

## GM計数管の基本的な性質を学ぶ、パルス波形を見る実験

### ■GM管の出力を活用する

GM計数管は、主に $\beta$ 線や $\alpha$ 線などの電離放射線による気体の電離を、高電圧によるガス増幅を利用して検出する放射線検出器と言える。

ガス増幅という電子なだれ現象を利用した機構によって、増量した電子群は電位勾配に従ってアノードに、陽イオン群は逆にカソードに到達し、電荷群として検出される。この電荷群は電流となって外部回路に流れ込み、抵抗器の両端で電圧パルスとして観測される。このパルスを電気信号として取り出して計数するのがGM計数管の基本的な原理である。

これまでの実験では、この電気信号をマイコンで処理して表示装置に出力していた。そのため、計数値は読み取って記録する必要があり、計数値の統計的な処理やグラフ化には、改めて手作業でパソコンやタブレットに入力する必要があった。GM管の出力をパソコンに入力できれば、その手間が省ける。大気圧GM管の出力は正電圧パルスであり、パソコンには音声信号として入力できる。必要なプログラムがあればデータとして取り込み、その後はパソコンで処理することができる。

ここでは、プログラミング言語 Python を使用して各種の放射線実験に適用する。

### ■Python とは

無料で使用できるオープンソースの言語であり、音声や画像を扱うライブラリーも完備している。コマンドラインで入力するインタプリタ言語なので、記述したプログラムがそのまま動作する。言い換えれば、コードの記述がそのままソース・プログラムとなっている。

Python 自体とライブラリーをインストールする必要はあるが、ネット環境から簡単にインストールできる。使用するパソコンやタブレットも、Windows や Android、iOS、Linux など、多くのプラットフォームで使用できる。

ただし、ここでは Windows10 を使用する。Python 自体や必要なライブラリーのインストールの方法、Python の文法などについては添付資料を参考にして調べてほしい。プログラムについても、多くの実例が公開されていて、そのまま利用できるものも少なくない。

### ■パルス波形を見るプログラム

実行が確認できたプログラムを例として挙げる。#以下はコメント文で、各ラインを説明としている。なお、添付ファイルはテキスト・ファイルで、rename すれば Python ファイルとして実行できる。

まず、ユーザーのホーム・ディレクトリに当該ファイルをコピーしてから、Windows の画面左下にある「ここに入力して検索」をクリックして、「cmd」と入力し、コマンド・プロンプトを起動する。プロンプトにファイル名を入力すれば Python が起動する。以下では、プログラムの流れとポイントを解説する。

冒頭では、必要なライブラリーを呼び込む。なお、cv2 はプログラムの終了時のキー入力に使用する。

```
import pyaudio #マイク入出力処理
import numpy as np
import matplotlib.pyplot as plt #グラフ処理
import cv2 #カメラ画像処理
```

def は関数で、0.12 秒間サンプリングして、グラフの描画までの処理を行う。まず、入力ストリームを開き、バッファには CHUNK 単位で取り込む。frames は全データで、リストとして用意し、バッファからリストに追記を繰り返す。後処理の都合で、リストからアレイに変換してからストリームを閉じる。

```
def streamandshow(): #入力からグラフ描画まで処理
    CHUNK = 1024 #入出力の長さ
    FORMAT = pyaudio.paInt16 #2Byte 整数
    CHANNELS = 2 #ステレオ
    RATE = 44100 #サンプリング周波数(Hz)
    RECORD_SECONDS = 0.12 #録音時間(秒)

    p = pyaudio.PyAudio()
    stream = p.open(format = FORMAT,
                    channels = CHANNELS,
                    rate = RATE,
                    input = True, #入力のみ
                    frames_per_buffer = CHUNK)

    frames = [] #空の list を用意
    for i in range(int(RATE* RECORD_SECONDS
// CHUNK)): #CHUNK 単位で切り捨て
        data_raw = stream.read(CHUNK)
        frames.append(data_raw) #追記

    frames_int16 = np.array(frames)

    stream.stop_stream() #ストリームを閉じる
    stream.close()
    p.terminate()
```

バイトデータを整数に変換し、ステレオ録音の片チャンネルを抽出する。パルスのレベル調節用に、最大値を求めて画面に表示する。

グラフ表示では、0.12 秒間に相当する 44100 × 0.12 = 5292 のフレームを全波形として表示し、刻々と更新する。

```
data = np.frombuffer(frames_int16, dtype='int16')
data_l = data[:,2] #ステレオの L チャンネル
data_max = max(data_l) #最大値を求める
print(data_max) #最大値を画面に出力

fig = plt.figure(num = 1, figsize = (6, 4))
#グラフの大きさを決める

plt.plot(data_l[100:]) #最初の 100 点をスキップ
plt.pause(0.3) #0.3 秒間だけ描画
plt.clf() #グラフを閉じる
```

「q」をキーボードから入力すると、プログラムは終了する。

```
while True: #キー入力を待つ
    if cv2.waitKey(1) & 0xFF == ord('q'):
        #q を入力すると、
            break #処理を終了
        streamandshow() #処理を反復
```

## ■実験方法

7 セグメント LED 表示の計数装置から出力信号を取り出す。イヤホン・ジャックとパソコンのマイク入力端子をケーブルで接続するが、0.022 μF 程度のコンデンサを並列に入れて、パルス波高が過大 (<5V) とならないようにすること。必要があれば、マイクのプロパティを開き、レベルを調整する。

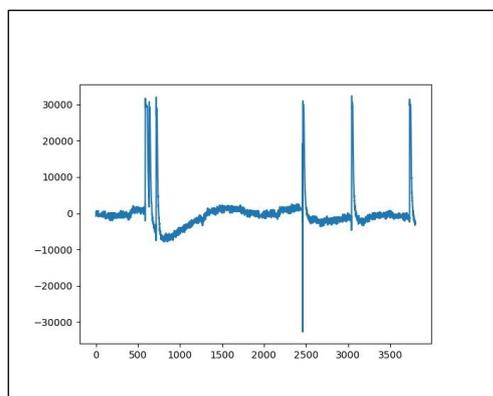


図 パルス波形の描画例