

Arduino を用いた分光器波長走査の自動化とそれによる 各種分光データの計測

齋藤研究室 1516115 小田辺光

1. 背景・目的

これまで自動化されていなかった分光器システムを、Arduino を用いて波長走査と計測を自動化することを目的とし研究を行った。

現在手動で使用している分光器（島津製 SPG-120S）を Arduino を活用して自動化することとした。この分光器の波長走査ユニット AT-120PL の仕様は表 1 の通りである。この波長走査を駆動・制御するコントローラが必要な為、その機能を果たす物を自作する。

AT-120PL 中の 2 相ステッピングモータを駆動するドライバーには、マイクロステップドライバー RD-021M8 を選定した。

表 1. AT-120PL の仕様

駆動モータ	2 相ステッピングモータ 定格 12 V, 0.4 A/相	
波長走査ステップ	0.2 nm/パルス(2 相励磁)	
励磁シーケンス 及び 波長走査方向	A(+) \rightarrow B(-) \rightarrow A(-) \rightarrow B(+) のとき、正方向 (正方向：波長が長くなる方向)	
パルス 速度	自起動 運転	最大 15 pps
	加減速 運転	最大 500 pps
外部出力信号	波長マーカ信号(波長 100 nm ごとに出力). 出力形式：オープンコレクタ(アクティブロー)	
	リミット信号(波長上限及び下限リミット信号). 出力形式：オープンコレクタ(アクティブハイ)	
制御信号コネクタ	JD-14S-C(A)(日本航空電子製)又は 57-40140(第一電子製)相当	
寸法・重量	幅 96×高さ 96×奥行 75 mm (突起物含まず). 約 0.85 kg	

2. 実験機器

波長走査に関係する主な機器は以下の通り。

- AT-120PL 島津小形分光器
- SPG-120 用波長駆動装置
- Arduino UNO
- 2-Phase Selectable Microstepping Motor Driver RD-021M8
- リレイ OMRON 2117P-US-DC5V
- HALOGEN LIGHT SOURCE HL-2000

3. 実験方法

本システムでは、分光器波長走査用回転軸の 1 回転は、波長 100 nm に対応し、必要なパルス数は 4000 である。したがって、1 パルス当たりの波長送りは 0.025 nm である。RD-021M8 の正逆決定の入力については、5V (High)印加時が、波長が増加する向きである。

図 1 に示した Arduino の試験プログラムにより、分光器波長走査の正常な動作を確認した。このプログラムでは、ch0 への入力の High/Low によりステッピングモータの回転する向きを決め、nm 個のパルスを送る動作を mm 回繰り返す（例では nm=400, mm=3 であるので、10 nm の波長送りを 3 回繰り返す）。

```
steppingmotor_2loops_input_nov27
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(0, INPUT);
  pinMode(1, INPUT);
}

// the loop function runs over and over again forever
void loop() {
  int x=2;
  int w=1000;
  int mm=3; //70;
  int nm=400;
  int pinp=12;
  int ipin0=0;
  int ipin1=1;
  int i=0;
  Serial.println(digitalRead(ipin0));
  while(1){
    digitalWrite(pinp,LOW); //回転の向き
    for(int m=1; m<=mm; m++){
      Serial.println(digitalRead(ipin1));
      for(int n=1; n<=nm; n++){
        digitalWrite(pinp,HIGH);
        i=i+1;
        delay(x/2);
        digitalWrite(pinp,LOW);
        delay(x/2);
      }
    }
  }
}
```

図 1 分光器波長走査のための Arduino Uno 用ステッピングモータ制御試験プログラム。

実際の測定・計測に向けさらに機能を付加し、図2に示した実用プログラムの制作を行った。

```

SM_loops_auto_manu_scan_jan17_2

int i;
float wl;
boolean dir;
int x=2;
//int pinstart=2; //波長走査開始ボタン入力 0:開始
//pin2はInterrupt0用
int pincancel=2; //interrupt0:途中キャンセル用ボタン入力 0:キャンセル
int pinstart=3; //←ハード変更。波長走査開始ボタン入力 0:開始
int pintrig=8; //トリガー用出力
int ud=7; //UD選択スライドスイッチ出力 (High:波長増加)
int ck=8; //マニュアル10nm走査用押しボタンスイッチ出力
int ckp=9; //マニュアル10nm走査用入力
volatile int pinrly=10; //relay用出力
int pinud=11; //UD選択用出力
int pinck=12; //リレス出力

void setup() {
  Serial.begin(9600);
  delay(200);
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(pinstart, INPUT_PULLUP);
  pinMode(pincancel, INPUT_PULLUP);
  pinMode(ud, INPUT_PULLUP);
  pinMode(ck, INPUT_PULLUP);
  pinMode(ckp, INPUT_PULLUP);
  pinMode(pintrig, OUTPUT);
  pinMode(pinrly, OUTPUT);
  pinMode(pinud, OUTPUT);
  pinMode(pinck, OUTPUT);
  i=0;
  digitalWrite(pintrig,LOW);
  attachInterrupt(digitalPinToInterrupt(2), poweroff,LOW);
}

```

```

// the loop function runs over and over again
void loop() {
  float wli=900.;
  float wlf=300.;
  float wls=10.;
  int nm=40*wls;
  int nr=40*(wli-wlf);
  int mm=(wli-wlf)/wls;
  // Serial.println(i);
  // Serial.println(wli+wlf);
  digitalWrite(pinud,LOW); //回転の向き
  while(digitalRead(pinstart)==0){
    for(int n=1; n<=nm; n++){
      for(int m=1; m<=nr; m++){
        digitalWrite(pinck,LOW);
        i=i-1;
        digitalWrite(pinck,HIGH);
        delay(x/2);
      }
      digitalWrite(pintrig,HIGH);
      delay(10*x);
      digitalWrite(pintrig,LOW);
      delay(1000*x);
    }
    wli=wli+.025*i;
    Serial.println(wli);
  }
  // 波長の戻し回転
  digitalWrite(pinud,HIGH); //回転の向き
  for(int n=1; n<=nr; n++){
    digitalWrite(pinck,LOW);
    i=i+1;
    digitalWrite(pinck,HIGH);
    delay(x/2);
  }
  wli=wli+.025*i;
  Serial.println(wli);
  while(digitalRead(ckp)==0){
    dir=digitalRead(ud); //回転の向き
    digitalWrite(pinud,dir); //回転の向き
    digitalWrite(pinck,LOW);
    if(dir==HIGH){i=i+1;}
    else{i=i-1;}
    wli=wli+.025*i;
    Serial.println(wli);
    digitalWrite(pinck,HIGH);
    delay(50*x);
  }
}

void poweroff(){
  digitalWrite(pinrly,HIGH);
}

```

図2 分光器波長走査のための Arduino Uno 用ステッピングモータ制御プログラム

主な機能と操作法の要点は以下の通り。

- デジタル 3 番ピン接続の押しボタンスイッチを押すと自動波長走査を開始 (例では 900 nm から 300 nm まで 10 nm 間隔で波長が減少する向きの走査を行い、終了後初期波長に復帰する)
- デジタル 2 番ピン接続の押しボタンスイッチを押すと緊急停止 (割り込み機能を利用)

- デジタル 8 番ピン接続の押しボタンスイッチを押すと、10 nm 単位の走査 (方向はデジタル 7 番ピン接続のスライド・スイッチで設定)
- デジタル 9 番ピン接続の押しボタンスイッチを押すと、押している間のみ走査 (方向はデジタル 7 番ピン接続のスライド・スイッチで設定)
- 分光器の波長範囲上限・下限リミット信号を受けると、リレー 2117P-US-DC5V を励磁し、駆動電源を強制的に遮断して破損防止

4. 実験結果・考察

このプログラムを使用してデータロガー (midi LOGGER GL900) で測定したフォトダイオード S2281(7K102)の光電流スペクトルを図3に示す。

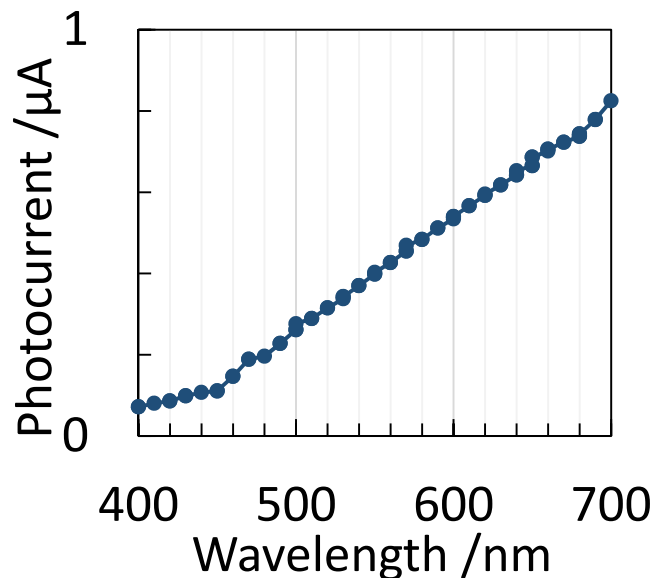


図3 本システムを使用して得られた Si フォトダイオード光電流スペクトル

5. まとめ

今回得られた要点は以下の通り。

- Arduino を活用して分光器の波長走査を駆動・制御するコントローラを自作し、プログラムの改良を重ね、要求される仕様を満たして自動化することに成功した。
- 自動化した分光器システムを用いて Si フォトダイオード光電流スペクトルを計測し、期待通りの結果を得ることができた。

6. 参考文献

- 高本孝頼「みんなの Arduino 入門」, リックテレコム, 2014