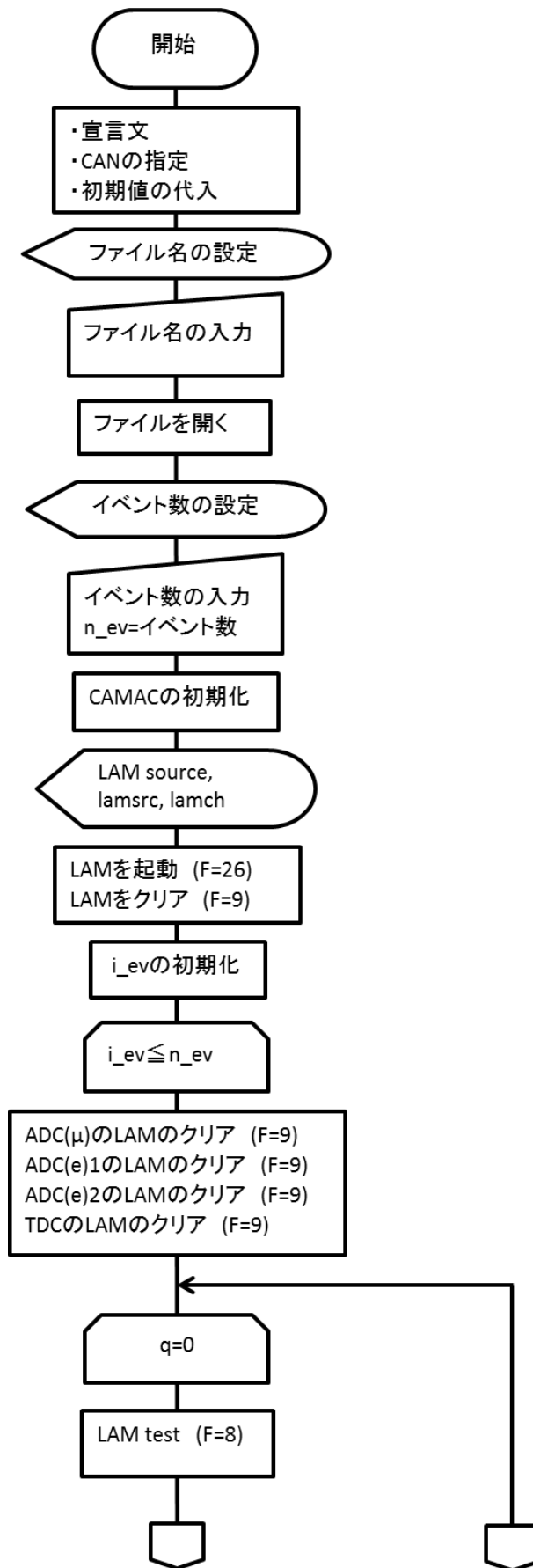
5.5 エレクトロニクスの設定

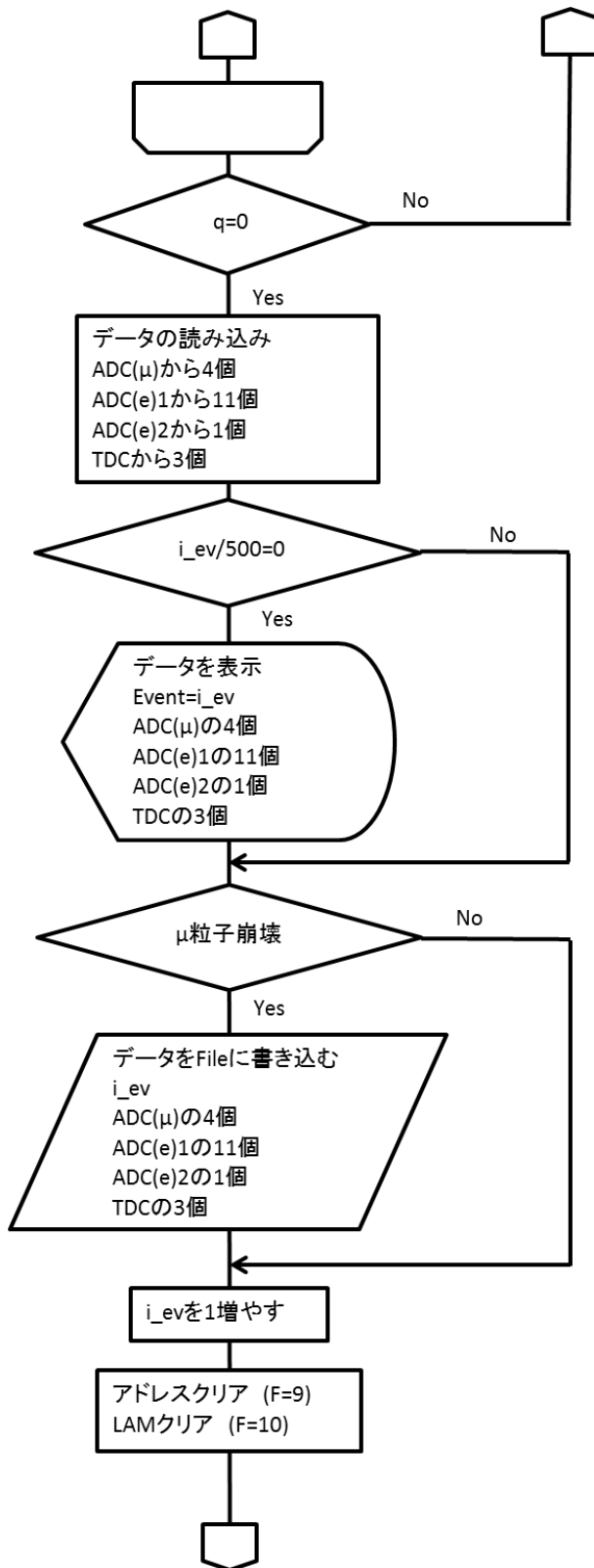
今回の実験では、前で示したモジュールを用いて、各々のカウンターからの信号を測定した。この時のカウンターに用いた光電子増倍管の型番、HV、ディスクリミネーターの Threshold、各カウンターの ADC(μ)、ADC(e) のペDESTALの値を表にまとめる。

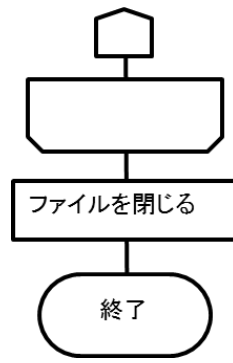
カウンター	PMT 型番	HV(V)	Threshold(mV)	ペDESTAL (ADC(μ))	ペDESTAL (ADC(e))
S1	H1161	1880	102	117	56
S2	H1161	1800	101	110	62
T1	H1161	2250	99	101	55
T2	H7195	1770		96	58
V1	H1161	1970			66
V2	H7195UV	1660			62
V3	H1161	1870			62
V4	H1161	2440			63
V5	H1161	2400			64
V6	H1161	2030			65
V7	H7195	2210			71
V8	H1161	2025			77

5.6 データ収集プログラム

今回の実験では、CAMAC 空のデータ収集プログラムに C 言語を用いる。次のページに用いたプログラムのフローチャートを示す。また、プログラムについては、巻末の付録を参照。プログラムの大まかな流れは、初めにコンピューターから ADC にテスト LAM の信号を送る。LAN OK の Q 信号が帰ってきたら、ADC、TDC のデータを読み込み、ファイルに書き込んだ後データと LAM のクリアを行う。そして再び ADC にテスト LAM この信号を送る。これをイベントの数だけ繰り返す。







6 データ解析

6.1 μ 粒子の寿命測定

6.1.1 データの解析方法

μ 粒子の寿命測定にあたって行うべきことは、主に以下の三つである。

1. イベントの選別
2. 測定で得られた TDC 分布に関数を FIT
3. TDC の時間較正

これらについて以下で説明する。

1. イベントの選別

測定において得られたデータのうち使用するイベントは「 μ 粒子がメインシンチレーター内で崩壊したイベント」である。このイベントを用いて解析を行う。

2. 測定で得られた TDC 分布に関数を FIT

原理で示したように得られる TDC 分布の関数は以下ようになる。

$$\frac{dN_{\text{崩壊}}}{dt} = \frac{N_0}{\tau} \exp\left(-\frac{t}{\tau}\right) \quad (33)$$

ここで、以下の関数を得られる TDC 分布に FIT させる。

$$y(t) = P_1 \exp\left(-\frac{t}{P_2}\right) \quad (34)$$

上記の式 (33) と式 (34) を比較すると式 (34) の P_2 が式 (33) の寿命 τ に比例していることがわかる。TDC 分布に式 (34) を FIT させ P_2 を出すことにより、寿命を求める。

3. TDC の時間較正

上記式 (34) で得られる P_2 の値は TDC のカウント数であり、このカウント数を時間の単位に変える必要がある。そこで、TDC1 カウントあたりの時間を求め、その時間と P_2 の値をかけることによって、寿命 τ を算出する。

以上のことを順に行い、解析した結果について述べる。

6.1.2 イベントの選別

今回の実験では、約1ヶ月の間に100万イベント（20時間）ずつデータを取りだめたので、 μ 粒子がメインシンチレーター内に入射したイベント数は2324万9253となった。全データのうち今回使用するイベントは「 μ 粒子がメインシンチレーター内で崩壊したイベント」である。 μ 粒子が崩壊したイベントの条件と、その条件の選出方法について以下で述べる。

条件

1. μ 粒子がメインシンチレーター内に入射
2. μ 粒子がメインシンチレーター内で崩壊

条件の選出方法

1. μ 粒子がメインシンチレーター内に入射
T1とS1のコインシデンスがONである。
2. μ 粒子がメインシンチレーター内で崩壊
シンチレーター内で電子が放出された、つまりS1(e)のカウンターがなっているとみなされる値を上回っているイベントである。具体的な値を決めるために μ 粒子が入射したときのS1(e)のADC分布を測定した。結果を以下に示す。

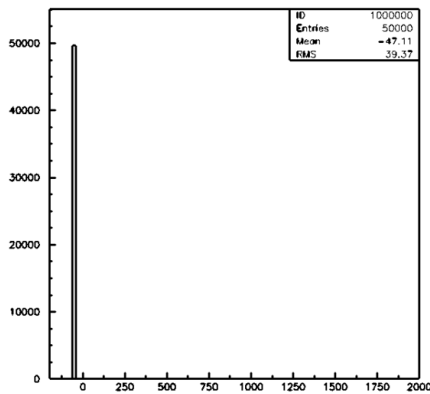


図 31 μ 粒子が入射したときのS1(e)

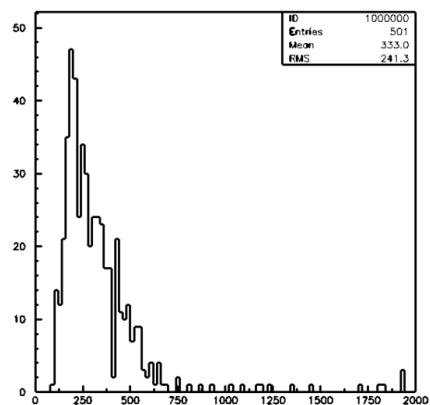


図 32 μ 粒子が入射し、電子に崩壊したときのみのS1(e)

図 31 から $S1(e) < 10$ をカットした図が図 32 である。よって $S1(e) > 10$ のときに μ 粒子が崩壊したとみなす。各条件でのイベント数を以下の表に示す。

1. μ 粒子がメインシンチレーター内に入射	23249253
2. μ 粒子がメインシンチレーター内で崩壊	183569

6.1.3 TDC 分布に関数を FIT

今回の測定で得られた TDC 分布に FIT させた関数は以下の二つである。

$$y(t) = P_1 \exp\left(-\frac{t}{P_2}\right) \quad (35)$$

$$y(t) = P_1 \exp\left(-\frac{t}{P_2}\right) + P_3 \quad (36)$$

得られる TDC 分布に上記の関数を FIT させ P_2 の値を出すことにより寿命を求める。TDC 分布のバックグラウンドの存在について考慮するため、 P_3 のパラメータを加えた式 (36) での FIT も試みた。以下でこの 2 式での FIT の結果を示す。また TDC 分布の度数が 0 もしくはかなり小さいものを省くため、TDC カウント数 100~2000 の間で FIT を行った。

・ 2 パラメータの式での FIT について

得られた TDC 分布に、式 (35) を FIT させた図を以下に示す。

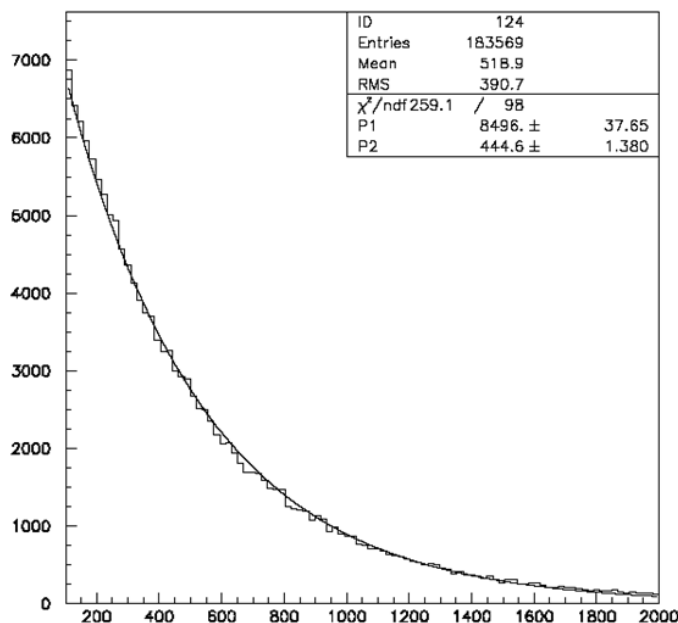


図 33 TDC 分布に式 (35) を FIT させた図

FIT を行った図 33 より、得られた結果を以下に示す。

	値	誤差
P_1	8496	37.65
P_2	444.6	1.38
χ^2/ndf	259.1/98	

これより、 P_2 の値は 444.6 ± 1.38 だということがわかった。

・3パラメーターの式での FIT について得られた TDC 分布に、式 (36) を FIT させた図を以下に示す。

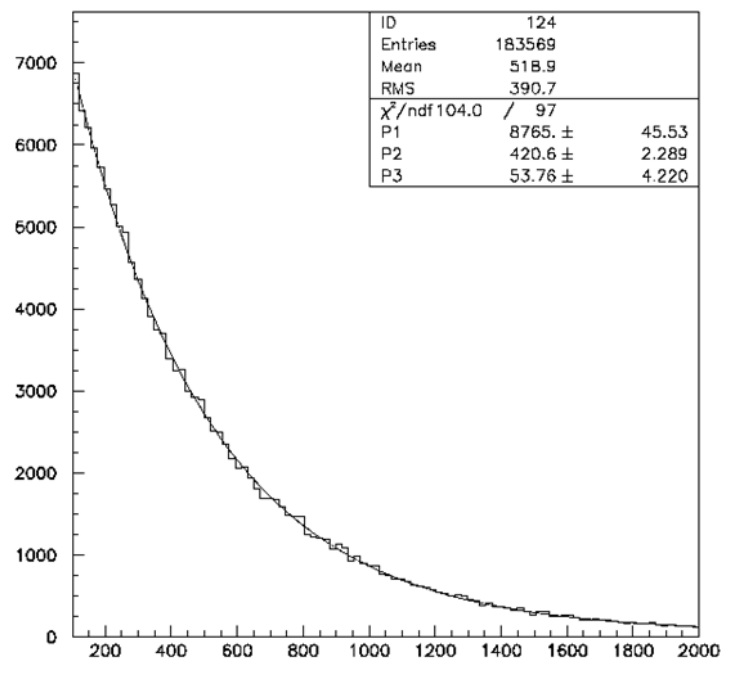


図 34 TDC 分布に式 (36) を FIT させた図

FIT を行った図 34 より、得られた結果を以下に示す。

	値	誤差
P_1	8765	45.53
P_2	420.6	2.289
P_3	53.76	4.22
x^2/ndf	104.0/97	

これより、 P_3 を考慮した時の P_2 の値は 420.6 ± 2.289 だということがわかった。

6.1.4 TDC の時間較正

TDC の 1 カウントあたりの時間を出すために、Clock Generator からの入力信号を TDC の start と stop にいれ、start と stop の間の時間を測定した。また stop には start よりも少し delay させた信号をいれそれぞれ 1000 イベントずつ測定した。TDC の時間較正のロジックを図 35 に示す。

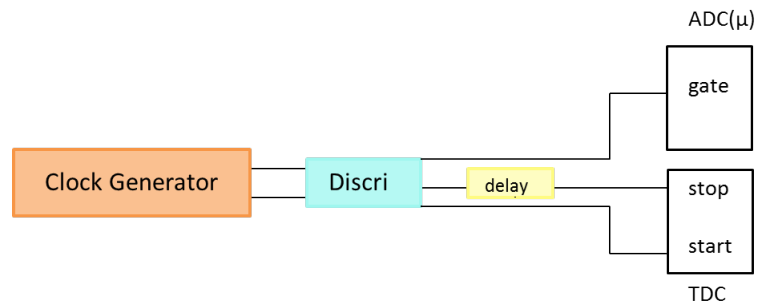


図 35 TDC 時間較正のロジック

TDC で測定した時間 (delay) と、TDC のカウント数の関係を表に示す。カウント数は 1000 イベントのデータの平均値である。

delay(nsec)	カウント数
10	3.72
50	11.605
59.8	11.092
100	19.17
100	21.828
150	32.276
200	42.704
270	53.064
250	53.296
300	63.996
370	72.757
350	74.839
506	99.814
606	120.402
722	144.003
830	165.211
984	196.995
1028	205.537
1436	286.029
1636	326.426
1888	376.71
2192	435.798
2392	477.707
2568	510.501
2760	549.799
3072	614.518
3296	658.075
3552	710.191
3720	743.386
3960	792.574

得られた結果をグラフにしたものを図 36 に示す。

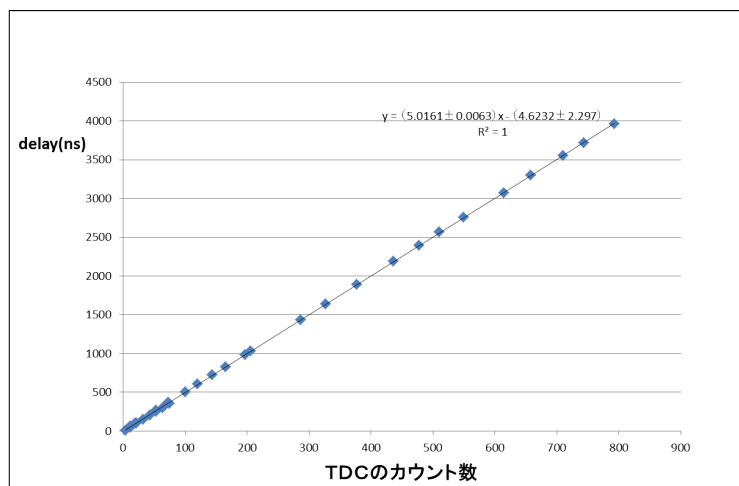


図 36 TDC 時間較正

図 36 より、式の傾きが 5.0161 ± 0.0063 なので TDC1 カウントあたり 5.0161 ± 0.0063 (ns) であることがわかった。また、誤差である ± 0.0063 (ns) は最小二乗法により算出している。

6.1.5 実験結果

得られた結果と寿命算出の際に用いる式を以下に示す。

	2パラメーターの式(式(35))	3パラメーターの式(式(36))
P_2	444.6 ± 1.38	420.6 ± 2.289

TDCの1カウントあたりの時間(a) $a = 5.0161 \pm 0.0063$ (ns)

$$(\text{寿命}\tau) = P_2 \times a(\text{ns}) \quad (37)$$

また寿命 τ の誤差は、誤差伝播の式

$$\Delta W = \sqrt{\left(\frac{\partial f}{\partial X}\right)^2(\Delta X)^2 + \left(\frac{\partial f}{\partial Y}\right)^2(\Delta Y)^2} \quad (38)$$

を用いて、

$$(\text{寿命}\tau\text{の誤差}) = \sqrt{(P_2)^2(\Delta a)^2 + (a)^2(\Delta P_2)^2} \quad (39)$$

が得られる。

これらより、 μ 粒子の寿命算出を行った結果を以下に示す。

・2パラメーターの式でFITした際の寿命算出の結果

$$\tau = 2.23 \pm 0.01(\mu\text{s})$$

・3パラメーターの式でFITした際の寿命算出の結果

$$\tau = 2.11 \pm 0.01(\mu\text{s})$$

結果より文献値との差は約1.5%、4%となった。

6.2 電子のエネルギースペクトラムの計測

6.2.1 データの解析方法

μ 粒子の崩壊により生成される電子のエネルギー分布の測定のために行うことは、主に次の二つである。

1. イベントの選別
2. ADC のエネルギー較正

概要は以下の通りである。

1. イベントの選別

測定により得られたデータの内、今回使用するデータは、「 μ 粒子が崩壊し、かつ放出された電子がメインシンチレーター内にとどまったイベント」である。なので、このイベントを用いて解析を行う。

2. ADC のエネルギー補正

電子のエネルギー分布を測定する際、S1 と S2 の値の和の ADC 分布を用いる。この ADC 分布にエネルギー較正を行い、結果の解析を行う。

6.2.2 イベントの選別

測定により得られたデータの内、今回使用するデータは、「 μ 粒子が崩壊し、かつ放出された電子がメインシンチレーター内にとどまったイベント」である。このイベントの条件と、条件の選出方法を以下で述べる。

条件

1. μ 粒子がメインシンチレーター内に入射
2. μ 粒子がメインシンチレーター内で崩壊
3. μ 粒子の崩壊後、放出された電子がメインシンチレーター内にとどまっている。つまりベトーカーがなっていない。

このイベントが選出されるように、ヒストグラムを作成するプログラム内で指定した。

条件の選出方法

1. μ 粒子がメインシンチレーター内に入射

6.1.2 で示した条件を用いる。

2. μ 粒子がメインシンチレーター内で崩壊

6.1.2 で示した条件を用いる。

3. μ 粒子の崩壊後、放出された電子がメインシンチレーター内にとどまっている。

μ 粒子の崩壊後、放出された電子がメインシンチレーター内にとどまっている条件はベトーカーがなっていない。つまり、T1、T2、V1~V8 の ADC(e)1、ADC(e)2 の値が、カウンターが反応していないとみなされる値を下回っているイベントである。その値を具体的に決めるために、 μ 粒子が崩壊したときの、T1、T2、V1~V8 の ADC(e)1、ADC(e)2 の ADC 分布を測定した。以下に測定結果を示す。図 37~図 44 が V1~V8 で、図 45、46 が T1、T2 である。

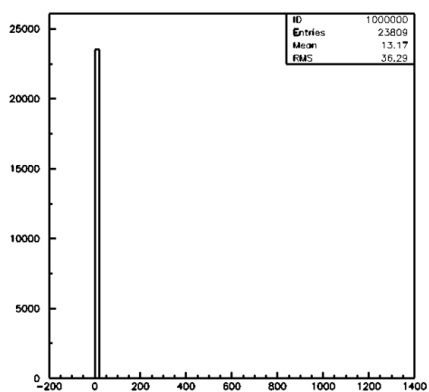


图 37 V1

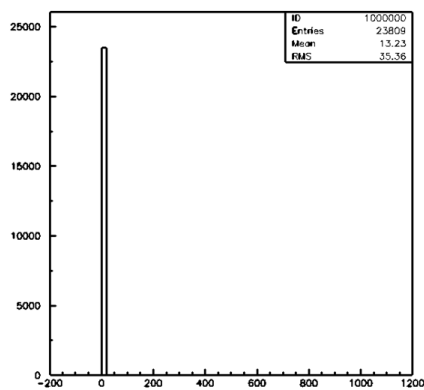


图 38 V2

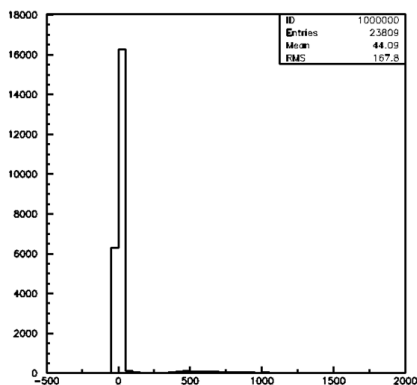


图 39 V3

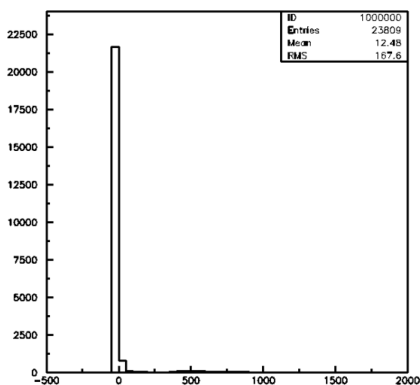


图 40 V4

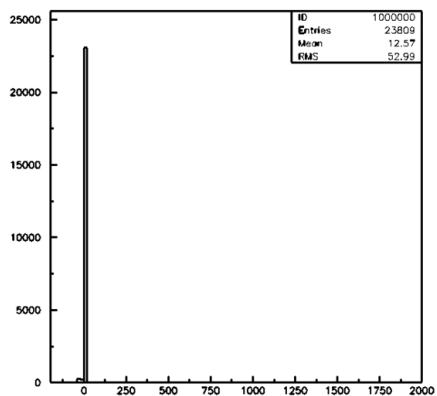


图 41 V5

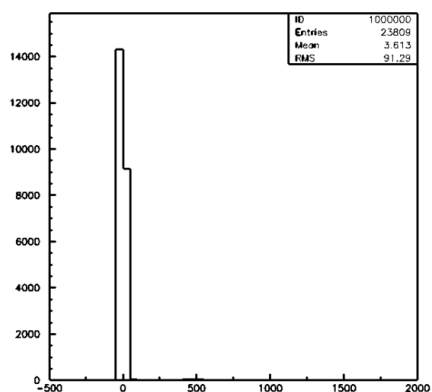


图 42 V6

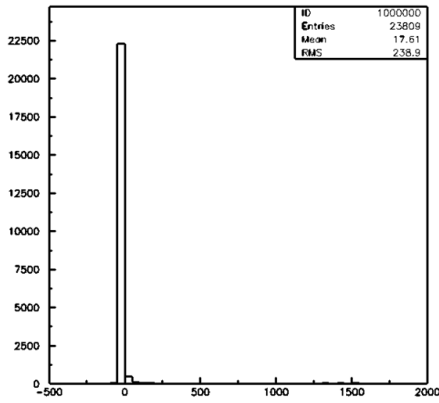


図 43 V7

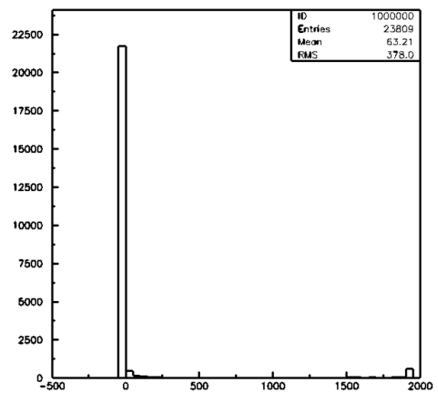


図 44 V8

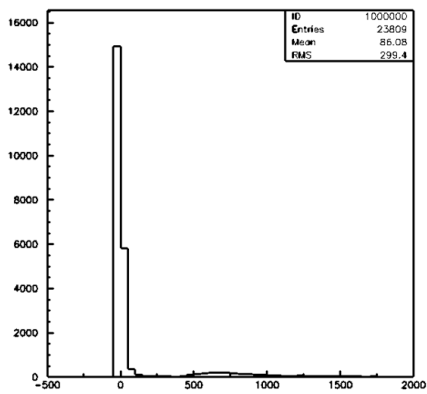


図 45 T1(e)

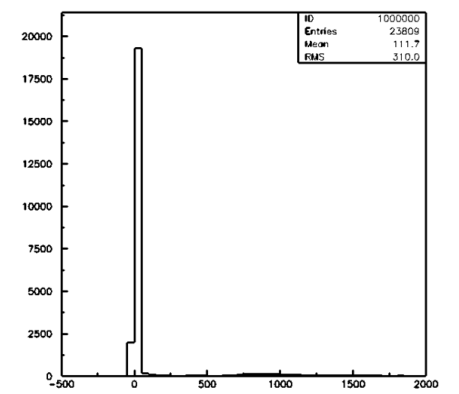


図 46 T2(e)

よって全てのベトーカウンターが、
 $V1 \sim V8 < 50$ 、 $T1(e)$ 、 $T2(e) < 100$

を満たすとき電子がメインシンチレーター内でとどまっているとみなす。

各条件でのイベント数と μ 粒子がメインシンチレーター内に入射したときを全体としたときの割合を以下の表に示す。

	イベント数	割合
μ 粒子がメインシンチレーター内に入射	23249253	
μ 粒子がメインシンチレーター内で崩壊	183569	0.78%
電子がメインシンチレーター内でとどまっている	105382	0.45%

6.2.3 電子のエネルギースペクトラム

前記の条件 1~3 において得られた電子のエネルギースペクトラムを以下に示す。

1. μ 粒子が崩壊して放出された全ての電子のエネルギー分布
2. 電子がメインシンチレーターから飛び出した時の分布
3. 電子がメインシンチレーター内でとどまっているときの分布

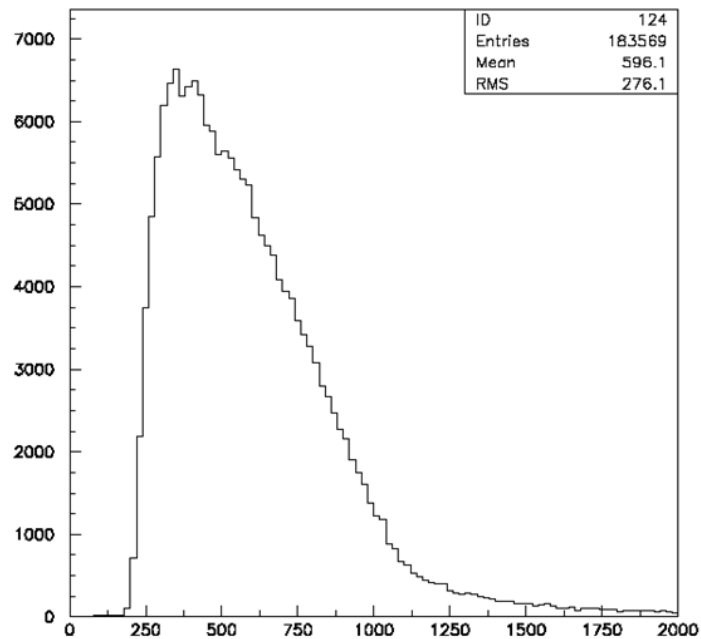


図 47 μ 粒子が崩壊して放出された全ての電子のエネルギースペクトラム

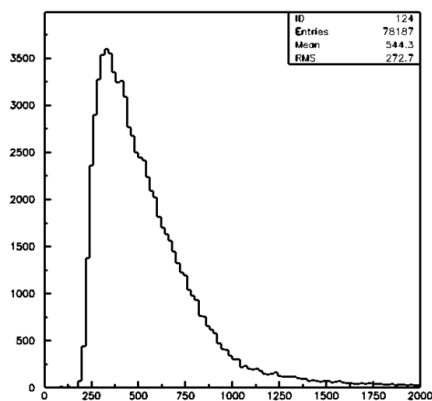


図 48 電子がメインシンチレーターから飛び出した時のスペクトラム

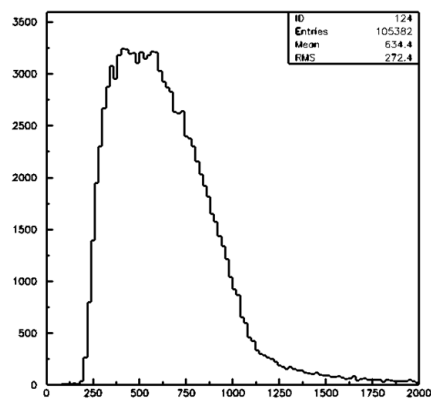


図 49 電子がメインシンチレーター内でとどまっているときの分布

これらのすべては、S1、S2 の値の和の ADC(e)1 分布である。

縦軸は度数、横軸は ADC(e)1 のカウント数である。

6.2.4 ADC のエネルギー較正

S1、S2 の値の和の ADC(e)1 分布について、エネルギー較正を行う。使用した μ 粒子が入射したデータは 10 万イベントで、このうち「 μ 粒子がメインシンチレーターを崩壊せずに通過したイベント」のみを使用する。また、宇宙線の角度分布と運動量分布をここでは無視している。また、ADC による個体差をなくするために ADC(e)1 を用いている。このイベントの条件と、条件の選出方法を以下で示す。

条件

1. μ 粒子がメインシンチレーター内に入射
2. μ 粒子がメインシンチレーター内で崩壊しない
3. μ 粒子が T2 を通過

このイベントが選出されるように、ヒストグラムを作成するプログラム内で指定した。

条件の選別方法

1. μ 粒子がメインシンチレーター内に入射
6.1.2 で示した条件を用いる。
2. μ 粒子がメインシンチレーター内で崩壊しない
6.1.2 で示した条件を用いて導く。
3. μ 粒子が T2 を通過

μ 粒子が T2 を通過した条件は、T2 の ADC(e)1 の値が、ベトーカーンターがなったとみなされる値を上回ったということである。その値を具体的に決めるために、 μ 粒子が入射したときの ADC(e)1 の分布を測定した。図 50 に測定結果を示す。

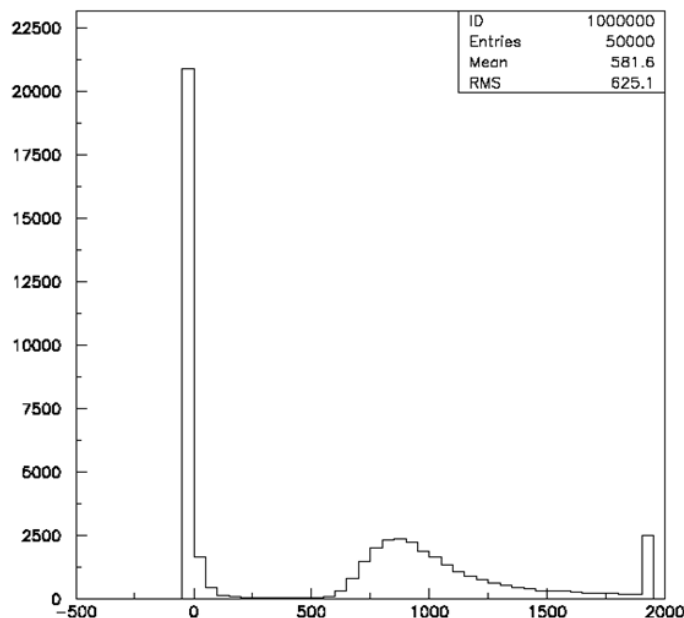


図 50 μ 粒子が入射したときの T2 の ADC(e)1 分布

よってカウント数 500 以上を T2 がなった（通過した）とみなす。

よって、上記の条件 1~3 において得られたイベント数は 49079 であった。測定結果を図 51 に示す。

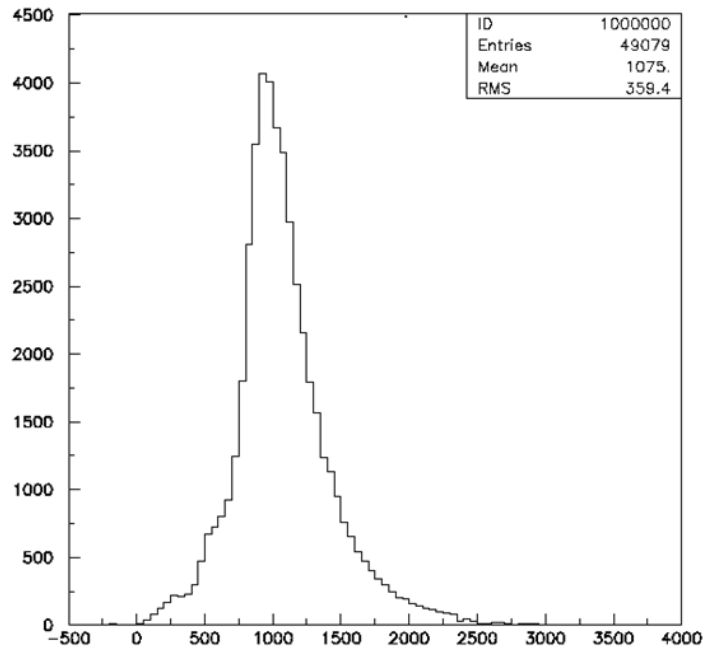


図 51 μ 粒子が入射したときの $S1(\mu)$ と $S2(\mu)$ の値の和の ADC 分布

図 51 より、ADC(e) のカウント数が、900 あたりに分布のピークがあることがわかる。ここで、原理で示したように、 μ 粒子がメインシンチレーターを垂直に通過したときのエネルギー損失は約 53MeV である。よって、図 51 のピークの位置であるカウント数 900 があたりのエネルギーが約 53MeV であることがわかった。

6.2.5 実験結果

電子のエネルギー分布の理論値と測定結果を以下で示す。

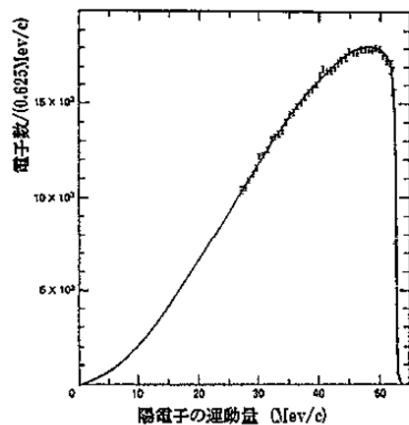


図 52 電子のエネルギースペクトラムの理論値

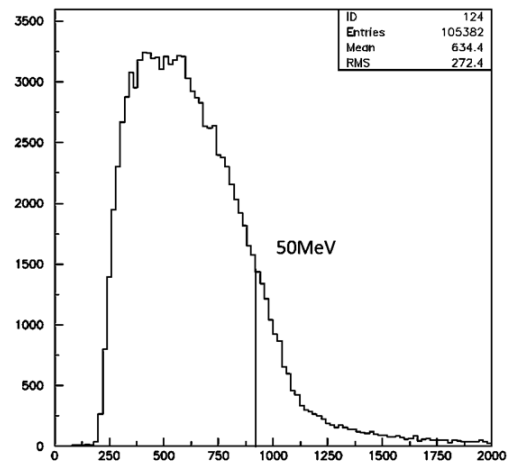


図 53 電子のエネルギースペクトラムの測定結果

図 52 と図 53 より、理論値の分布のピークが約 50MeV であるのに対し、測定結果の分布のピークはそれよりも低い位置にあり、エネルギーの値が約 50MeV であるカウント数 900 あたりのイベント数は、かなり低い結果となった。

7 考察

7.1 寿命測定 of 考察

μ 粒子の寿命測定

P_3 を考慮しなかった場合、 μ 粒子の寿命 τ は

$$\tau = 2.23 \pm 0.01(\mu s)$$

と本実験で測定された。理論値は、国立天文台編・平成 24 年・第 85 冊・理科年表より

$$\tau = 2.19703 \pm 0.00002(\mu s)$$

であり、 P_3 を考慮しなかった場合、理論値と測定値の差は約 1.5 % となった。

P_3 を考慮した場合、 μ 粒子の寿命 τ は

$$\tau = 2.11 \pm 0.01(\mu s)$$

と本実験で測定された。

したがって、 P_3 を考慮した時、理論値と測定値の差は約 4.0 % となった。

P_3 も考慮して FIT を行った理由は、バックグラウンドの効果も考慮するためである。そのため TDC のカウント数を 2000 までではなく、7000 までにして測定結果を図 54 を示した。

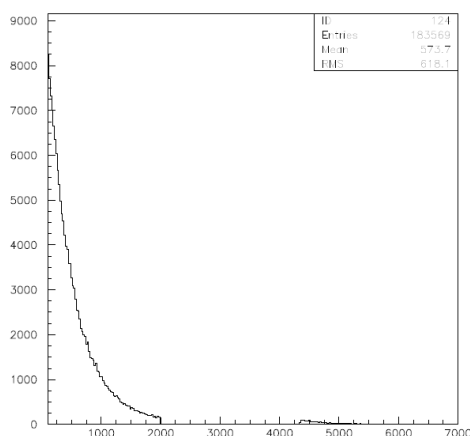


図 54 TDC 分布に P_3 を考慮させて 7000 までで測定した図

図 54 からわかるように、TDC のカウント数 6000 付近でイベント数がほぼ 0 となる。これより、バックグラウンドの効果は考慮する必要がなく、 P_3 も考慮する必要がないことがわかった。よって、 P_3 を考慮しなかった場合の τ の結果についてのみ考える。

理論値と測定値の差が 1.5 % になった理由として考えられるのは、TDC の分布に式 (29) を FIT させる際の範囲の取り方である。なぜなら、範囲の取り方によって P_2 の値が変化したからである。カウント数が 0 付近ではふるまいが式 (29) とは異なり、その場合 FIT させる範囲から除く必要がある。

次にこの効果について考える。

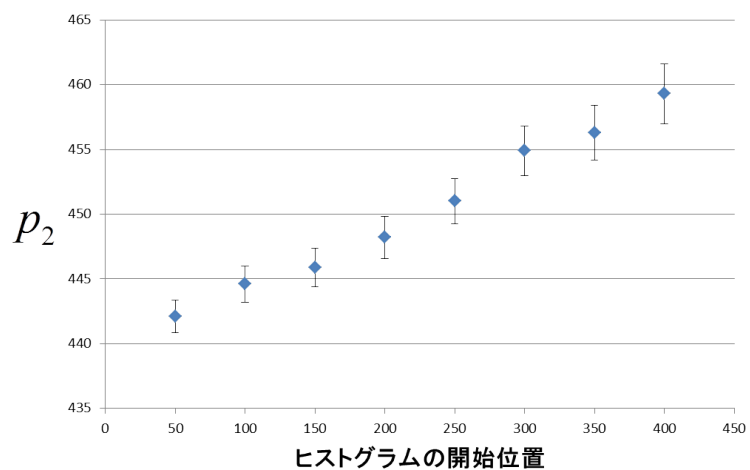


図 55 TDC 分布に FIT させる際の範囲による P_2 の変化

図 55 よりヒストグラムの開始位置によって P_2 の値が変化していることが分かった。なのでヒストグラムの開始位置をどこにするかを決定する方法を今後考えるべきである。

7.2 電子エネルギースペクトラムの測定の考察

実験結果より、電子エネルギー分布の理論値と測定結果には違いがみられた。そこで、この違いがみられた原因としてあげられる2つの効果について考える。

7.2.1 光の減衰の効果の検討

メインシンチレーター内を電子が通過し、発光して出たシンチレーション光がS1またはS2の光電子増倍管に届くまでの距離が長いと、光の減衰の効果が表れる可能性がある。S1とS2で測定されるエネルギーは等しくなるはずである。しかし、光の減衰があると、発光源からS1またはS2までの距離に差があるとき、測定されるエネルギーにも差が出てくると考えられる。そのため、S1とS2で測定された、電子のエネルギーの相関図により、光の減衰の効果の有無について調べた。測定結果を図56に示す。

図56の縦軸がS1(e)、横軸がS2(e)である。

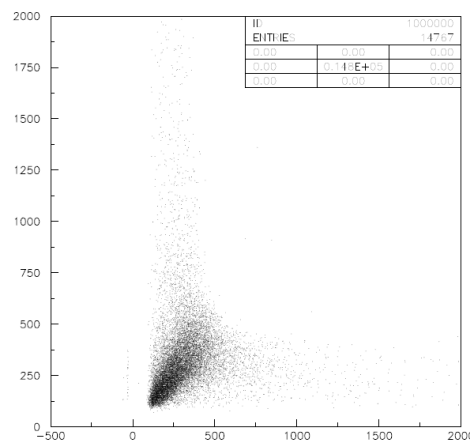


図56 S1(e)とS2(e)で測定された電子のエネルギーの相関図

ここで、光の減衰の効果がある場合に予想される相関図を図57に示す。

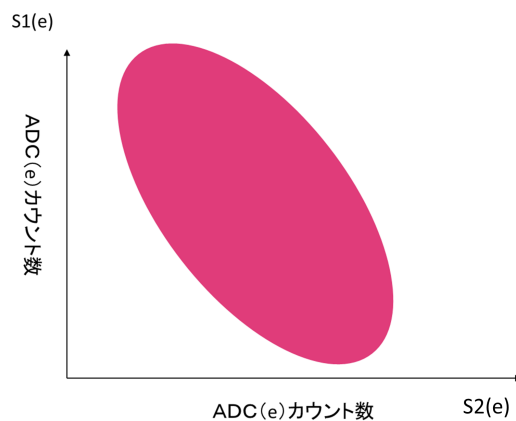


図57 光の減衰の効果がある場合に予想される相関図

光の減衰の効果があると、図57の赤の領域に多く分布する。図56と図57を比較すると測定結果の分布は図57の赤の領域には多く分布していないように見える。そのため、光の減衰の効果は小さいと考えられる。

7.2.2 制動放射の効果の検討

シンチレーター内を走る電子のエネルギーが十分に高い場合、制動放射によってエネルギーの一部を γ 線として放出する。この γ 線が検出器の外に出てしまうと測定したエネルギーはもともと電子が持っていたエネルギーより低くなる。ここで、シンチレーター内を走る電子が制動放射によって γ 線を放出しエネルギーの一部を失う効果について考える。これを調べるために今回は Geant4 を用いて μ 粒子が崩壊する際のシミュレーションを行った。

1. Geant4

Geant4 とは、粒子と物質の相互作用のシミュレーションを行うプログラムである。1994 年に CERN と日本などが国際協力開発に合意し、1998 年に完成した。C++ オブジェクト指向技術で、大規模プログラムを多人数で分担構築することに成功した。特徴としては、幾何学的な記述が優れ、新しい反応過程組み込みが容易で、ソースが公開されているところである。LHC、高エネルギーのほか、粒子線治療、宇宙ステーション、火星基地などの宇宙計画、宇宙線、原子核、地下実験などへの応用に用いられている。Geant4 の構造の概要と主なプログラムは付録参照。

2. シミュレーション方法

Geant4 を用いた今回のシミュレーションは μ 粒子をシンチレーターに入射し、 e 、 ν_μ 、 $\bar{\nu}_e$ に崩壊する事象とする。まず、入射する μ 粒子の運動エネルギーを次に示す方法で決定する。6.2.4 と同様に $ADC(\mu)$ において、 μ 粒子がメインシンチレーターを通り抜けた時の $S1(\mu)$ と $S2(\mu)$ の和の分布を示す。

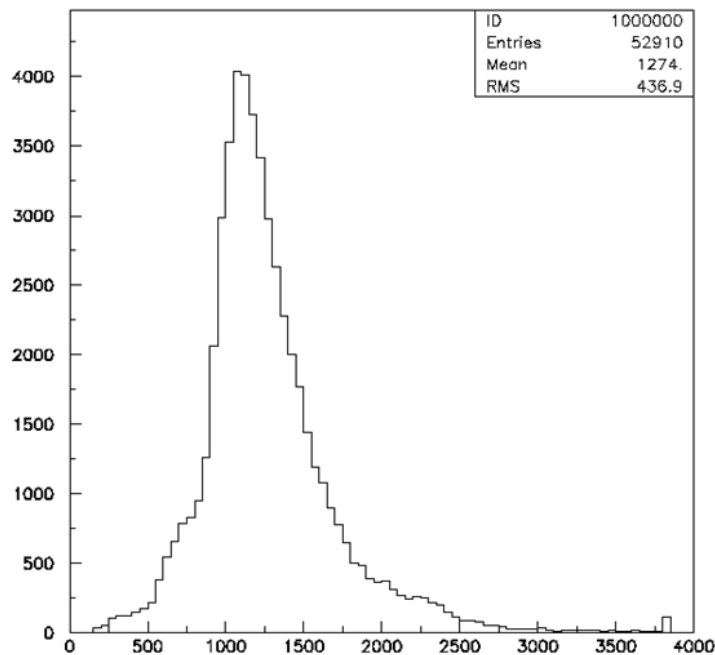


図 58 μ 粒子が入射したときの S1 と S2 の値の和の ADC 分布

図 58 より、 $ADC(\mu)$ のカウント数が、1100 付近に分布のピークがあることがわかる。ここで、原理で示したように、 μ 粒子がメインシンチレーターを垂直に通過したときのエネルギー損失は約 53MeV である。よって、図 58 のピークの位置であるカウント数 1100 付近のエネルギーが約 53MeV であることがわかった。

μ 粒子が電子に崩壊してメインシンチレーターにとどまったときの $S1(\mu)$ と $S2(\mu)$ の和の ADC 分布を以下で示す。

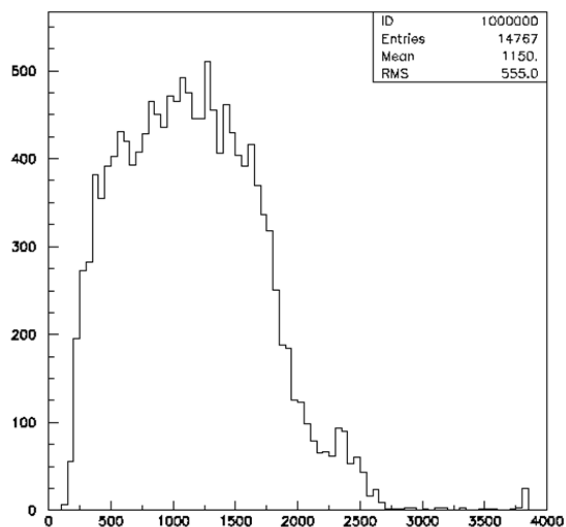


図 59 μ 粒子が電子に崩壊してメインシンチレーターにとどまったときの $S1(\mu)$ と $S2(\mu)$ の和の ADC 分布

図 59 の分布のピークが 1100 付近にあり、その場所でのエネルギーの値が約 50MeV であると図 58 からわかる。よって、入射する μ 粒子の運動エネルギーは 50MeV と決定する。

また、 μ 粒子の寿命と μ 粒子が電子に崩壊した後の電子のエネルギーとの相関関係を図 60 に示す。

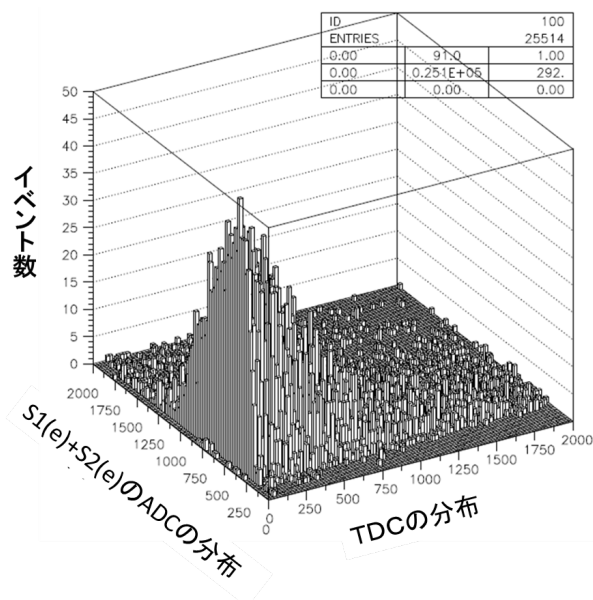


図 60 μ 粒子が電子に崩壊してメインシンチレーターにとどまったときの $S1(\mu)$ と $S2(\mu)$ の和の ADC 分布と μ 粒子の寿命の相関関係

図 60 より、 μ 粒子の寿命と μ 粒子が電子に崩壊した後の電子のエネルギーとの相関関係はみられない。

3. シミュレーション結果

運動エネルギー 50MeV の μ 粒子を 10 万ビーム入射したときの電子のエネルギー損失の合計を図 61 で示し、また、その時の μ 粒子が崩壊して電子が生成された直後の電子の運動エネルギーを図 62 で示す。1 ビームでのシンチレーター内での様子を図 63 で示す。左から運動エネルギー 50MeV の μ 粒子 (赤線) が入射し、ニュートリノ 2 本 (緑線) と電子 (赤線) に崩壊した。そして電子から γ 線 (緑線) が放出されている。

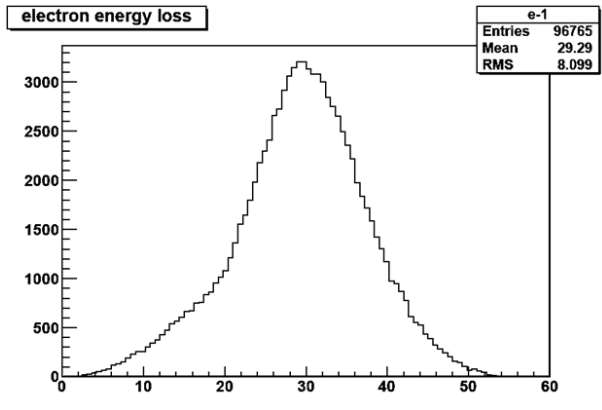


図 61 電子のエネルギー損失の合計

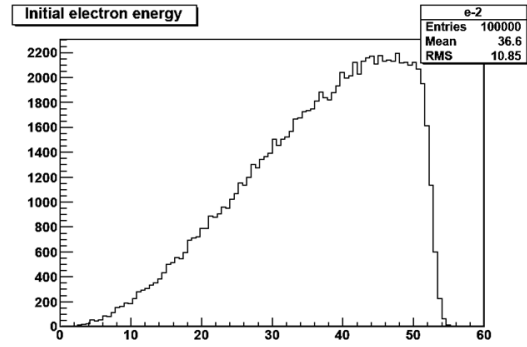


図 62 μ 粒子崩壊直後の電子の運動エネルギー

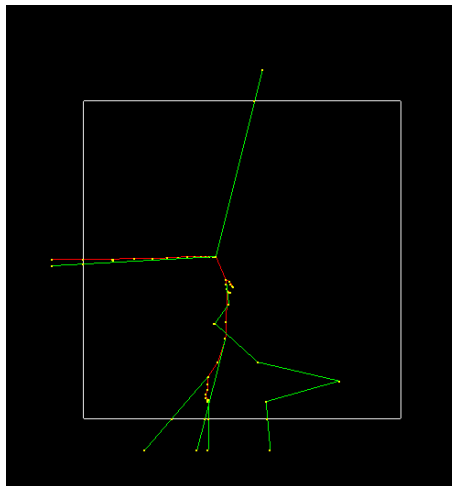


図 63 μ 粒子を入射させたときの崩壊の様子

図 61 と図 62 は明らかに形が異なっていることがわかる。これは、電子はシンチレーターの中で止まるので、電子がシンチレーターを通過中にエネルギーを失っていると考えられる。実際、Geant4 シミュレーションでは γ 線が放出されるので、この γ が持ち去ったと考えるのが妥当である。したがって、制動放射の効果も影響していると考えられる。

4. シミュレーション結果の考察

図 62 と図 51 を比較すると一致していると思われる。また、図 61 より、Geant4 シミュレーションでの電子のエネルギースペクトラムのピークは約 30MeV とわかる。図 52 より、今回の実験で測定した電子のエネルギースペクトラムのピークは約 30MeV なので、シミュレーションの値と一致することがわかった。しかし、ヒストグラムの両端（低エネルギー部、高エネルギー部）の振る舞いは一致していない。そのため一致しない理由を考える。高エネルギー部の振る舞いに効果をもたらしていると考えられるのはミュオニック原子の効果である。ミュオニック原子とは負の電荷をもった μ^- が原子核に束縛された状態である。そのため、自由粒子の μ^- とは異なる振る舞いをする。その効果を考慮して導いたグラフが図 64 である。

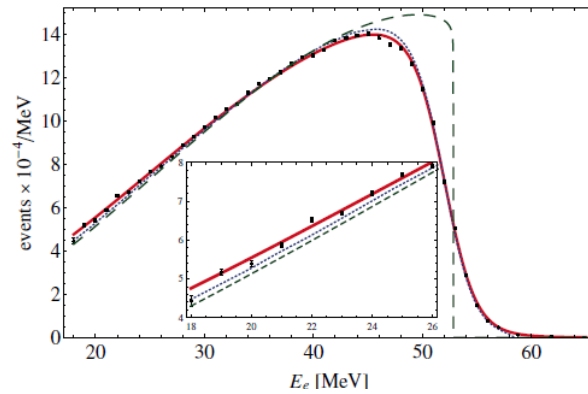


図 64 電子エネルギースペクトラム（ミュオニックアトム効果を考慮） 注 1

図 64 の高エネルギー部にテールがみられる。このため、ミュオニック原子の効果により、測定した電子のエネルギースペクトラムの高エネルギー部にテールがみられると考えられる。また、光電子増倍管に直接電子が入射した場合の効果についても考える必要がある。低エネルギー部についてはまだ考察が十分ではなく、今後の課題である。そのためには、Discriminator のしきい値によりカットされているパルスについても考えなければならないと思われる。

8 まとめ

8.1 実験のまとめ

1. μ 粒子の寿命測定

μ 粒子の寿命 τ は $2.230 \pm 0.007(\mu\text{s})$ となり、文献値と 1.5 % の差となった

2. 電子エネルギースペクトラムの測定

理論値と実験値では、分布の形もピークも異なる。原因は制動放射が考えられる。そこで Geant4 でシミュレーションすると、シミュレーション値と実験値のピークは一致したが分布の形は異なった。高エネルギー部のテールについてはミュオニックアトムの影響が考えられるが、ほかにも要因があると思われる。低エネルギー部の急な立ち上がりについては今後の課題である。

8.2 謝辞

私たちの卒業研究のためにお忙しい中、熱心にご指導して下さった林井先生をはじめ、研究するにあたって相談にのっていただいた宮林先生、アドバイスを下さった諸先輩方に心より感謝しております。この一年で学んだことを今後生かしていけるよう努めていきたいと思っております。本当にありがとうございました。V8、動いてくれてありがとう。

8.3 参考文献

- [1] 奈良女子大学理学部物理科学科 2014 年度卒業生 新井智穂 長谷川香織
「2013 年度卒業論文 μ 粒子の寿命と電子のエネルギースペクトラムの測定」
- [2] 奈良女子大学理学部物理科学科 2013 年度大学院後期課程 修了 岩下友子
「Geant4 の使い方入門」
- [3] Andrzej Czarnecki, Matthew Dowling, Xavier Garcia i Tormo, William J. Marciano, and Robert Szafron
「Michel decay spectrum for a muon bound to a nucleus」 (arXiv:1406.3575v2[hep-ph] 14 Oct 2014)
- [4] 平成 24 年・第 85 冊・理科年表
国立天文台
- [5] William R. Leo
「Technique for Nuclear and Particle Physics Experiments」
- [6] http://atlas.kek.jp/Geant4Poster_A4.pdf
「Geant4 (ジアント 4) - KEK」
- [7] 小川 岩雄「放射線 (改訂版)」
- [8] W.-M. Yao, "Review of Particle Physics",
J.Phys,G33,120(2006).
- [9] W.-M. Yao, "Review of Particle Physics",
J.Phys,G33,104(2006).

付録 A ele.c (CAMAC からのデータ収集)

```
/***** ele.c **** created 2012/Feb./1th*****  
 * Original version was written by S.Ono 2002/Jan./27th  
 * This is simplified version having only CAMAC control/IO part.  
 * LAM clear was moved to outside of "if(q!=0)". 2003/Jul./9th  
 * 2013/11/27 :update for stopped muon spectrum reading.  
 *****/  
/**** original title comment ****/  
 *****/  
#include <fcntl.h>  
#include <sys/time.h>  
#include <unistd.h>  
#include <sys/types.h>  
#include <stdio.h>  
#include <string.h>  
#include <stdlib.h>  
#include <math.h>  
#include <sys/errno.h>  
#include "camlib.h"  
#include <time.h>  
#include <netinet/in.h>  
  
FILE *fp; /* The file to save the taken data.*/  
  
int main(){  
    int i_ev, n_ev;  
    int q,x,data21,data22,data23,data24,data31,data32,data33;  
    int data1,data2,data3,data4,data5,data6,data7,data8,data9,data10,data11,data12;  
    int cadcq1,cadcq2,ctdcqn,cfadcn;  
    int ch1,ch2,ch3,ch4,ch5,ch6,ch7,ch8,ch9,ch10,ch11,ch12;  
    int Nrec;  
    int lamsrc;  
    int lamch;  
  
    char fname[36];  
  
        int qq;  
        int head;  
  
    //    const int Nmax=500;  
  
    cadcq1=1; /* muon ADC-Qmode module number */  
    cadcq2=2; /* electron ADC-Qmode module number */  
    ctdcqn=3; /* TDC module number */  
    cfadcn=5; /* FADC module address */  
    ch1=0;  
    ch2=1;  
    ch3=2;  
    ch4=3;  
    ch5=4;  
    ch6=5;  
    ch7=6;  
    ch8=7;  
    ch9=8;  
    ch10=9;  
    ch11=10;  
    ch12=11;  
  
    /*=====  
    * Ask the file name to save the taken data.  
    * Also open the data file.
```



```

*****/

printf("File name to save data?\n");
scanf("%s",fname);

/*      fp=fopen("../data2012/data0301-4.dat", "w"); */
fp=fopen(fname, "w");

/*****
 * How many events do you take?
*****/
printf("Number of events?\n");
scanf("%d",&n_ev);
fprintf(fp,"%d\n",n_ev);
/*      n_ev=500000; */

/*****
 * Open CCP interface device file.
 * If it fails, exit.
*****/
if(COPEN()){
printf("ccp open error\n");
exit(-1);
}

/*****
 * Initialize CAMAC.
*****/
CSETCR(0);
CGENZ();
CGENC();
CREMI();

lamsrc=cadcqn1;
//lamsrc=cadcqn2;
lamch=ch1;

printf("LAM source : %d,%d\n" ,lamsrc,lamch);

/*****
 * Enable LAM and Clear it.
*****/
CAMAC(NAF(lamsrc,lamch,26),&data1,&q,&x); /* F=26 is enable. */
CAMAC(NAF(lamsrc,lamch, 9),&data1,&q,&x); /* F=9 is clear. */

CAMAC(NAF(cadcqn2,ch1,26),&data1,&q,&x); /* F=26 is enable. */
CAMAC(NAF(ctdcqn,ch1,26),&data1,&q,&x); /* F=26 is enable. */
CAMAC(NAF(cadcqn2,ch1,9),&data1,&q,&x); /* clear LAM. */
CAMAC(NAF(ctdcqn,ch1,9),&data1,&q,&x); /* clear LAM */
/*****
 * Again send enable command to prepare the first event.
*****/
CAMAC(NAF(lamsrc,lamch,26),&data1,&q,&x);
CAMAC(NAF(cadcqn2,ch1,26),&data1,&q,&x); /* F=26 is enable. */
CAMAC(NAF(ctdcqn,ch1,26),&data1,&q,&x); /* F=26 is enable. */

/*****
 * send message to the user.
*****/
printf("CAMAC initilize done.\n");
printf("Number of event = %d\n",n_ev);

```

```

/*====
 * Event loop.
 *====*/
i_ev = 1;
while( i_ev <= n_ev )
{

CAMAC(NAF(cadcqn1,ch1,9),&data1,&q,&x); /* F=9 LAM clear. */
CAMAC(NAF(cadcqn2,ch1,9),&data1,&q,&x); /* F=9 LAM clear. */
CAMAC(NAF(ctdcqn,ch1,9),&data1,&q,&x); /* F=9 LAM clear. */

/*-----
 * Test LAM.
 *-----*/

do {
CAMAC(NAF(lamsrc,lamch,8),&data1,&q,&x); /* F=8 is test LAM.*/
} while ( q==0);

/*-----
 * If no event comes yet, q is set to be 0,
 * otherwise, the digitized event is there!
 *-----*/

/*-----
 * Read the digitized data from the register.
 *-----*/
CAMAC(NAF(cadcqn1,ch1,2),&data21,&qq,&x);
CAMAC(NAF(cadcqn1,ch2,2),&data22,&qq,&x);
CAMAC(NAF(cadcqn1,ch3,2),&data23,&qq,&x);
CAMAC(NAF(cadcqn1,ch4,2),&data24,&qq,&x);
CAMAC(NAF(cadcqn2,ch1,2),&data1,&qq,&x);
CAMAC(NAF(cadcqn2,ch2,2),&data2,&qq,&x);
CAMAC(NAF(cadcqn2,ch3,2),&data3,&qq,&x);
CAMAC(NAF(cadcqn2,ch4,2),&data4,&qq,&x);
CAMAC(NAF(cadcqn2,ch5,2),&data5,&qq,&x);
CAMAC(NAF(cadcqn2,ch6,2),&data6,&qq,&x);
CAMAC(NAF(cadcqn2,ch7,2),&data7,&qq,&x);
CAMAC(NAF(cadcqn2,ch8,2),&data8,&qq,&x);
CAMAC(NAF(cadcqn2,ch9,2),&data9,&qq,&x);
CAMAC(NAF(cadcqn2,ch10,2),&data10,&qq,&x);
CAMAC(NAF(cadcqn2,ch11,2),&data11,&qq,&x);
//          CAMAC(NAF(cadcqn2,ch12,0),&data12,&qq,&x);
//
//2014/11/18:
// Use new adc at slot 5, channel 0, since we can not read the adc
// at channel 11.
CAMAC(NAF(4,0,2),&data12,&qq,&x);
CAMAC(NAF(ctdcqn,ch1,2),&data31,&qq,&x);
CAMAC(NAF(ctdcqn,ch2,2),&data32,&qq,&x);
CAMAC(NAF(ctdcqn,ch3,2),&data33,&qq,&x);

/*-----
 * Update the event counter(i_ev), and send message for
 * every 50 events or there are data for big electron signal.
 *-----*/

if( i_ev%500 == 1 || data1>50 )
//      if(i_ev%2==1)
//      if( i_ev%500 == 1 || data1>50 )
{
printf("Event= %d\n",i_ev);

```

```

printf("data21= %d\n",data21);
printf("data22= %d\n",data22);
printf("data23= %d\n",data23);
printf("data24= %d\n",data24);
printf("data1= %d\n",data1);
printf("data2= %d\n",data2);
printf("data3= %d\n",data3);
printf("data4= %d\n",data4);
printf("data5= %d\n",data5);
printf("data6= %d\n",data6);
printf("data7= %d\n",data7);
printf("data8= %d\n",data8);
printf("data9= %d\n",data9);
printf("data10= %d\n",data10);
printf("data11= %d\n",data11);
printf("data12= %d\n",data12);
printf("data31= %d\n",data31);
printf("data32= %d\n",data32);
printf("data33= %d\n",data33);

}

head=-1;
Nrec=0;

/*-----
 * The read data is written into the file.
 *-----*/

if( data1<10 )
{

fprintf(fp," %d",head);
fprintf(fp," %d",i_ev);
fprintf(fp," %d",data21);
fprintf(fp," %d",data22);
fprintf(fp," %d",data23);
fprintf(fp," %d",data24);
fprintf(fp," %d",data1);
fprintf(fp," %d",data2);
fprintf(fp," %d",data3);
fprintf(fp," %d",data4);
fprintf(fp," %d",data5);
fprintf(fp," %d",data6);
fprintf(fp," %d",data7);
fprintf(fp," %d",data8);
fprintf(fp," %d",data9);
fprintf(fp," %d",data10);
fprintf(fp," %d",data11);
fprintf(fp," %d",data12);
fprintf(fp," %d",data31);
fprintf(fp," %d",data32);
fprintf(fp," %d\n",data33);

}

/*-----
 * Clear LAM to wait for the next event.
 * fuc=9 ; LAM clear for usual module
 * FADC
 * fuc=9 ; address clear

```

```

    *           =10 ; LAM clear
    *-----*/
    i_ev++;

    CAMAC(NAF(4,0,9),&data1,&q,&x);
    CAMAC(NAF(lamsrc,lamch,9),&data1,&q,&x);
    CAMAC(NAF(lamsrc,lamch,10),&data1,&q,&x);

}/*   end of event loop   */

/*=====
Instructions for termination.
*=====*/
CAMAC(NAF(lamsrc,lamch,24),&data1,&q,&x); /* F=24 is desable lam.*/
CAMAC(NAF(cadcqn1,ch1,24),&data1,&q,&x); /* F=26 is enable. */
CAMAC(NAF(cadcqn2,ch1,24),&data1,&q,&x); /* F=26 is enable. */
CAMAC(NAF(ctdcqn,ch1,24),&data1,&q,&x); /* F=26 is enable. */

CCLOSE();                /* CAMAC close. */
fclose(fp);              /* Close data file.*/

return 0;

}

```

付録 B mu-e.kumac (paw によるヒストグラム作成)

```
macro mu-e file=data1211-2.dat max=500000. min=0. fit=yes
mess file name =[file] max=[max] min=[min] fit=[fit]
**fortran/file 66 [file].ps
**meta 66 -111
**hi/del *
opt utit
*--
* data file
*---
*fortran/file 1 ./data/[file].txt
fortran/file 1 /home/ug2013/data2014/[file]
*fortran/file 1 [file]
call readmue.f(10)
close 1

opt zfl1
opt soft
set *
igset txfp -130
*set xsiz 18
*set ysiz 10
set asiz 0.4
zone 1 1
opt nbox

nt/list

set dmod 1
opt stat
*hi/pl 10
*hi/pl 10(0:40)
*set btyp 3
*set htyp 244
*opt fit
*if ([fit].EQ.'yes') then
*hi/fit 10 g
*endif

*title 'test '
*atitle 'ADC count' 'Number of events / bin'
*pic/print ./eps/[file].eps

**close 66

*mess ./eps/[file].eps is created
```

付録 C readmue.f (ヒストグラムの選別条件)

```

subroutine readmue(id)
  PARAMETER (NWPAWC=2000000)
  COMMON /PAWC/ H(NWPAWC)
  CHARACTER*8 CHTAGS(21)
  CHARACTER*8 HEADER(21)
C   DATA CHTAGS /'IEV ','T1Q ','CSIQ ','T2Q ','
C   +   'T3Q ','CSIV ','TDA ','TDT2'/

  DATA CHTAGS/'IEV ','S1m ','S2m ','T1m ','T2m ','
+   'S1e ','S2e ','T1e ','T2e ','
+   'V1 ','V2 ','
+   'V3 ','V4 ','V5 ','V6 ','V7 ','
+   'V8 ','TS1 ','TS2 ','TS12 ','S12e'/

  Integer nvhit
  DIMENSION idat(19)
  DIMENSION data(20)

C   CALL hlimit(NWPAWC)
  call hbookn(id,'muon data ',21, '//MUON',1000,CHTAGS)

C   read(1,*) ( HEADER(I),I=1,1)
  read(1,*) ( HEADER(I),I=1,9)
C   print *, ' head(1-4)=',HEADER(1),HEADER(2),HEADER(3),HEADER(4)
C   print *, ' head(5-7)=',HEADER(5),HEADER(6),HEADER(7)
C   print *, ' head(8-9)=',HEADER(8),HEADER(9)

  num=200000000
  ii=0
  ne=0
  read(1,*, end=100 ) itot
  print *, ' itot=',itot

  do 10 i=1, num
C--
C   data text line should be les than 140....
C--
  ii=ii+1
  read(1,*, end=100 ) iflag, iev, (idat(k),k=1,19)

C
C   idump= i- (i/1000)*1000
C   if(idump.eq.1) then
C     print *, ' iev=',i, ' S1_adc(mu)=' ,idat(1), ' S1_adc(e)=' ,idat(5)
C   endif
  data(1)=iev
  data(2)=idat(1)-117
  data(3)=idat(2)-110
c   data(3)=idat(1)-117+idat(2)-110
  data(4)=idat(3)-101
  data(5)=idat(4)-96
  data(6)=idat(5)-56
  data(7)=idat(6)-62
  data(8)=idat(7)-55
  data(9)=idat(8)-58
  data(10)=idat(9)-66
  data(11)=idat(10)-62
  data(12)=idat(11)-62

```

```

data(13)=idat(12)-63
data(14)=idat(13)-64
data(15)=idat(14)-65
data(16)=idat(15)-71
data(17)=idat(16)-77
c   if (idat(17).ge.2001.and.idat(17).le.4347) then
c       idat(17)=0
c   endif
c   if (idat(17).ge.4096) then
c       data(18)=idat(17)-2346
c   else
c       data(18)=idat(17)
c   endif
c   if (idat(18).ge.4096) then
c       data(19)=idat(18)-2346
c   else
c       data(19)=idat(18)
c   endif
c   if (idat(19).ge.2001.and.idat(19).le.4347) then
c       idat(19)=0
c   endif
c   if(idat(19).ge.4096) then
c       data(20)=idat(19)-2346
c   else
c       data(20)=idat(19)
c   endif
data(21)=idat(5)+idat(6)-187

C--
C Selection
C 1)skip if there is a hit at T2.
C 2)skip there is no hit of S1 at the electron timing.
C--

c   if(data(4)>1875) goto 10
c   if(data(6)<10) goto 10
c   if(data(10)<50) goto 10
c   if(data(11)<50) goto 10
c   if(data(12)<50) goto 10
c   if(data(13)<50) goto 10
c   if(data(14)<50) goto 10
c   if(data(15)<50) goto 10
c   if(data(16)<50) goto 10
c   if(data(8)<100) goto 10
c   if(data(9)<100) goto 10
c   nvhit=0
c   do 1 j=8, 9
c       if(data(j) < 100) then
c           nvhit=nvhit+1
c       endif
c1   continue
c
c   do 2 k=10, 17
c       if(data(k) < 50) then
c           nvhit=nvhit+1
c       endif
c2   continue

c   if(nvhit < 10) goto 10
ne=ne+1
idump= ne- (ne/50)*50
if(idump.eq.1) then
print *,'electron signal: iev=',i, ' S1_adc(mu)=' ,idat(1),
+ ' S1_adc(e)=' ,idat(5), ' nvhit=' ,nvhit ,' ne=' ,ne

```

```
endif

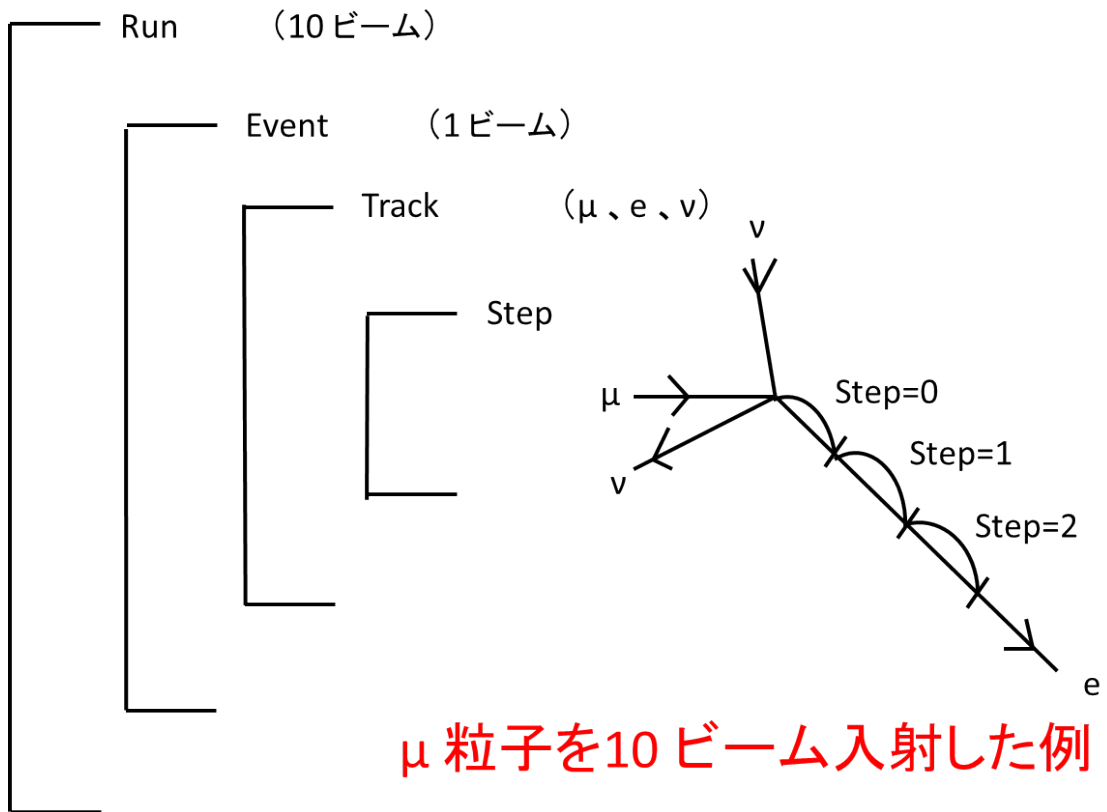
C--
C Selection
C skip if there is a hit of veto counter.
C--
C if(nvhit.ge.1) go to 10
C
C-- select only events which have at least one hit for veto counter
C
C if(nvhit.eq.0) go to 10

call hfn(id,data)

c
c fadc data read
c

10 continue
100 continue
print *, ' ne=',ne
close(1)
end
```


付録 D Geant4 の構造の概要



Geant4の主な物理量のコマンド

SteppingAction.cc

aTrack->GetTrackID()	Track ID
aTrack->GetParentID()	Parent ID
aTrack->GetDefinition()->GetParticleName()	Particle
aStep->GetTotalEnergyDeposit()	dEStep
aTrack->GetKineticEnergy()	KineE
aStep->GetStepLength()	StepLeng
aTrack->GetCurrentStepNumber()	Step

その他の物理量はSteppingVerbose.ccに書かれているので必要に応じて持ってくる。

注：fStep→aStep、fTrack→aTrackにする。

電子のエネルギー損失をヒストグラムにしたい場合

- EventAction.ccでファイルを定義し、ファイルを開く。
- SteppingAction.ccでParticle = eの場合、Eladdにdeを代入する。
- EventAction.hhでEdepElに1 Stepごとのdeを足していく。
- イベントごとの粒子のエネルギー損失の値 (= EdepEl) をファイルに書き込む。
- EventAction.ccでファイルを閉じる。

EventAction.cc

最初

```
#include <fstream>
std::ofstream ofs;
```

始まり

```
ofs.open("edep.dat", std::ios::out);
```

edep.datというファイルを作り、
ファイルを開く。

終わり

```
ofs.close();
```

ファイルを閉じる。

ループの中

```
EdepEl=0.;
```

初期値を0にする。

```
ofs << EdepEl << " " <<<G4endl; EdepElの値をedep.dat
```

ファイルに書き出す。

SteppingAction.cc

```
G4double edep = aStep->GetTotalEnergyDeposit(); edepの定義
```

```
if (aTrack->GetDefinition()->GetParticleName()=="e-") Particle =e-を  
条件にする。
```

```
eventaction->Eladd(edep); Eladdにedepを代入
```

注：粒子の定義は PhysicsList.ccにあるので、
粒子の名前はPhysicsList.ccを参照する。

EventAction.hh

```
public:  
void Eladd(G4double de) {EdepEl +=de}; EdepElにdeを足していく。
```

```
private:  
G4double EdepEl;
```

付録 E exampleN03.cc

```
//
// *****
// * License and Disclaimer *
// *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//
//
// $Id: exampleN03.cc,v 1.36 2008/11/24 13:54:38 lgarnier Exp $
// GEANT4 tag $Name: geant4-09-02 $
//
//
//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

#include "G4RunManager.hh"
#include "G4UImanager.hh"

#include "Randomize.hh"

#include "ExN03DetectorConstruction.hh"
#include "ExN03PhysicsList.hh"
#include "ExN03PrimaryGeneratorAction.hh"
#include "ExN03RunAction.hh"
#include "ExN03EventAction.hh"
#include "ExN03SteppingAction.hh"
#include "ExN03SteppingVerbose.hh"

#ifdef G4VIS_USE
#include "G4VisExecutive.hh"
#endif

#if defined(G4UI_USE_TCSH)
#include "G4UITerminal.hh"
#include "G4UITcsh.hh"
#elif defined(G4UI_USE_XM)
#include "G4UIXm.hh"
#elif defined(G4UI_USE_WIN32)
#include "G4UIWin32.hh"
#elif defined(G4UI_USE_QT)
#include "G4UIQt.hh"
#include "G4Qt.hh"
#endif
```

```

#else
#include "G4UITerminal.hh"
#endif

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

int main(int argc,char** argv)
{
  // Choose the Random engine
  //
  CLHEP::HepRandom::setTheEngine(new CLHEP::RanecuEngine);

  // User Verbose output class
  //
  G4VSteppingVerbose::SetInstance(new ExN03SteppingVerbose);

  // Construct the default run manager
  //
  G4RunManager * runManager = new G4RunManager;

  // Set mandatory initialization classes
  //
  ExN03DetectorConstruction* detector = new ExN03DetectorConstruction;
  runManager->SetUserInitialization(detector);
  //
  G4VUserPhysicsList* physics = new ExN03PhysicsList;
  runManager->SetUserInitialization(physics);

  // Set user action classes
  //
  G4VUserPrimaryGeneratorAction* gen_action =
      new ExN03PrimaryGeneratorAction(detector);
  runManager->SetUserAction(gen_action);
  //
  ExN03RunAction* run_action = new ExN03RunAction;
  runManager->SetUserAction(run_action);
  //
  ExN03EventAction* event_action = new ExN03EventAction(run_action);
  runManager->SetUserAction(event_action);
  //
  G4UserSteppingAction* stepping_action =
      new ExN03SteppingAction(detector, event_action);
  runManager->SetUserAction(stepping_action);

  // Initialize G4 kernel
  //
  runManager->Initialize();

#ifdef G4VIS_USE
  // Initialize visualization
  //
  G4VisManager* visManager = new G4VisExecutive;
  visManager->Initialize();
#endif

  // Get the pointer to the User Interface manager
  //
  G4UImanager* UI = G4UImanager::GetUIpointer();

  if (argc!=1) // batch mode
  {
    G4String command = "/control/execute ";
    G4String fileName = argv[1];
    UI->ApplyCommand(command+fileName);
  }
}

```

```

    }
else          // interactive mode : define visualization UI terminal
{
    G4UIsession* session = 0;
#if defined(G4UI_USE_TCSH)
    session = new G4UITerminal(new G4UITcsh);
#elif defined(G4UI_USE_XM)
    session = new G4UIXm(argc,argv);
    UI->ApplyCommand("/control/execute visTutor/gui.mac");
#elif defined(G4UI_USE_WIN32)
    session = new G4UIWin32();
    UI->ApplyCommand("/control/execute visTutor/gui.mac");
#elif defined(G4UI_USE_QT)
    session = new G4UIQt(argc,argv);
    UI->ApplyCommand("/control/execute visTutor/gui.mac");
#else
    session = new G4UITerminal();
#endif
#ifdef G4VIS_USE
    UI->ApplyCommand("/control/execute vis.mac");
#endif
    session->SessionStart();
    delete session;
}

// Job termination
// Free the store: user actions, physics_list and detector_description are
//                owned and deleted by the run manager, so they should not
//                be deleted in the main() program !
#ifdef G4VIS_USE
    delete visManager;
#endif
    delete runManager;

    return 0;
}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

```

付録 F EventAction.hh

```
//
// *****
// * License and Disclaimer *
// *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//
//
// $Id: ExN03EventAction.hh,v 1.12 2007/07/02 13:22:08 vnivanch Exp $
// GEANT4 tag $Name: geant4-09-02 $
//
//
//.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
//.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

#ifndef ExN03EventAction_h
#define ExN03EventAction_h 1

#include "G4UserEventAction.hh"
#include "globals.hh"

class ExN03RunAction;
class ExN03EventActionMessenger;

//.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

class ExN03EventAction : public G4UserEventAction
{
public:
    ExN03EventAction(ExN03RunAction*);
    virtual ~ExN03EventAction();

    void BeginOfEventAction(const G4Event*);
    void EndOfEventAction(const G4Event*);

    void AddAbs(G4double de, G4double dl) {EnergyAbs += de; TrackLABs += dl;};
    void AddGap(G4double de, G4double dl) {EnergyGap += de; TrackLGap += dl;};

    void SetPrintModulo(G4int val) {printModulo = val;};
    void Eladd(G4double de){ EdepEl += de;};
    void Muadd(G4double de){ EdepMu += de;};
};
```

```

void Kineadd(G4double Kine1){KinetE += Kine1;};

private:
  G4double  EdepEl;
  G4double  EdepMu;
  G4double  KinetE;

  ExN03RunAction*  runAct;

  G4double  EnergyAbs , EnergyGap;
  G4double  TrackLAbs , TrackLGap;

  G4int     printModulo;

  ExN03EventActionMessenger*  eventMessenger;

//energy deposit of electron and muon.
};

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

#endif

```

付録 G EventAction.cc

```
//
// *****
// * License and Disclaimer *
// *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//
//
// $Id: ExN03EventAction.cc,v 1.29 2008/01/17 17:31:32 maire Exp $
// GEANT4 tag $Name: geant4-09-02 $
//
//
//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

#include "ExN03EventAction.hh"

#include "ExN03RunAction.hh"
#include "ExN03EventActionMessenger.hh"

#include "G4Event.hh"
#include "G4TrajectoryContainer.hh"
#include "G4VTrajectory.hh"
#include "G4VVisManager.hh"
#include "G4UnitsTable.hh"

#include "Randomize.hh"
#include <iomanip>
#include <fstream>
std::ofstream ofs;

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

ExN03EventAction::ExN03EventAction(ExN03RunAction* run)
:runAct(run),printModulo(1),eventMessenger(0)
{
    eventMessenger = new ExN03EventActionMessenger(this);
    ofs.open("edep2.dat", std::ios::out);
    if(! ofs.good()){
        G4cout<<" EventAction; error opening edep.dat. ofs.good="<<ofs.good()<<G4endl;
    }
}
```



```

}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

ExN03EventAction::~ExN03EventAction()
{
    ofs.close();
    G4cout<<" EventAction: edep.dat file is closed"<<G4endl;
    delete eventMessenger;
}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

void ExN03EventAction::BeginOfEventAction(const G4Event* evt)
{
    G4int evtNb = evt->GetEventID();
    if (evtNb%printModulo == 0) {
        G4cout << "\n---> Begin of event: " << evtNb << G4endl;
        CLHEP::HepRandom::showEngineStatus();
    }
    //--
    // initialize energy sum
    //--
    EdepEl=0.;
    EdepMu=0.;
    KinetE=0.;
}

// initialisation per event
EnergyAbs = EnergyGap = 0.;
TrackLABs = TrackLGap = 0.;
}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

void ExN03EventAction::EndOfEventAction(const G4Event* evt)
{
    //accumulates statistic
    //
    runAct->fillPerEvent(EnergyAbs, EnergyGap, TrackLABs, TrackLGap);

    //print per event (modulo n)
    //
    G4int evtNb = evt->GetEventID();
    if (evtNb%printModulo == 0) {
        G4cout << "---> End of event: " << evtNb << G4endl;

        G4cout
            << "   Absorber: total energy: " << std::setw(7)
            << G4BestUnit(EnergyAbs, "Energy")
            << "   total track length: " << std::setw(7)
            << G4BestUnit(TrackLABs, "Length")
            << G4endl
            << "   Gap: total energy: " << std::setw(7)
            << G4BestUnit(EnergyGap, "Energy")
            << "   total track length: " << std::setw(7)
            << G4BestUnit(TrackLGap, "Length")
            << G4endl;
    }
    //--
    // write energy sum to file.
    //--
    G4cout <<" Eventaction write EdelEl and EdepMu"<<G4endl;
    ofs << EdepEl<<"   "
        <<EdepMu<<"   "
        <<KinetE<<"   "<<G4endl;
}

```

```
G4cout <<" EdepEl, EdepMu, KineE="<< EdepEl<<", "<<EdepMu<<", "  
      <<KinetE<<G4endl;  
}  
}  
//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
```

付録 H SteppingAction.cc

```
//
// *****
// * License and Disclaimer *
// *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//
//
// $Id: ExN03SteppingAction.cc,v 1.15 2006/06/29 17:49:13 gunter Exp $
// GEANT4 tag $Name: geant4-09-02 $
//
//
//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
//#include
#include "ExN03SteppingAction.hh"

#include "ExN03DetectorConstruction.hh"
#include "ExN03EventAction.hh"

#include "G4Step.hh"
#include <fstream>
//extern
//std::ofstream ofs;

////#include "G4RunManager.hh"

//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

ExN03SteppingAction::ExN03SteppingAction(ExN03DetectorConstruction* det,
                                           ExN03EventAction* evt)
:detector(det), eventaction(evt)
{
// ofs.open("edep.dat", std::ios::out);
// if(! ofs.good()){
// G4cout<<" SteppingAction; error opening edep.dat. ofs.good="<<ofs.good()<<G4endl;
// }

}

//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
```

```

ExN03SteppingAction::~ExN03SteppingAction()
{
//   ofs.close();
//   G4cout<<" edep.dat file is created"<<G4endl;
}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

void ExN03SteppingAction::UserSteppingAction(const G4Step* aStep)
{
// get volume of the current step

//   G4cout<<" we are in UserSteppingAction"<<G4endl;

G4Track * aTrack=aStep->GetTrack();

G4VPhysicalVolume* volume
= aStep->GetPreStepPoint()->GetTouchableHandle()->GetVolume();

// collect energy and track length step by step
G4double edep = aStep->GetTotalEnergyDeposit();
G4double Kine = aTrack->GetKineticEnergy()+ edep;
G4double stepl = 0.;
if (aStep->GetTrack()->GetDefinition()->GetPDGCharge() != 0.)
    stepl = aStep->GetStepLength();

G4cout<<" SteppingAction TrkID = "<<std::setw(6)<< aTrack->GetTrackID()
    <<" Parents ID ="<<std::setw(6)<< aTrack->GetParentID()
    <<" name      ="<<std::setw(6)<< aTrack->GetDefinition()->GetParticleName();
G4cout<<" edep    ="<<std::setw(6)<< edep
    <<" stepl="<<std::setw(6)<< stepl
    <<" kine ="<<std::setw(6)<<Kine
    << G4endl;

//   ofs<<edep<<G4endl;

//   eventaction->EdepEl += edep;
//   eventaction->EdepMu += edep;
if (aTrack->GetDefinition()->GetParticleName()
    == "e-")
    eventaction->Eladd(edep);
if (aTrack->GetDefinition()->GetParticleName() == "mu-")
    eventaction->Muadd(edep);
if (aTrack->GetDefinition()->GetParticleName()
    == "e-" && aTrack->GetParentID() == 1
    && aTrack->GetCurrentStepNumber() == 1){
    eventaction->Kineadd(Kine);
    G4cout<<"Kineadd is called"<<G4endl;
}
if (volume == detector->GetAbsorber()) eventaction->AddAbs(edep,stepl);
if (volume == detector->GetGap())      eventaction->AddGap(edep,stepl);

//example of saving random number seed of this event, under condition
//// if (condition) G4RunManager::GetRunManager()->rndmSaveThisEvent();
}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

```