

実験テーマ	制御工学実験 I <u>マイコン基礎・応用</u>
所属・氏名 (共同実験者名は括弧内)	熊本高等専門学校 制御情報システム工学科 <u>4 年 21 番 氏名 下石 龍生</u>
実験場所	<u>5 号棟 3 階 実験室</u>
実施日(第 1 週, 第 2 週)	<u>令和 3 年 7 月 13 日(火曜日), 令和 3 年 7 月 20 日(火曜日)</u>
レポート締切日	<u>令和 3 年 8 月 3 日(月曜日)</u>
レポート提出日	<u>令和 3 年 8 月 3 日(月曜日)</u>

評価項目 (A: 達成できている, B: 概ね達成できている, C: ほとんど達成できていない, D: 達成できていない)		自己評価 (A~D)	担当評価 (A~D)
実施評価	実験開始までに実験テキストや実験ノートを準備できており, 事前課題がある場合は, それに取り組んでいた.	A	
	担当者による指示をよく聞き, 不注意による無用な誤りなく安全に実験を行うことができた.	A	
	回路やプログラムを自分で作成し, グループワークの場合は自らの役割を全うするなど, 課題に対して積極的に取り組むことができた.	A	
	与えられた課題を時間内に達成し, 結果を正確に記録または出力できた.	A	
	使用器具の後片付けや実験場所の清掃をきちんと行った.	A	
レポート評価	章立ては適切であり, それぞれの章における記載内容は <u>自作のものである</u> . 引用がある場合は, その旨を明記している.	A	
	図・表の書き方は裏面の要領に準じており, <u>自作のものである</u> . (担当者が許可しない限り, 指導書の図すら引用してはいけない)	A	
	使用器具や実験環境について, 実験結果を再現するのに十分な情報を記載している.	A	
	課題に関する計測結果や出力結果を整理して記載し, 結果に対する独自の考察を述べている.	A	
	研究課題に取り組み, 適切な参考文献を基に答えを導き出している.	A	

※提出期限に遅れた場合, 遅れた週の数に応じて減点する.

実施点 (50)	レポート点 (50)	合計点 (100)
-------------	---------------	--------------

1 実験目的

本実験では、Arduino を用いた自動ブラインドシステムを構築する。その中でマイコンの割り込み機能について学習し、割り込み技術を習得することを目的とする。

2 概要

2.1 外部割り込み

外部割り込み [2] とは、外部からの信号で割り込みが発生する仕組みのことである。割り込みが発生すると、CPU は通常の命令の実行を中断し、割り込みサブルーチンを起動する。割り込みサブルーチンとして準備されたプロセスが完了すると、元の命令に戻り、途中から実行を再開する。外部割り込みを発生させるデバイスの例として、マウス、キーボード、スイッチなどが挙げられる。

2.2 タイマー割り込み

タイマー割り込み [3] とは、設定した時間間隔で割り込みを発生させる仕組みであり、定期的に発生させたいサブルーチンを呼び出すことができる。割り込みの動作は外部割り込みなどのほかの割り込みプロセスと変わらない。

3 構築したシステム

3.1 ハードウェア

以下の図 1 に作成した回路を示す。

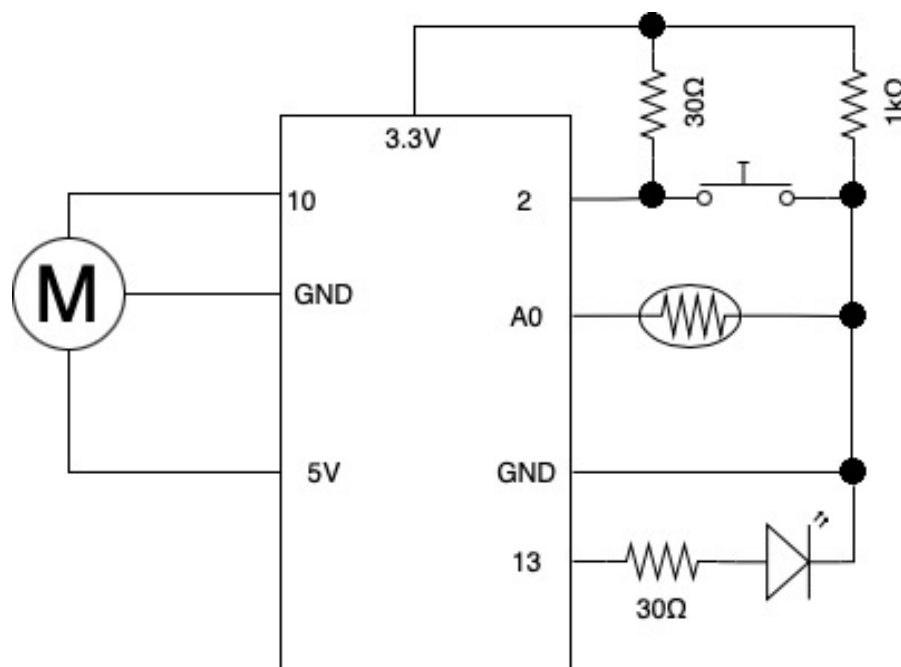


図 1: 作成した回路

Arduino Uno のピン 10, GND, 5V にはサーボモータを接続した。また、ピン 13 には LED とそれに伴う抵抗 (30Ω) を接続した。フォトトランジスタとタクトスイッチにはプルダウン抵抗を接続し、出力状態、センサの状態を反転させている。よって、タクトスイッチを押したときのみ、Arduino で ON を読み取る

ことができる。

3.2 ソフトウェア

以下の Code1 に作成した Arduino のソースコードを示す。

Code 1: Arduino で作成した Code

```
1 #include "Servo.h"
2 #include "MsTimer2.h"
3
4 Servo myservo;
5 void brind(){
6   if(analogRead(A0)<450){
7     myservo.write(0);
8   }else if(analogRead(A0)>510){
9     myservo.write(180);
10  }
11 }
12
13 void setup() {
14   Serial.begin(9600);
15   pinMode(2, INPUT);
16   pinMode(13, OUTPUT);
17
18   attachInterrupt(0,interrupt,FALLING);
19
20   myservo.attach(10);
21   MsTimer2::set(100, brind);
22   MsTimer2::start();
23 }
24
25 void loop() {
26   digitalWrite(13, HIGH);
27   delay(1000);
28   digitalWrite(13, LOW);
29   delay(1000);
30 }
31
32 void interrupt(){
33   Serial.println(analogRead(A0));
34 }
```

1 行目, 2 行目ではサーボモータとタイマ割り込みのためのライブラリをインポートしている。5 ~ 11 行目ではソフトウェア起動時の初期動作を表している。シリアルモニタを起動後, ピン 2 をフォトレジスタの入力ポートとして割り当て, ピン 13 を LED の出力ポートとして割り当てている。25 ~ 30 行目では L チカを行う動作を書いている。18 行目にある “attachInterrupt(0, interrupt, FALLING)” は, 外部割り込みを命令するコードである。タクトスイッチが押されたとき, 32 行目から記している自作の interrupt 関数を動かしている, interrupt 関数の中では, シリアルモニタにフォトランジスタから読み取った明るさの数値を表示している。また, 21, 22 行目ではタイマ割り込みを命令するコードである。100ms 周期で自動的に自作の brind 関数を動かしている。brind 関数では, フォトランジスタから読み取った明るさの数

値からサーボモータを動かす動作を記している。割り込み動作と、メインの LOOP 文を用いることで、バラバラの周期で、自分が指定したタイミング、周期でプログラムを動作させている。

4 考察

4.1 ハードウェア

最初に回路を作成したとき、タクトスイッチを押していない間に連続してフォトトランジスタで読み取った値が出力され、スイッチを押したときのみ値の出力が停止する動作をした。その原因が、プルダウン抵抗を回路に組み込んでいなかったことであった。プルダウン抵抗を挟むことにより、マイコンは常に GND に接続され、スイッチを押したときのみ入力加わる回路ができる。以下の図 2 にプルアップ抵抗の回路図とプルダウン抵抗の回路図を示す。

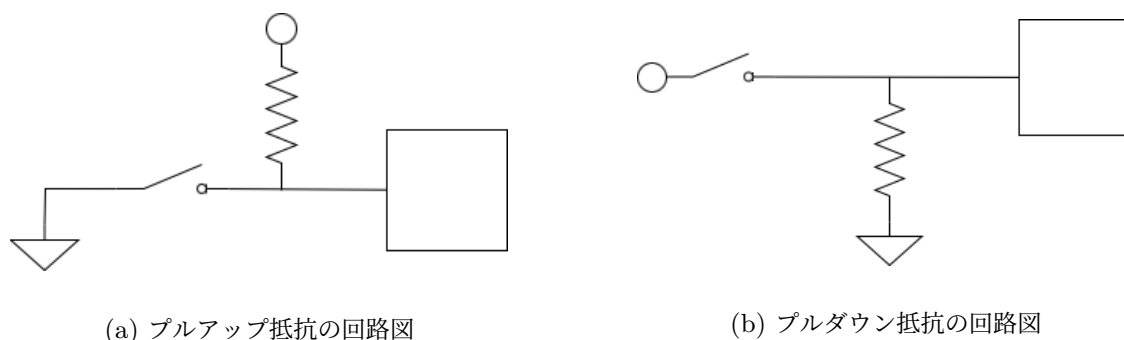


図 2: プルアップ抵抗とプルダウン抵抗の回路

プルアップ抵抗もプルダウン抵抗も、マイコンが常に抵抗に接続されるようになり、「浮く」状態を防ぐ事ができる。したがって、回路の動作を安定させることができる。

4.2 ソフトウェア

今回の実験で作成したプログラムにはタイマ割り込みと外部割り込みを用いている。これらの使い方と動作に与える影響を考察する。

Code 2: 割り込みの使い方

```
1 attachInterrupt(interrupt,function,mode);  
2 MsTimer2::set(sec,function);
```

Code2 の 1 行目には外部割り込み、2 行目にタイマ割り込みの構文を示した。attachInterrupt では、割り込み番号、割り込んだ際にする動作関数、割り込みを発生させるトリガを指定する。Code1 では、割り込み番号 0 で、interrupt という名前の自作関数を動かしている。このとき、割り込みを発生させるトリガとして FALLING を指定しているが、トリガには他にも複数の種類が存在する。以下の表 1 にトリガの種類を示す。 [5]

表 1: トリガの種類

トリガの名前	概要
LOW	ピンが LOW のとき発生
CHANGE	ピンの状態が変化したときに発生
RISING	ピンの状態が LOW から HIGH に変わったときに発生
FALLING	ピンの状態が HIGH から LOW に変わったときに発生

2 行目のタイマ割り込み [3] では、周期 (ms) と動作させる関数を指定している。Code1 では 100ms 周期で bring という自作関数を呼び出しており、100 ミリ秒ごとに LED を点灯させる関数を動作させている。

どちらの関数も割り込みを行うことでメインの処理を一時中断して割り込み動作を行っている。Code1 で割り込んでいる関数の動作は単純な動作であり、メインに影響を及ぼすことは少ない。しかし、プログラムに更に多くの機能を追加していき、割り込みの中に更に割り込みを入れたり、割り込み関数の中に処理量が大きなものを入れたりすることで、プログラマが予期しない動作をしすることがある。これらを防ぐためには、処理時間を考慮した余裕のある割り込みを行ったり、割り込みで行う処理を軽くするなどの対応をしてソフトウェアの安定化を図る必要がある。

5 チャタリングとは

トグルスイッチや押しボタンスイッチなどの機械式スイッチでは、チャタリング [4] という現象が生じる。チャタリングが生じた際の波形の様子を以下の図 3 に示す。

チャタリングとはスイッチを押したときに接点がピタッと 1 度で接続されず、バウンド、もしくは擦れが起こることによる現象で、チャタリングが発生時は複数回 ON、OFF の切買いが発生し、最終的に ON に落ち着く。

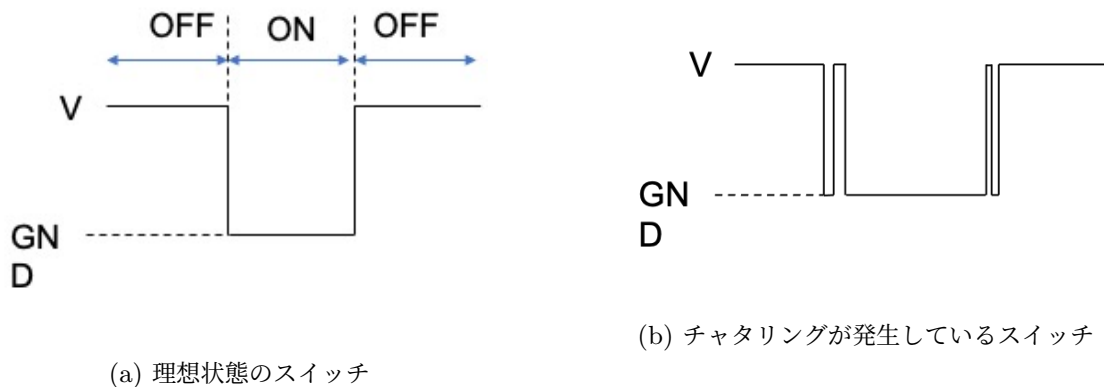


図 3: スイッチによる信号状態 p

参考文献

- [1] 熊本高等専門学校 制御情報システム工学科授業資料, “マイコン基礎・応用” (最終閲覧日 2021 年 7 月 6 日)
- [2] Tech Village, “ハードウェアの仕組みとソフトウェア処理 —— マイコンの動作を理解する” (最終閲覧日 2021 年 7 月 6 日)

http://www.kumikomi.net/archives/2009/11/post_23.php?page=5

- [3] Arduino 日本語リファレンス, “MsTimer2(ミリ秒単位で指定するタイマ)” (最終閲覧日 2021 年 8 月 3 日)

<http://www.musashinodenpa.com/arduino/ref/index.php?f=1&pos=2027>

- [4] marutsu, “スイッチのチャタリングの概要” (最終閲覧日 2021 年 7 月 13 日)

https://www.marutsu.co.jp/pc/static/large_order/1405_311_ph

- [5] Arduino 日本語リファレンス, “attachInterrupt(interrupt, function, mode)” (最終閲覧日 2021 年 8 月 3 日)

<http://www.musashinodenpa.com/arduino/ref/index.php?f=0&pos=3069>