

光センサの基礎データを採取する。その目的は、ノイズ対策なので検討に役立てることにある。センサの応答速度は、十分に高速(legOS 解説によれば、AD 変換時間は 16.3  $\mu$ sec)である。この速度があれば、蛍光灯の明滅、ストロボの発光を十分に観測できると考えられるので、これらを観測する。

## 1．観測時のサンプリング速度

サンプリング速度がどの程度なのか、どこまで高速になるか、を確認する。

### 1-1．標準 API モード

標準 API を利用した場合、以下のコードで約 100  $\mu$ sec/cycle(10000 回の観測で約 1sec)であった。

```
p_time = get_system_up_time() ;
for ( i=0 ; i<10000 ; ++i) {          /* Total 1000msec=100 $\mu$ sec/cycle */
    t_light = LIGHT ( SENSOR_1 ) ;
    log[i] = (char)t_light ;
}
c_time = get_system_up_time() ;
```

### 1-2．生 API モード

legOS 解説にある生(直接)API を使用したところ、以下のコードで約 10  $\mu$ sec/cycle(10000 回の観測で約 100msec)であった。

```
p_time = get_system_up_time() ;
for ( i=0 ; i<10000 ; i=i+10 ) {
    t_light = (unsigned int)(SENSOR_1); log[i] = (char)(t_light>>8);
    t_light = (unsigned int)(SENSOR_1); log[i+1] = (char)(t_light>>8);
    t_light = (unsigned int)(SENSOR_1); log[i+2] = (char)(t_light>>8);
    t_light = (unsigned int)(SENSOR_1); log[i+3] = (char)(t_light>>8);
    t_light = (unsigned int)(SENSOR_1); log[i+4] = (char)(t_light>>8);
    t_light = (unsigned int)(SENSOR_1); log[i+5] = (char)(t_light>>8);
    t_light = (unsigned int)(SENSOR_1); log[i+6] = (char)(t_light>>8);
    t_light = (unsigned int)(SENSOR_1); log[i+7] = (char)(t_light>>8);
    t_light = (unsigned int)(SENSOR_1); log[i+8] = (char)(t_light>>8);
    t_light = (unsigned int)(SENSOR_1); log[i+9] = (char)(t_light>>8);
}
c_time = get_system_up_time() ;
```

また、以下のコードでは約 50  $\mu$ sec/cycle であった。

```
p_time = get_system_up_time() ;
for ( i=0 ; i<10000 ; ++i) {
    t_light = (unsigned int)(SENSOR_1) ;
    t_light = t_light>>8 ;
    log[i] = (char)t_light ;
    for ( j=0 ; j<2 ; ++j ) {
        tt = ++j ;
        tt3 = --j ;
    }
}
c_time = get_system_up_time() ;
```

## 2. 蛍光灯の観測

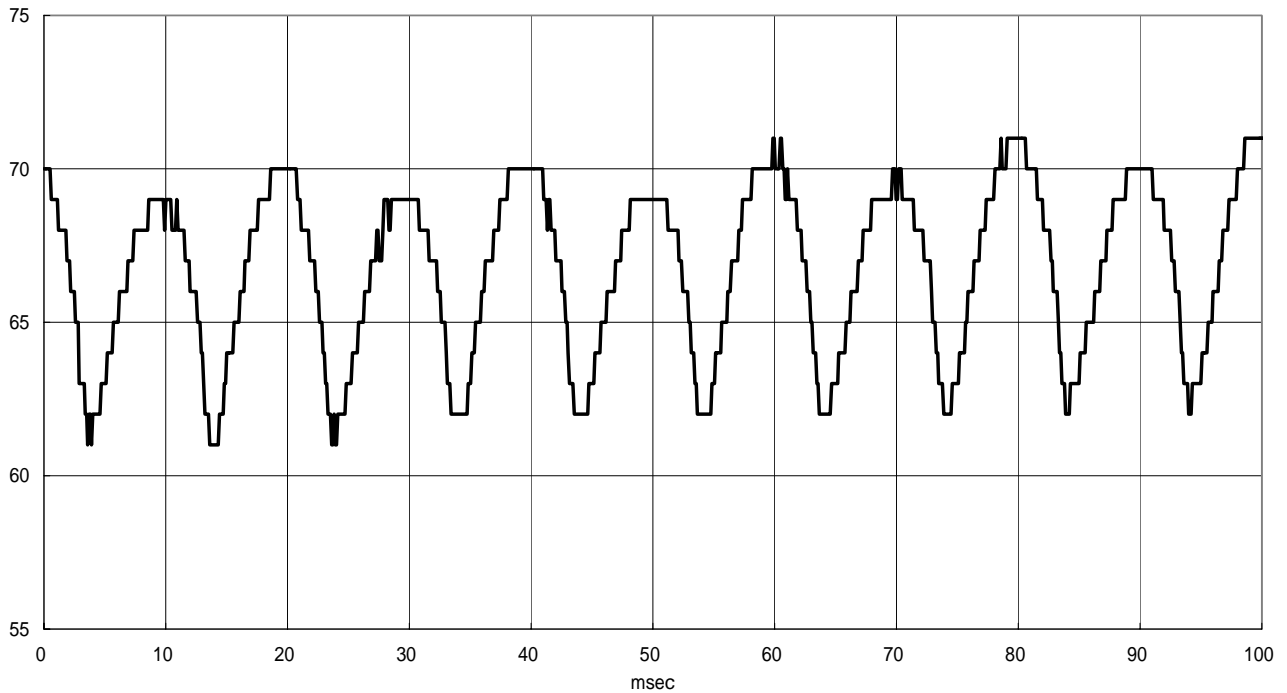
蛍光灯を観測した結果を以下に示す。

蛍光管 15Watt 直管

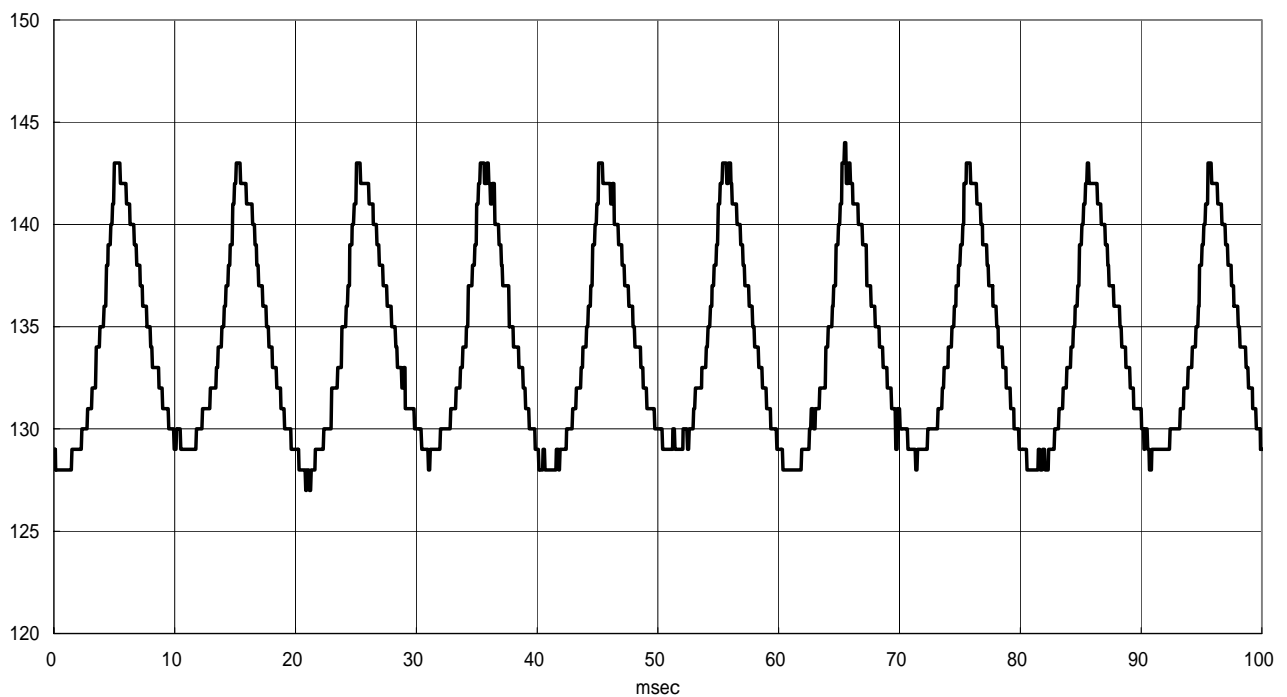
測定位置 管端より約 10cm、管球表面より約 5cm

測定地の商用電源は 50cycle/sec であり、100cycle/sec の明滅が計測された。

蛍光灯 標準API 100msec  
100  $\mu$  secサンプリング



蛍光灯 生API 100msec  
50  $\mu$  secサンプリング



### 3. ストロボの観測

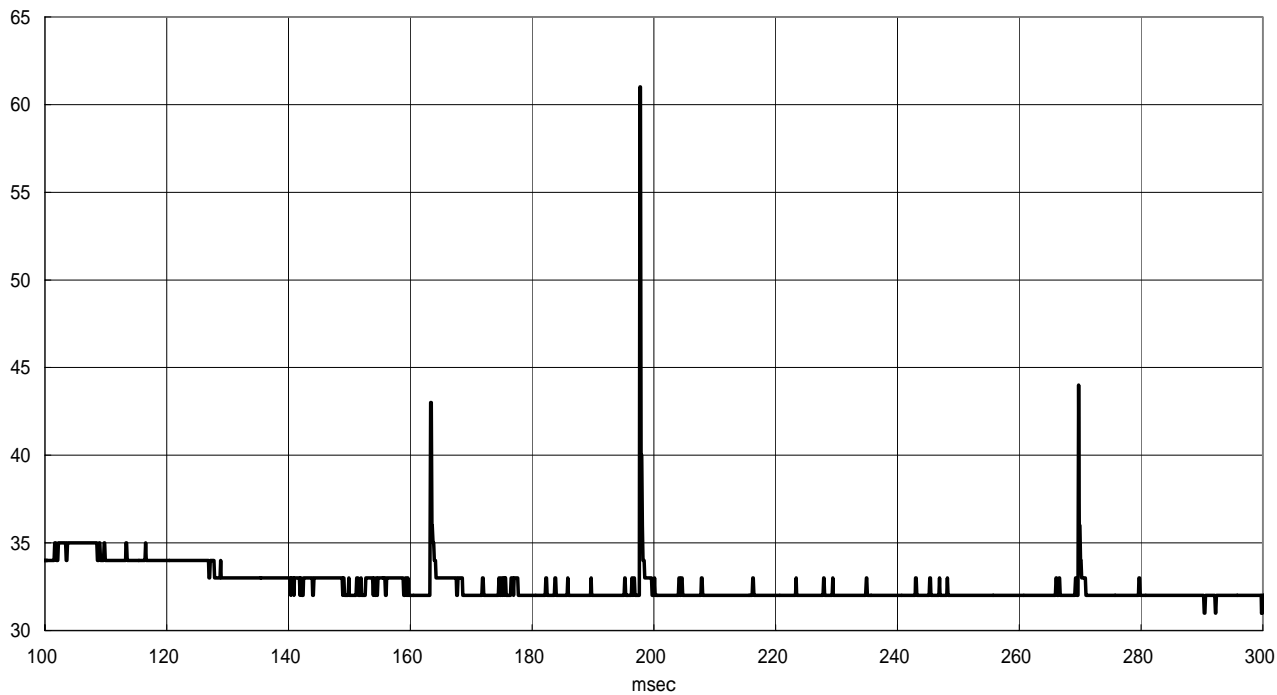
ストロボの観測結果を以下に示す。

ストロボ デジカメ Casio EX-Z40 内臓ストロボ  
測定位置 距離約 10cm でセンサに正対

標準 API を用いて観測したデータを示す。

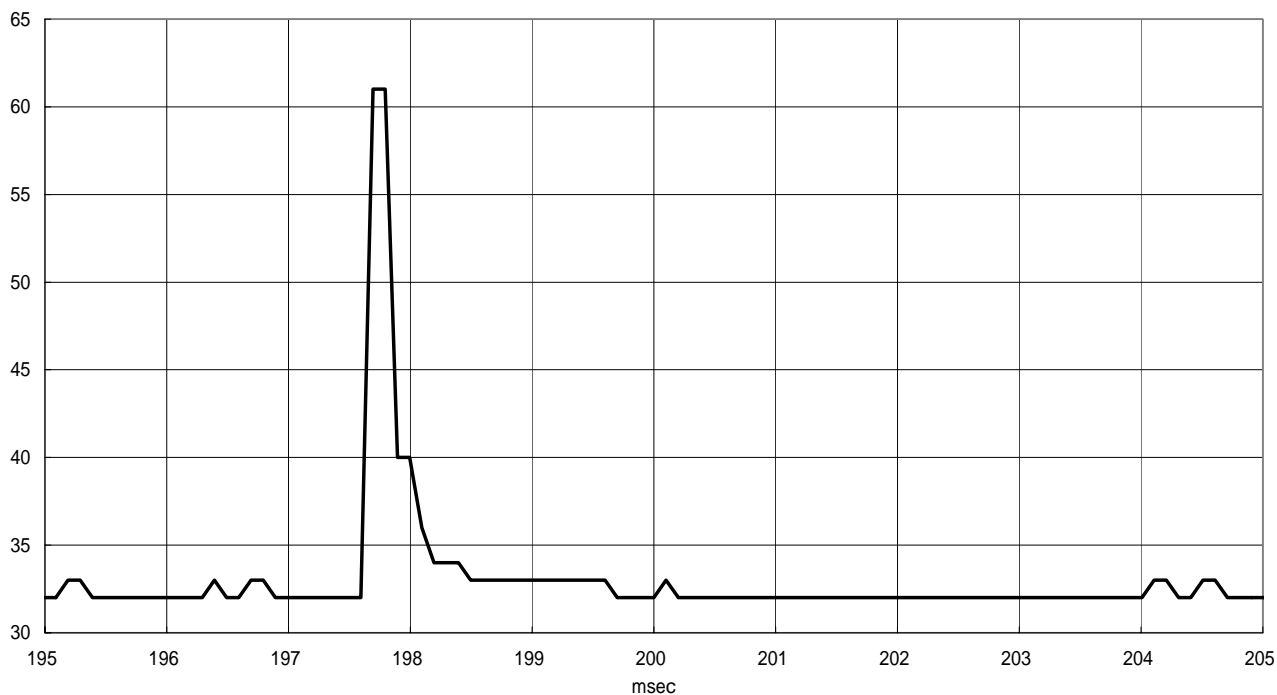
3回の分割発光を観測した。この発光は、いわゆる赤目防止のためか。

ストロボ 標準API 200msec  
100  $\mu$  secサンプリング



上図の中心部分を拡大したものを以下に示す。

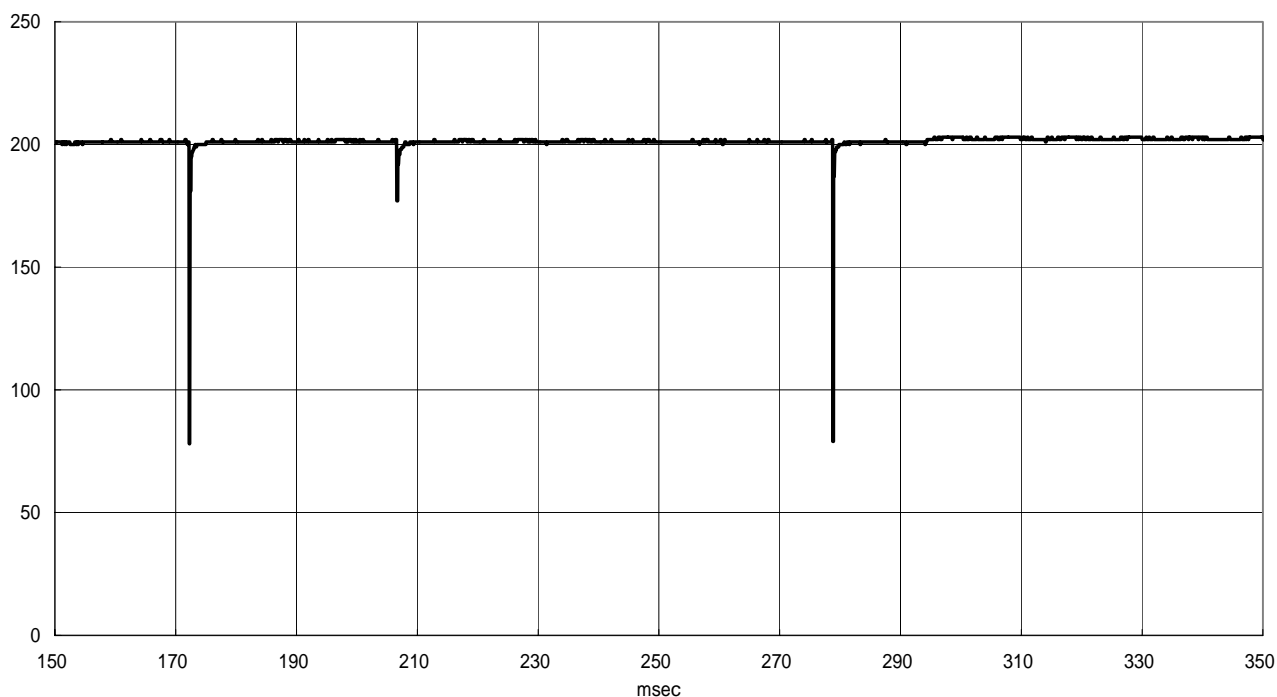
ストロボ 標準API 10msec  
100  $\mu$  secサンプリング



生 API で観測したデータを以下に示す。

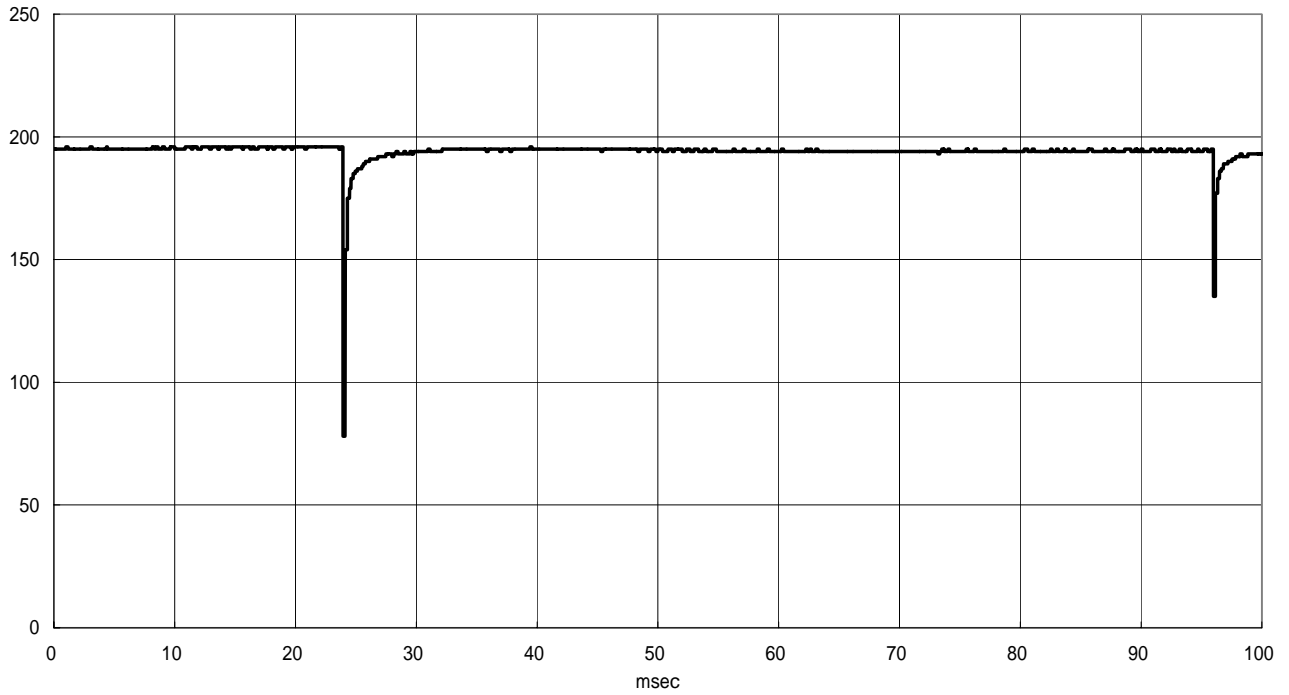
この観測でも 3 回の分割発光している。発光レベルのバランスが標準 API での観測と異なっている。異なる理由は不明。動作バラツキがあるように感じられる。

ストロボ 生API 200msec  
50  $\mu$  secサンプリング



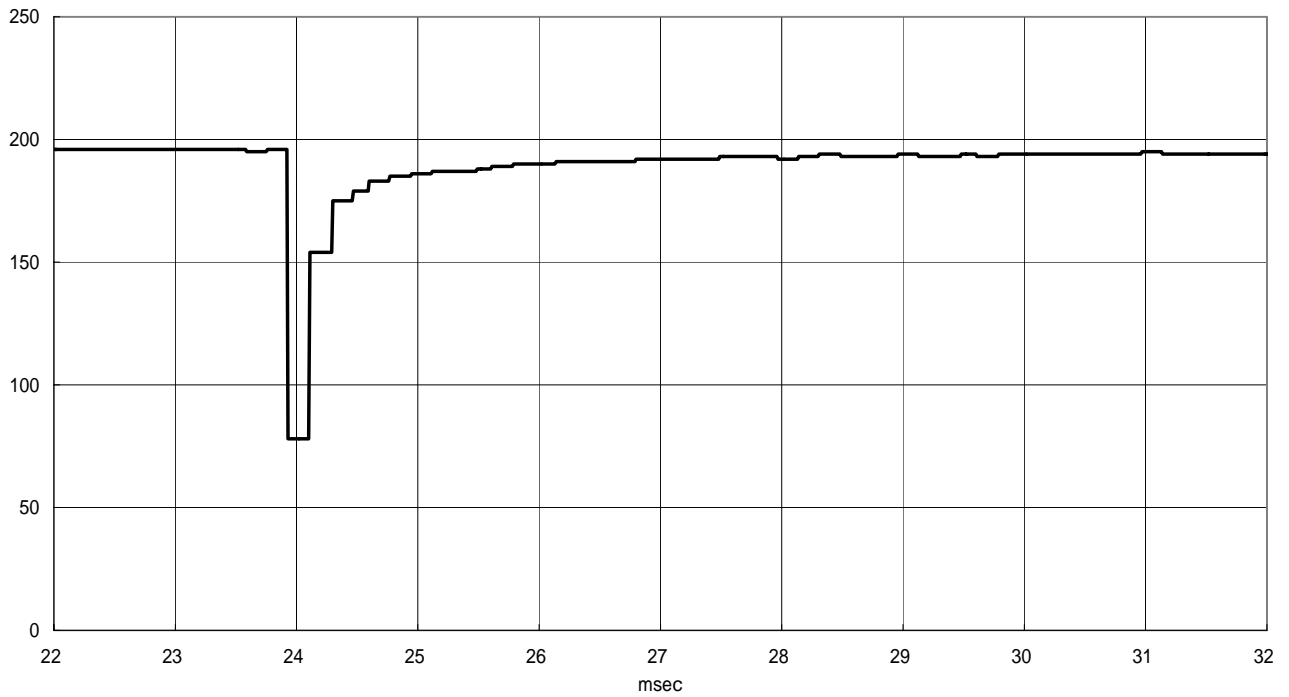
この観測では、3 回の分割発光の 2 回分を観測したものと考えられる。手動で発光と観測のタイミングを決定しているため、確かなことは分からない。

ストロボ 生API 100msec  
10  $\mu$  secサンプリング



上記データを拡大したものを以下に示す。

ストロボ 生API 10msec  
10  $\mu$  secサンプリング



#### 4 . 考察

10  $\mu$  sec サンプリングのデータをみると、同一測定値が複数回(十数回づつ)観測されている。legOS 解説によれば、AD 変換には「約 16.3  $\mu$  sec を要する」とのことであり、10  $\mu$  sec 毎にセンサ値を入力した場合は毎回測定値を入力できる訳ではない、と考えられる。しかし、十数回にわたって同一値が得られる理由は分からない。このあたりは、BrickOS、ハードウェアの調査が必要だろう。

いま少し考えるために、50  $\mu$  sec サンプリングのデータを調べてみると、2~3 回同一値が入力されている。また、100  $\mu$  sec サンプリングのデータをみても、2 回同一値が入力されている例がみられる。

以上から、現実には 100  $\mu$  sec を超える時間間隔でしか、新しい測定値を得られない、と想像される。

これらの測定は、全て手動で行っており、距離など測定条件はあまり正確ではありません。また、ストロボの計測は手でタイミングをとっているため再現状況が悪く、多くの実験をしていません。

出力ポートの出力を直流リレーに入力し、リレー接点でストロボを発光させれば、相当に再現性の良い実験ができると思う。単体ストロボ、小型直流リレーが手元があれば、再現性の良い実験ができると思うのですが、残念ながら実現していない。手元に機材がないのが残念。