

システム実験

基礎実験4レポート

6119019056 山口力也

2019/05/17 日提出

1 基礎実験第4回の概要

基礎実験第4回目の実験の目的, 実施した実験の概要, および理解した事柄を 100 200 字程度で説明せよ.

1.1 実験目的

以下の示す事柄を実験目的とした.

- 割り込みの動作原理を知る.
- ポーリングと割り込みの処理の違いを理解する.
- タイマ割り込みの動作原理を理解して使いこなせる.
- 外部割り込みの動作原理を理解して使いこなせる.

1.2 実験概要

実験内容としてマイコンを用いた割り込み処理として”タイマ割り込み”と”外部割り込み”を扱った.

1.3 理解した事柄

マイコンのプログラムをより効率的に実行する割り込み機能の動作原理を理解し, 実装方法を習得し理解した.

2 回路実装

基礎実験 4 で使用した回路のブレッドボード配線図を報告せよ。
以下 1 に使用した回路のブレッドボード配線図を示す。

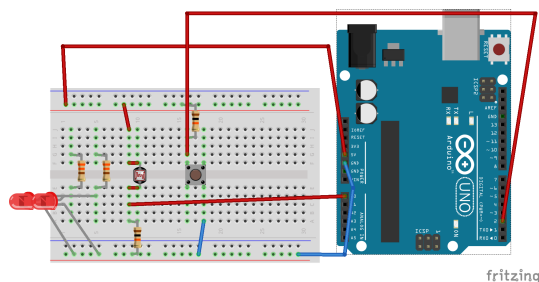


図 1: 基礎実験 4 で使用した回路の配線図

3 タイマ割り込みによる LED の点滅

課題 2.4.1 において作成したスケッチおよび完成させた図 2.63 を報告せよ。
また,MsTimer2 を使用せずにポーリング処理のみを使用した場合どのような問題点があるのか調査し, 報告せよ。

以下ソースコード 1 に作成したスケッチを示す。

ソースコード 1: 課題 2.4.1

```
1 #include <MsTimer2.h>
2 const int LED1_PIN = 13; //13番ピンにLED を接続
3 const int LED2_PIN = 9; //9番ピンにLED を接続
4
5 void flash() { //割り込みサブルーチン (LED の点滅)
6   for ( int i = 255; i >= 0 ; i--){ //発光状態から消灯状態
7     analogWrite(LED2_PIN,i); //LED2_PIN に PWM 出力
8     delay(5000/256); //256回 (発光->消灯間 5000ms)
9   }
10
11 }
12
13 void setup() {
14   pinMode(LED1_PIN, OUTPUT); //13番ピンを出力ポートに設定
15   pinMode(LED2_PIN,OUTPUT); //9番ピンを出力ポートに設定
16   MsTimer2::set(20, flash); // タイマ割り込み間隔の設定 (500
    ms)
```

```

17 MsTimer2::start(); //タイマ割り込み開始
18 }
19
20 void loop() { //ポーリング処理
21   digitalWrite(LED1_PIN,HIGH); //13番ピンに 5V 出力
22   delay(500); //500ms 待機
23   digitalWrite(LED1_PIN,LOW); //13番ピンに 0V 出力
24   delay(500); //500ms 待機
25 }

```

また、以下図 2 に完成させた図 2.63 を示す。

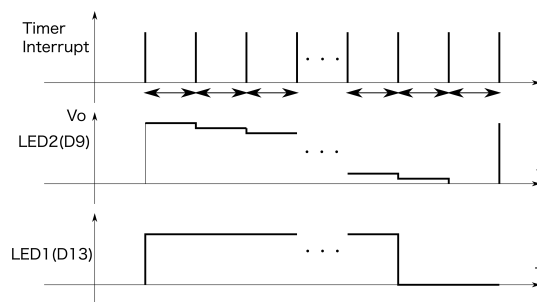


図 2: 周期の異なる LED の同時点滅

MsTimer2 を使用せずにポーリング処理のみを使用した場合は、20ms ごとに呼び出される割り込みに 500ms 待機が入っているので、うまく割り込めず 500ms ずつ処理がずれていくと思われる。

4 タイマ割り込みによる照度センサのデータ取得

課題 2.4.2 において作成したスケッチを報告せよ。また、「手で光を遮る」「光を遮らない」状態を素早く繰り返した時に生じた取りこぼしについて原因を考察せよ。さらに、取りこぼしを回避する方法案を記せ。

以下ソースコード 2 に作成したスケッチを示す。

ソースコード 2: 課題 2.4.2

```

26 unsigned long timePrev = 0; //基準となる時間を格納
27
28 void check() {
29   static int sensorvalue = analogRead(A0); //
      A0 ピンの AD 変換結果を取得
30   Serial.println(sensorvalue);

```

```

31 }
32 void setup() {
33   Serial.begin(9600); //シリアル通信を 9600kbps で初期化
34   pinMode(13,OUTPUT); //13番ピンを出力ポートに設定
35 }
36
37 void loop() { //millis 関数による時間計測
38   unsigned long timeNow = millis(); //
        millis 関数を用いて現在の時間情報を取得
39   if(timeNow - timePrev >= 500){ //500ms 以上経過
40     check(); //check 関数の処理を実行
41     timePrev = timeNow; //時間情報の更新
42   }
43   else{}
44 }

```

「手で光を遮る」「光を遮らない」状態を素早く繰り返した時に生じた取りこぼした原因は,millis 関数の値にある.500ms ごとにセンサの値を読み取っているため 500ms の間にあった変化に関してはセンサは読み込めない. 取りこぼしを回避するには millis 関数の値をより小さくして細かくセンサの値を読み取る必要があるが, 細かくしすぎると値を読み取る回数が増えて処理が重くなる.

5 スイッチ動作の外部割り込みによる LED の点灯・消灯

演習 2.4.5 において生じるチャタリングについて調査し報告せよ. また,attachInterrupt を使用せずにポーリング処理のみを使用した場合, どのような問題が生じるのか調査し, 報告せよ.

チャタリングとは, 物理的なスイッチなどを使用した時, スイッチを押した瞬間に ON と OFF の状態が非常に短い間隔で振動することである. 今回の実験では物理スイッチを用いているため,ON にした瞬間に ON と OFF が交互に動作して誤動作を起こしている.

また,attachInterrupt を使用せずにポーリング処理のみを使用した場合, ループ関数の中で処理を定義する必要がある. ループ関数で処理を行っている途中に外部割り込みがあった場合, 外部割り込みを認識できないか, 反応が遅れる.. またループ処理の中で信号を検知する場合, 常に信号があるかないかを確認する必要があるため処理が多くなる.

6 スイッチ動作に外部割り込みによる LED 発行の フェードイン・フェードアウト

課題 2.4.3 で作成したスケッチを報告せよ。またスケッチで工夫した点を工夫せよ。

以下ソースコード 3 に作成したスケッチを示す。

ソースコード 3: 課題 2.4.3

```
45 const int LED_PIN = 9; //LED を D9 に定義
46 volatile int flag = LOW; //フェードイン, フェードアウト判定用の
   変数を用意
47 int i = 0;
48 unsigned long timePrev = 0; //時間を格納
49 /* 初期設定 */
50 void setup() {
51     pinMode(LED_PIN, OUTPUT); //D9 を出力に設定
52     /*外部割り込みの設定
53     割り込みに使用するディジタルポートをD2 に設定(1番目の引
       数="0")
54     割り込みサービスルーチンblink 関数を実行(2番目の引数=関数
       名)
55     (例)割り込み要因としてスイッチの状態が"LOW"のときに発生 (3
       番目の引数)
56     */
57     attachInterrupt(0, blink, FALLING);
58 }
59
60 void loop() {
61     if ( flag == LOW ){
62         if( i == 255 ){
63             i = 0;
64         }
65         analogWrite(LED_PIN, i);
66         i++;
67         timePrev = millis();
68         while(millis() - timePrev <= 3000/255){
69             //3000/255秒待つ
70         }
71     }
72     else{
73         if( i == 0 ){
74             i = 255;
75         }
76         analogWrite(LED_PIN, i);
```

```

77     i--;
78     timePrev = millis();
79     while(millis() - timePrev <= 3000/255){
80         //3000/255秒待つ
81     }
82 }
83 }
84 void blink() { //割り込みの条件にあわせ
    LED がフェードイン・フェードアウトを繰り返す
85     flag = !flag;
86 }

```

スケッチで工夫した点は、スイッチが押された時ように flag を用意しておき、flag が変更したらそれぞれ加算か減算の for 文に入るように if 文で制限したところである。

7 発展課題 2.4.1

発展課題 2.4.1 において作成したスケッチと工夫点を報告せよ。また、照度センサの値の変化をグラフ化し、その結果を報告せよ。

以下ソースコード 4 に作成したスケッチを示す。

ソースコード 4: 発展課題 2.4.1

```

87 #include <MsTimer2.h>
88 const int LED1_PIN = 13; //13番ピンにLEDを接続
89 const int LED2_PIN = 9; //9番ピンにLEDを接続
90 volatile int flag = 0; //10秒経過flag
91 int output = LOW;
92 unsigned long timePrev10 = 0; //基準となる時間を格納(10秒用)
93 unsigned long timePrev500 = 0; //基準となる時間を格納(500
    ms 用)
94 void check() { //割り込みサービスルーチン(LEDの点滅)
95     int sensorvalue = analogRead(A0); //A0ピンのAD変換結果を取得
96     float vo = sensorvalue*(5.0/1024.0);
97     float L = 222*vo; //照度値に変換
98     Serial.println(L); //シリアルモニタに表示
99     if ( flag == 1){ //10秒経過したら
100         digitalWrite(LED2_PIN,HIGH); //LED2_PINを点灯させる
101         flag = 0; //flag初期化
102     }
103     else { //そうでないなら
104         digitalWrite(LED2_PIN,LOW); //LED2_PINは消灯
105     }

```

```

106     digitalWrite(LED1_PIN,LOW);
107 }
108 void setup() {
109     Serial.begin(9600); //シリアル通信を 9600kbps で初期化
110     pinMode(LED1_PIN, OUTPUT); //13番ピンを出力ポートに設定
111     pinMode(LED2_PIN,OUTPUT); //9番ピンを出力ポートに設定
112     MsTimer2::set(1000, check); // タイマ割り込み間隔の設定 (500ms)
113     MsTimer2::start(); //タイマ割り込み開始
114 }
115
116 void loop() { //ポーリング処理
117
118     unsigned long timeNow10 = millis(); //
119         millis 関数を用いて現在の時間情報を取得
120     if((timeNow10 - timePrev10) >= 10000){ //10秒経過
121         timePrev10 = timeNow10; //時間情報の更新
122         flag = 1; //10秒経過を知らせる
123     }
124     if((millis() - timePrev500) >= 500){ //500ms 経過
125         digitalWrite(LED1_PIN,output);
126         timePrev500 = millis(); //時間情報を更新
127         output = !output; //出力を反転
128     }
129 }

```

スケッチで工夫した点は、10 秒経過用に flag を用意して 1000ms ごとに入るタイマ割り込みの処理に LED2 を点灯させる処理を入れることで、1000ms 待たなくても自動的に待つようにしたところだが、タイマ割り込みの処理が重くなるのでよくなかったかもしれないと思っている。

また以下図 3 に照度センサの値の変化のグラフを示す。

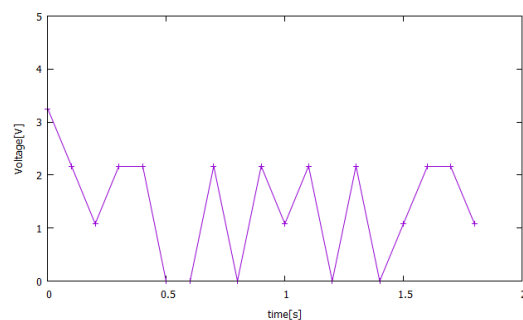


図 3: 照度センサの値の変化

8 発展課題 2.4.2

発展課題 2.4.2 において作成したスケッチと工夫点を報告せよ。また、照度センサの値の変化をグラフ化し、その結果を報告せよ。

以下ソースコード 5 に作成したスケッチを示す。

ソースコード 5: 発展課題 2.4.2

```
129 const int LED1_PIN = 13; //13番ピンにLED を接続
130 const int LED2_PIN = 9; //9番ピンにLED を接続
131 int output = LOW;
132 int flag = 0;
133 unsigned long timePrev1 = 0; //基準となる時間を格納 (10秒用)
134 unsigned long timePrev10 = 0; //基準となる時間を格納 (10秒用)
135 unsigned long timePrev500 = 0; //基準となる時間を格納 (500
    ms 用)
136 unsigned long timePrev1000 = 0; //基準となる時間を格納 (1000
    ms 用)
137 void setup() {
138     Serial.begin(9600); //シリアル通信を 9600kbps で初期化
139     pinMode(LED1_PIN, OUTPUT); //13番ピンを出力ポートに設定
140     pinMode(LED2_PIN, OUTPUT); //9番ピンを出力ポートに設定
141 }
142 void loop() { //ポーリング処理
143     if((millis() - timePrev10) >= 10000){ //10秒経過
144         timePrev10 = millis(); //時間情報の更新
145         flag = 1; //10秒経過フラグ
146     }
147     if((millis() - timePrev500) >= 500){ //500ms 経過
148         output = !output; //出力を反転
149         digitalWrite(LED1_PIN, output);
150         timePrev500 = millis(); //時間情報の更新
151     }
152     if((millis() - timePrev1000) >= 1000){ //1000ms 経過
153         int sensorvalue = analogRead(A0); //
            AO ピンの AD 変換結果を取得
154         float vo = sensorvalue*(5.0/1024.0);
155         float L = 222*vo; //照度値に変換
156         Serial.println(L); //シリアルモニタに表示
157         delay(1); //安定用
158         timePrev1000 = millis(); //時間の更新
159         if ( flag == 1){ //10秒経過したら
160             digitalWrite(LED2_PIN, HIGH); //LED2_PIN を点灯させる
161             flag = 0; //値を初期化
162         }
163         else { //そうでないなら
```



```

164     digitalWrite(LED2_PIN,LOW); //LED2_PIN は消灯
165 }
166 }
167 }

```

スケッチで工夫した点は,500ms,1000ms,10000ms 経過用にそれぞれフラグを用意してそれぞれの秒数が経過したときに動作をするようにした. この動作は delay 関数を使わずにできたが自宅の開発環境では System.println() のあとに安定用の delay を用いないとエラーがでてしまったためそこだけ delay 関数を使ってしまった.

また以下図 3 に照度センサの値の変化のグラフを示す.

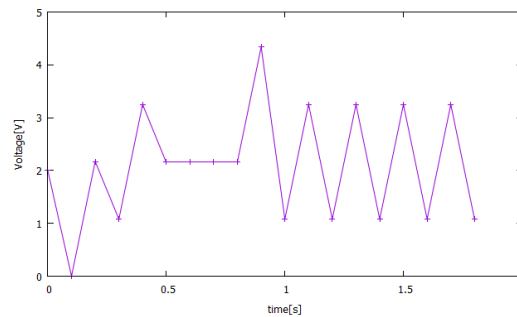


図 4: 照度センサの値の変化

9 発展課題 2.4.3

発展課題 2.4.3 において作成したスケッチと工夫点を報告せよ.
以下ソースコード 6 に作成したスケッチを示す.

ソースコード 6: 発展課題 2.4.3

```

168 const int LED_PIN = 13; //LED を D13 に定義
169 const int SW_PIN = 2; //SWITCH を D2 に定義
170 volatile int swnow = LOW; //現在の状態
171 volatile int swprev = LOW; //前の状態
172 unsigned long timePrev = 0; //基準となる時間を格納
173 volatile int output = LOW; //ディジタル出力の値を格納する変
    数:初期:LOW
174 /* 初期設定 */
175 void setup() {
176     pinMode(LED_PIN,OUTPUT); //D13 をディジタル出力に設定
177     pinMode(SW_PIN,INPUT);
178     /*外部割り込みの設定

```

```

179     割り込みに使用するデジタルポートをD2 に設定(1番目の引
        数="0")
180     割り込みサービスルーチンblink 関数を実行(2番目の引数=関数
        名)
181     (例)割り込み要因としてスイッチの状態が"LOW"のときに発生 (3
        番目の引数)
182     */
183     attachInterrupt(0,blink,FALLING);
184 }
185
186 void loop() {
187     swprev = swnow;
188     swnow = digitalRead(SW_PIN);
189     if(swprev == LOW || swnow == HIGH){//ON から OFF になったとき
190         timePrev = millis();
191         while(millis() - timePrev >= 10){
192             //10ms 待つ
193         }
194     }
195     digitalWrite(LED_PIN,output); //スイッチを押すたびに点灯・消
        灯を繰り返す
196 }
197
198 void blink() { //割り込みの条件にあわせ
        LED が点灯・消灯を繰り返す
199     timePrev = millis();
200     output = !output; //デジタル出力を反転
201     while(millis()-timePrev >= 10 ){
202         //10ms 待つ
203     }
204 }

```

スケッチで工夫した点は,swprev と swnow を使って ON から OFF 状態に推移した時と OFF から ON 状態に推移した時にチャタリング防止に 10ms 待つプログラムとしたところである. しかしながらこのプログラムでは入力に遅延が生じてしまうため, もう少し他の解決策も考えられる.