

A decorative graphic on the left side of the slide, consisting of white lines and circles on a blue background, resembling a circuit board or a network diagram.

平成27年度 東京工業大学ロボット技術研究会 回路講習会③

日時:平成27年5月21日(木) 18:00~19:30

場所:S221

講義担当者:けり @Ryokeri14

第3回 全内容

1. 宿題(マルチバイブレータ)の解説
2. デジタル回路入門
3. デジタル出力
4. デジタル出力同士をつないではいけない
5. デジタル出力端子を使う
6. デジタル入力
7. スイッチとデジタル入力
8. スイッチのチャタリング
9. 遅延回路
10. 論理回路とは
11. 代表的な論理回路「AND」「OR」「NOT」
12. その他の論理回路「NAND」「NOR」「XOR」
13. なぜ「その他」なのか
14. 論理回路を使う
15. 汎用ロジックIC
16. ICにつけるパソコンの存在
17. モーター
18. モーターの種類
19. モーターを動かす
20. PWM制御
21. Hブリッジ回路
22. Hブリッジ回路の注意①デッドタイム
23. Hブリッジ回路の注意②FETの電圧
24. 回路修正後のモーター駆動表
25. モーター駆動ロジック回路
26. モータードライバICとは
27. モータードライバIC:TA7291P
28. サーボモーター
29. シリアル通信
30. シリアル通信の種類
31. シリアル通信のやり方
32. 宿題(加算器)

第3回 目次

1. [宿題の解説](#)
2. [デジタル回路入門](#)
3. [論理回路](#)
4. [モーター](#)
5. [PWM制御](#)
6. [Hブリッジ回路](#)
7. [モータードライバ](#)
8. [シリアル通信](#)
9. [宿題](#)

目次に戻る

※クリックするとそのページにジャンプします。

第3回 目次

1. [宿題の解説](#)
2. [デジタル回路入門](#)
3. [論理回路](#)
4. [モーター](#)
5. [PWM制御](#)
6. [Hブリッジ回路](#)
7. [モータードライバ](#)
8. [シリアル通信](#)
9. [宿題](#)

目次に戻る

1. マルチバイブレータ

※クリックするとそのページにジャンプします。

宿題の解説

- 周期T[s]は

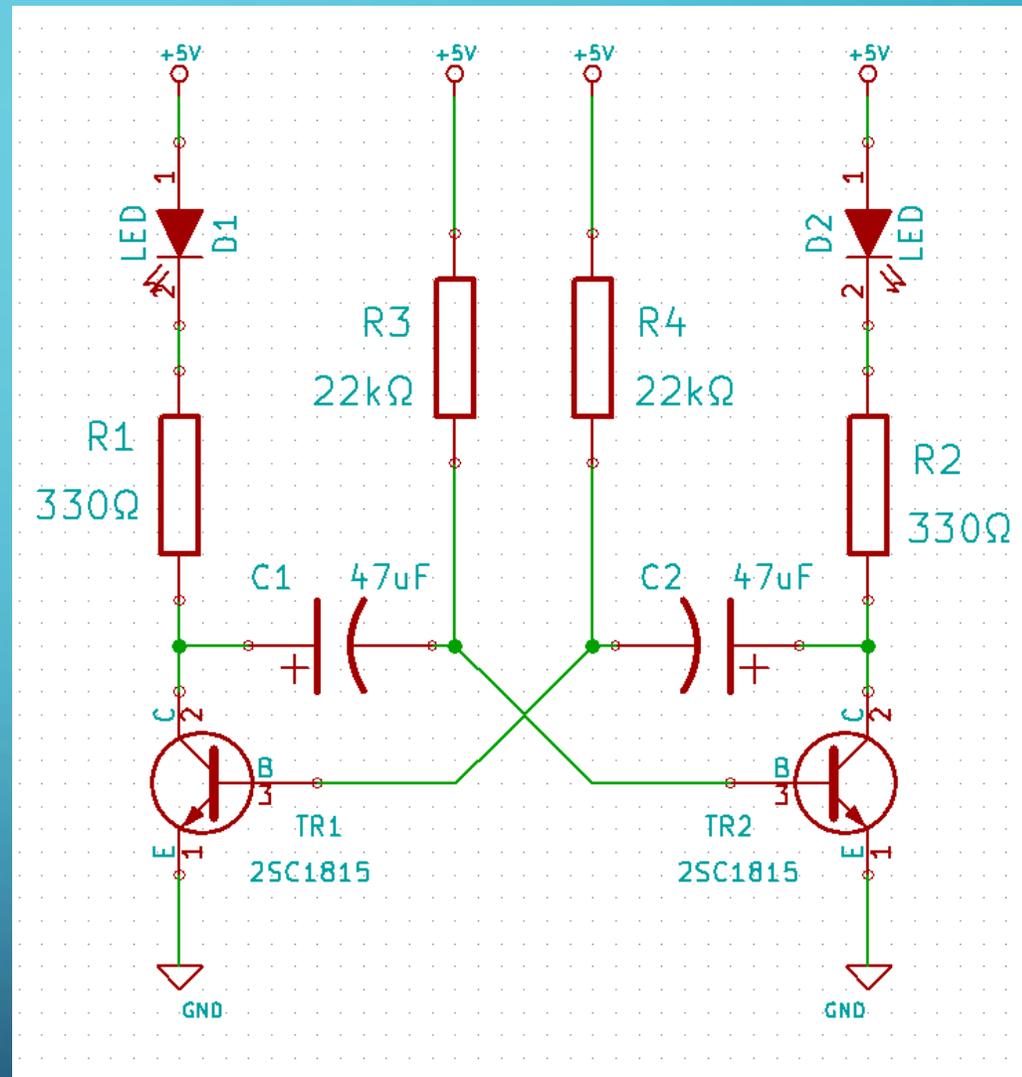
$$T = \log_2(C1 \times R3 + C2 \times R2) \text{ [s]}$$

- よって、右の回路の場合は

$$\begin{aligned} T &= 0.69 \times 0.0001 \times 22000 \times 2 \\ &= 3 \text{ [s]} \end{aligned}$$

となるはず.

- 点滅の仕組みは、難しいので、知りたい人は個人的に聞きに来てください.



第3回 目次

1. [宿題の解説](#)
2. [デジタル回路入門](#)
3. [論理回路](#)
4. [モーター](#)
5. [PWM制御](#)
6. [Hブリッジ回路](#)
7. [モータードライバ](#)
8. [シリアル通信](#)
9. [宿題](#)

目次に戻る

1. デジタル回路入門
2. デジタル出力
3. デジタル入力
4. スイッチとデジタル入力
5. スイッチのチャタリング
6. 遅延回路

※クリックするとそのページにジャンプします。

デジタル回路入門

- デジタル回路は0と1からなる.
- 電氣的には, 0がGND電位で, 1がVCC電位を指すことが多い.
- マイコンの中身はデジタル回路でできている.

デジタル出力

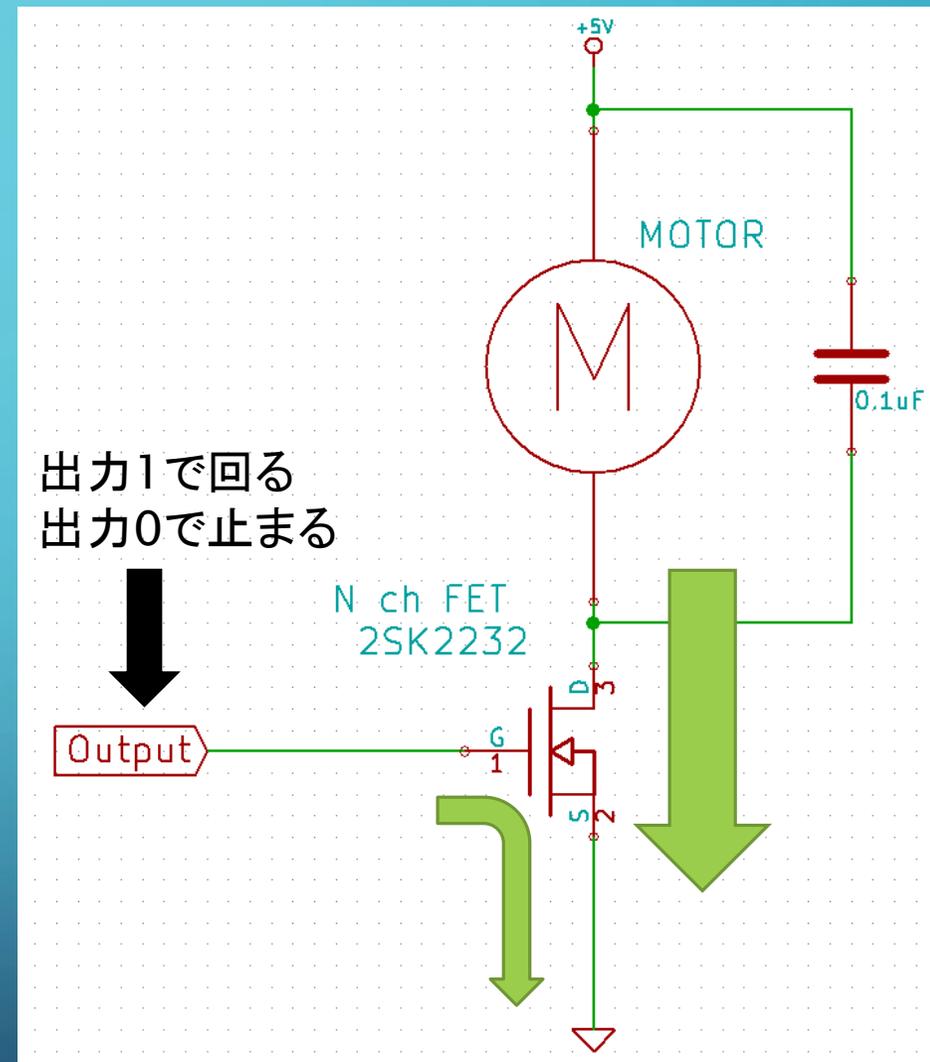
- デジタル出力は0, 1の出力であるが, 電子回路では0をGND電位, 1をVCC電位として使うことが多い.
- デジタル出力は「電流を流す」, 「流さない」ではなく, 端子が「VCCに**接続される**」, 「GNDに**接続される**」ということを表す.
- デジタル出力端子を使わないとき, その端子には**何もつながなくてよい**.

デジタル出力同士をつないではいけない

- デジタル出力端子は, GNDにもVCCにもつながり得る.
- 2つのデジタル出力ピンを直でつなぐと, ショートする恐れがある.
→マイコンが壊れる
- デジタル出力端子を電源端子につなぐのもNG.
→マイコンが壊れる

デジタル出力端子を使う

- デジタル出力端子の最大電流は20mA程度である(ものによって違う)
→LED数個ならば光る.
- モーターなどの大電流を要するものは動かない.
→トランジスタ, FETを使おう.

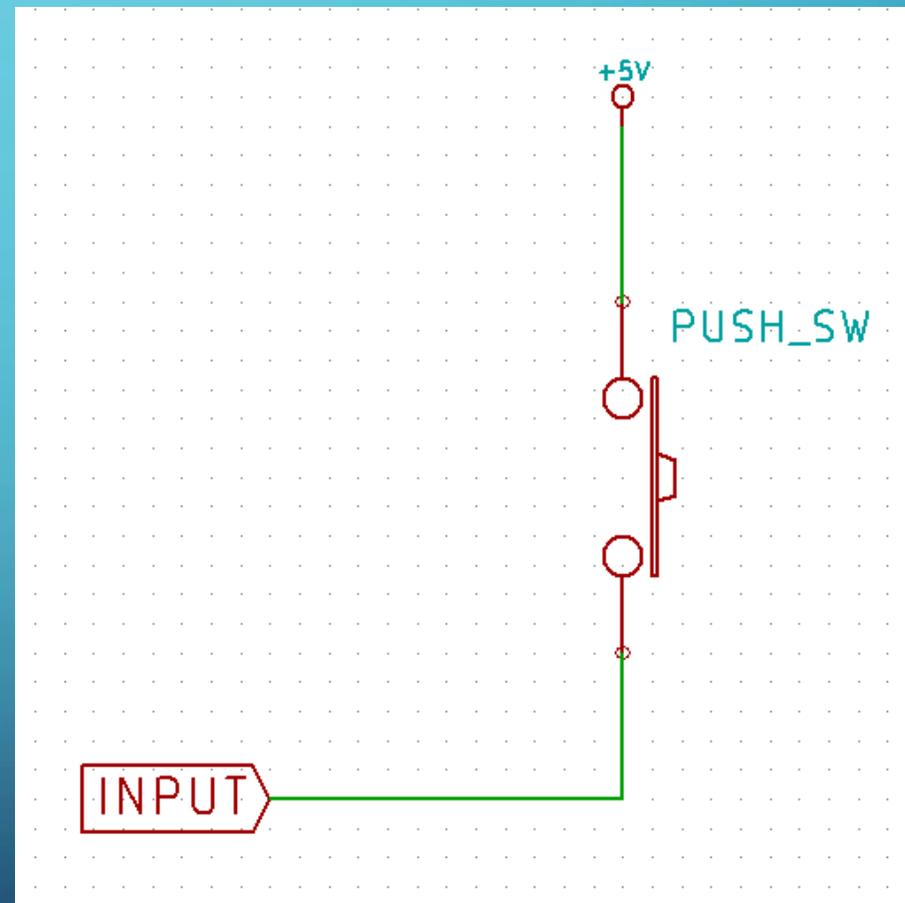


デジタル入力

- デジタル入力端子はピンの電圧が0Vか5Vかを読み取る.
- デジタル入力端子に何もつながないとどうなるか？
 - 電圧が安定しない.
 - それでも0Vか5Vかを読み取ろうとする.
 - 読み取り結果が0Vだったり5Vだったりまちまち.
 - マイコンに負担がかかる.
 - マイコンが熱くなる.
 - 最悪, マイコンが壊れる.
- デジタル入力端子を使わないときはVCCまたはGNDにつないでおこう.

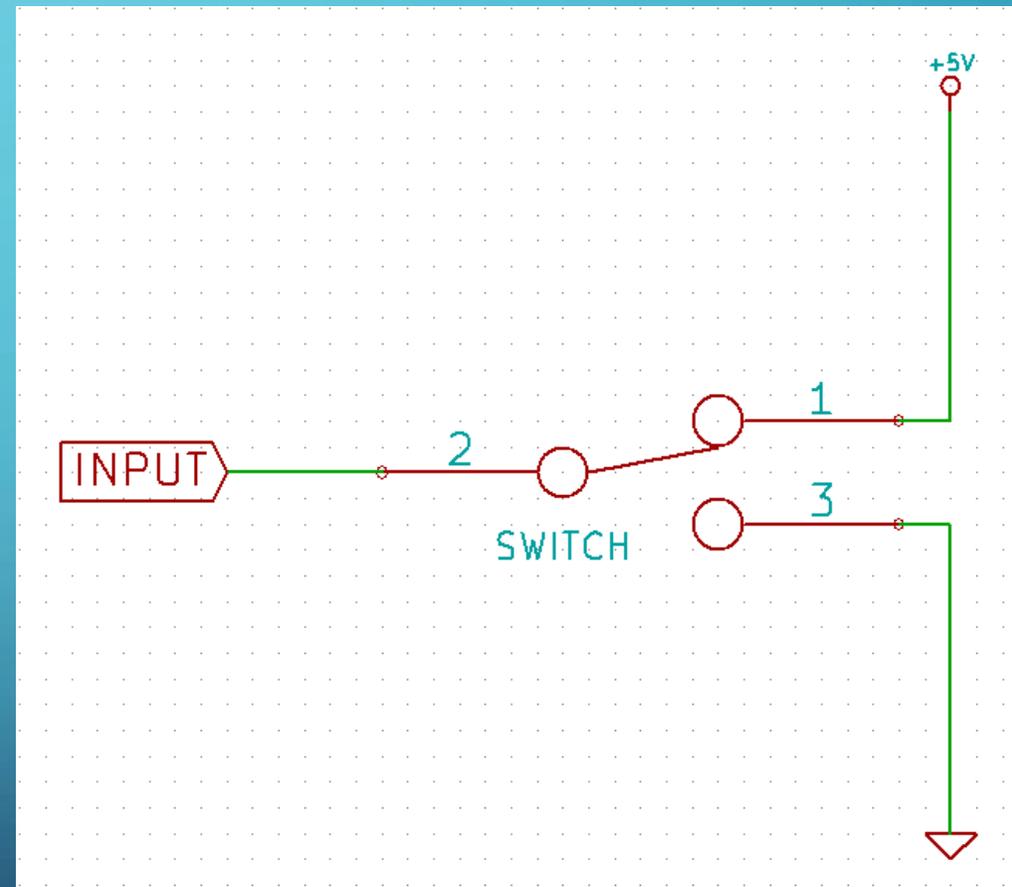
スイッチ と デジタル入力 ①

- スイッチを押すと, INPUTはVCCにつながる.
 - いい感じに思える
- スイッチを押していないとき, INPUTはなにもつながれていない
 - デジタル入力がオープン状態
 - 誤作動
- **つまり, これはダメな例**



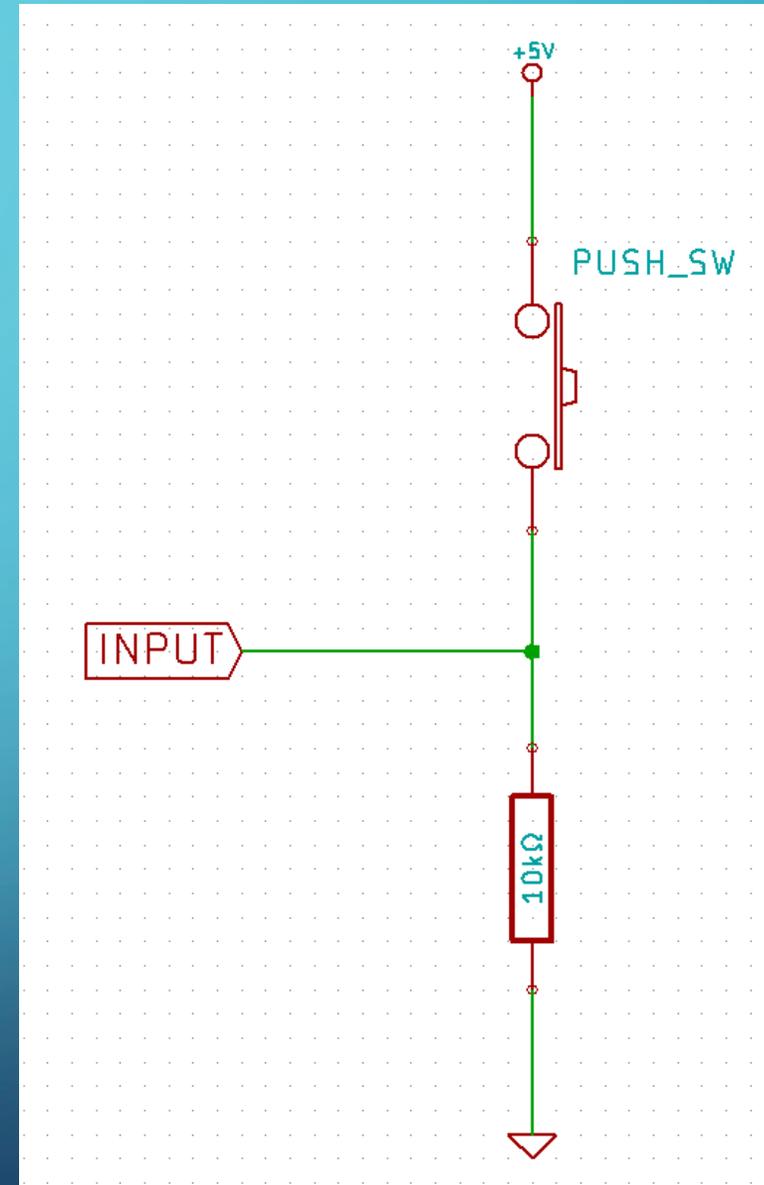
スイッチ と デジタル入力 ②

- 次にこんな回路を考える.
- スイッチを押している間GND
- スイッチをはなすとVCC
→いい感じに思える
- では, スイッチを押している途中は?
 - 微小時間だがオープン状態になる.
 - たとえ微小でも, 誤作動につながる.
- **つまり, これもダメな例**



スイッチ と デジタル入力 ③

- 10k程度の大きめの抵抗でVCCかGNDにつないでおく.
- スイッチを押す前はGND
- スイッチを押したらVCC
→これでOK
- VCCにつなぐ抵抗をプルアップ抵抗,
GNDにつなぐ抵抗をプルダウン抵抗という.
- これは電子工作でとても重要なことなので覚えておこう.



スイッチのチャタリング

- チャタリングとは

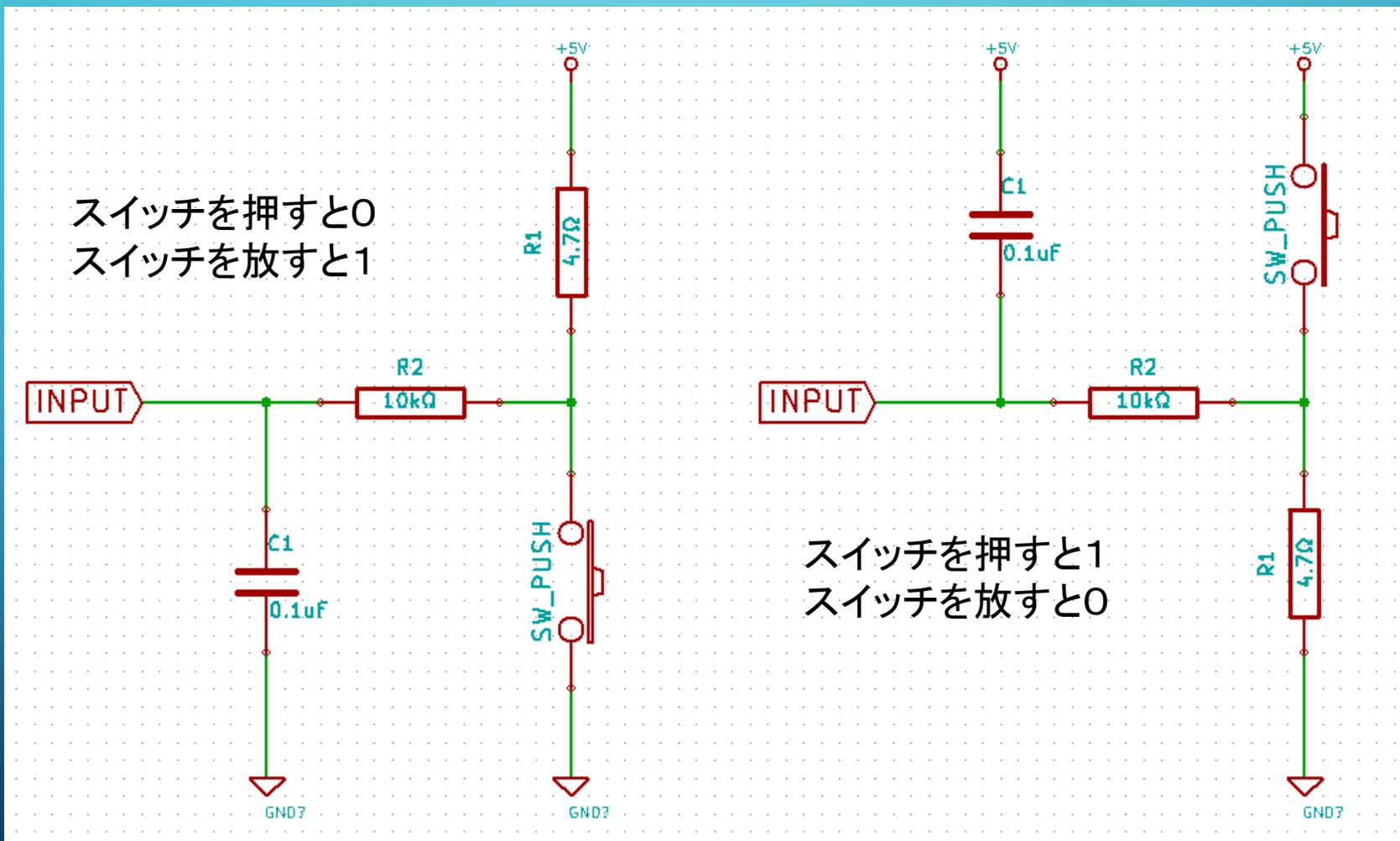
- 人間がスイッチを1回だけ押したと思っても、実際は**スイッチのバウンド**により何回も押されていることがある。
- マイコンは非常に**敏感**だから、それをしっかり数えてしまう。

- チャタリングの対策

- ハードウェアで対策→コンデンサと抵抗を使って**遅延回路**を組む。
- ソフトウェアで対策→数ミリ秒時間に**余裕をもって**スイッチのデータを読む。

遅延回路

- コンデンサが充放電されるまで、状態を維持する回路



第3回 目次

1. [宿題の解説](#)
2. [デジタル回路入門](#)
3. [論理回路](#)
4. [モーター](#)
5. [PWM制御](#)
6. [Hブリッジ回路](#)
7. [モータードライバ](#)
8. [シリアル通信](#)
9. [宿題](#)

目次に戻る

※クリックするとそのページにジャンプします。

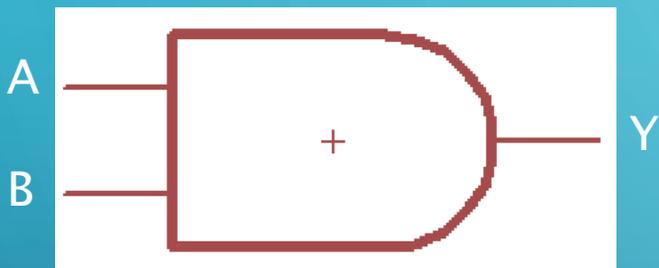
1. 論理回路とは
2. 代表的な論理回路
「AND」「OR」「NOT」
3. その他の論理回路
「NAND」「NOR」「XOR」
4. なぜ、「その他」なのか
5. 論理回路を使う
6. 汎用ロジックIC 74HCxx
7. ICにつけるパソコンの存在

論理(ロジック)回路とは

- デジタル回路は0と1から構成されている.
- 論理回路はその最小単位.
- 論理回路を組み合わせることでデジタル回路を構成できる.

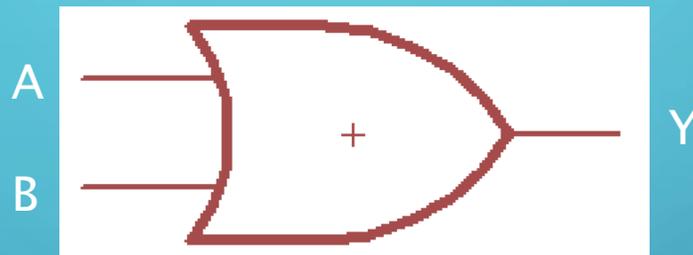
代表的な論理回路「AND」「OR」「NOT」

AND回路



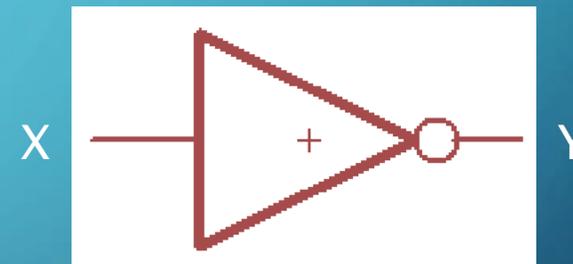
入力A	入力B	出力Y
0	0	0
0	1	0
1	0	0
1	1	1

OR回路



入力A	入力B	出力Y
0	0	0
0	1	1
1	0	1
1	1	1

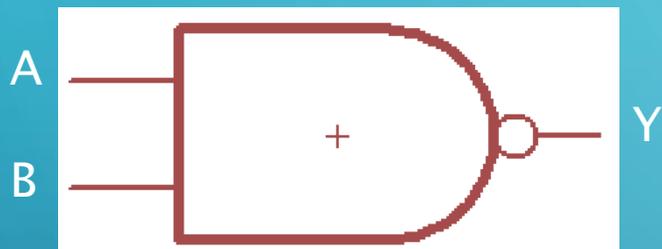
NOT回路



入力X	出力Y
0	1
1	0

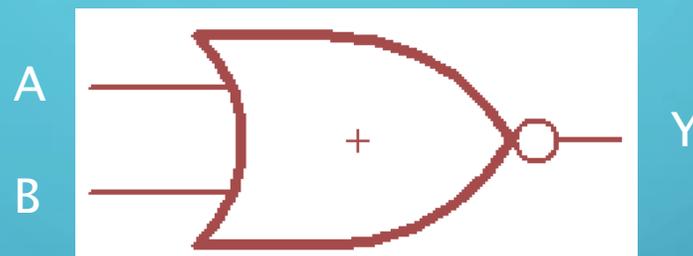
その他の論理回路「NAND」「NOR」「XOR」

NAND回路



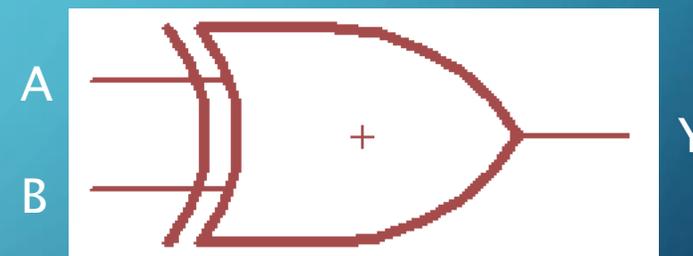
入力A	入力B	出力Y
0	0	1
0	1	1
1	0	1
1	1	0

NOR回路



入力A	入力B	出力Y
0	0	1
0	1	0
1	0	0
1	1	0

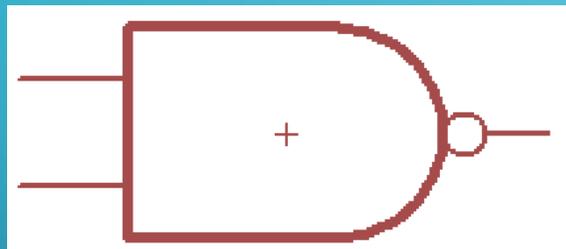
XOR回路



入力A	入力B	出力Y
0	0	0
0	1	1
1	0	1
1	1	0

なぜ「その他」なのか

NAND回路



=

AND回路

+

NOT回路

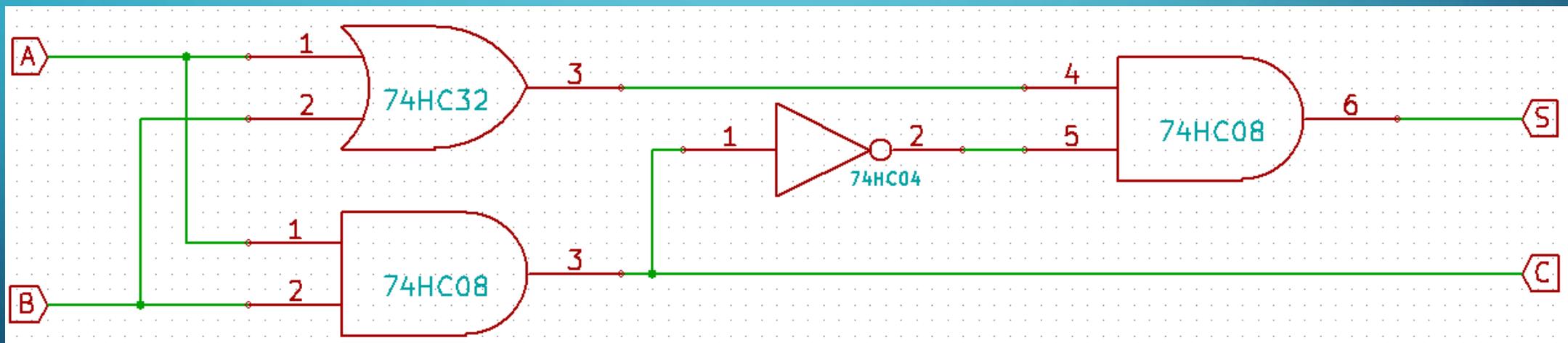


- 同様に、「NOR」「XOR」も「AND」「OR」「NOT」だけで作れる
- さらに言うと、「NAND」だけもしくは「NOR」だけですべての論理回路を組むことができる。

論理回路を使う

1 bit半加算器

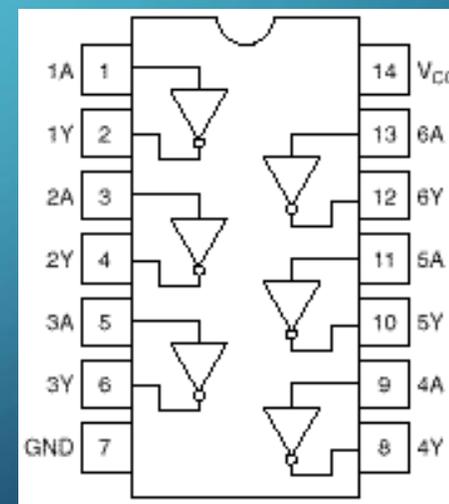
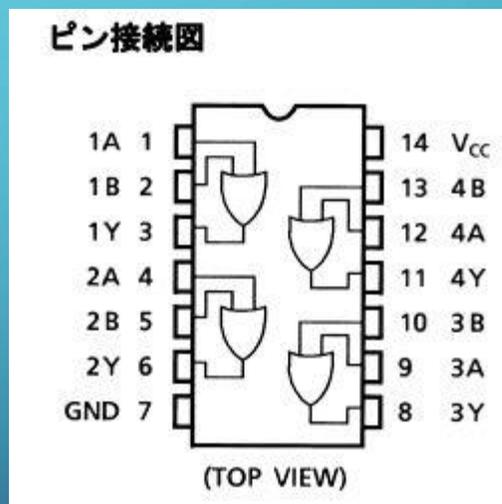
入力A	入力B	出力S	出力C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



汎用ロジックIC 型番: 74HCxx

ひとつのICの中に同じ論理回路がいくつか入っている

論理回路	ICの型番	論理回路の入数
AND回路	74HC08	4
OR回路	74HC32	4
NOT回路	74HC04	6
NAND回路	74HC00	4
NOR回路	74HC02	4
XOR回路	74HC86	4



ICにつけるパスコンの存在



- IC(マイコンも)の電源には、必ず**パスコン**(バイパスコンデンサ)をつける.
- コンデンサの種類はセラミックコンデンサで、容量は0.1 uF.
- ICは出力をコントロールする. その時、パスコンがないと**電気が足りなく**なって高速な反応ができなくなり、誤作動してしまう.
- できるだけICの**近く**にパスコンをつけた方が効果大きい.
- とにかく、**ICの電源には、パスコンをつけよう.**

第3回 目次

1. [宿題の解説](#)
2. [デジタル回路入門](#)
3. [論理回路](#)
4. [モーター](#)
5. [PWM制御](#)
6. [Hブリッジ回路](#)
7. [モータードライバ](#)
8. [シリアル通信](#)
9. [宿題](#)

目次に戻る

1. モーターとは
2. モーターの種類
3. モーターを動かす

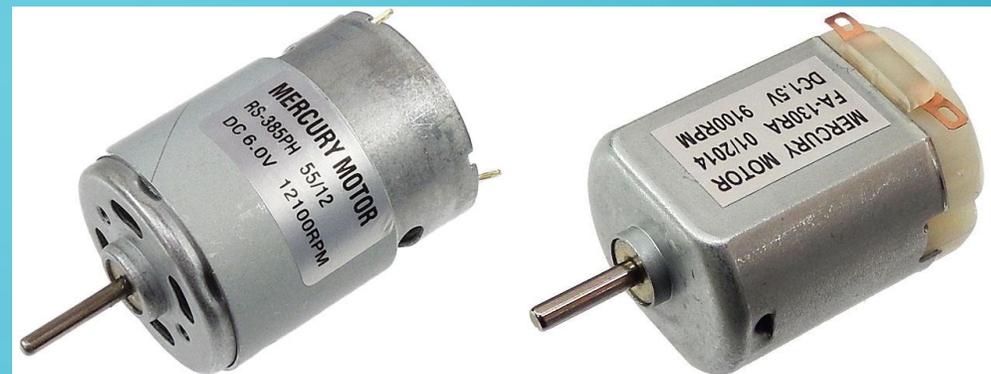
※クリックするとそのページにジャンプします。

モーターとは

- 電気エネルギーを運動エネルギーに変える.
- 大きな電流が流れる.
 - マイコンに直接つなげない
→FETかモータードライバICを使う.
- ギアを使って減速する.
 - トルク(力)を大きくする.

モーターの種類

- DCモーター と ACモーター
- ブラシモーター と ブラシレスモーター
- ステッピングモーター
- サーボモーター
- 振動モーター



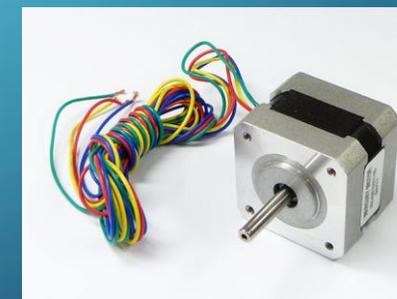
DCブラシモーター



振動モーター



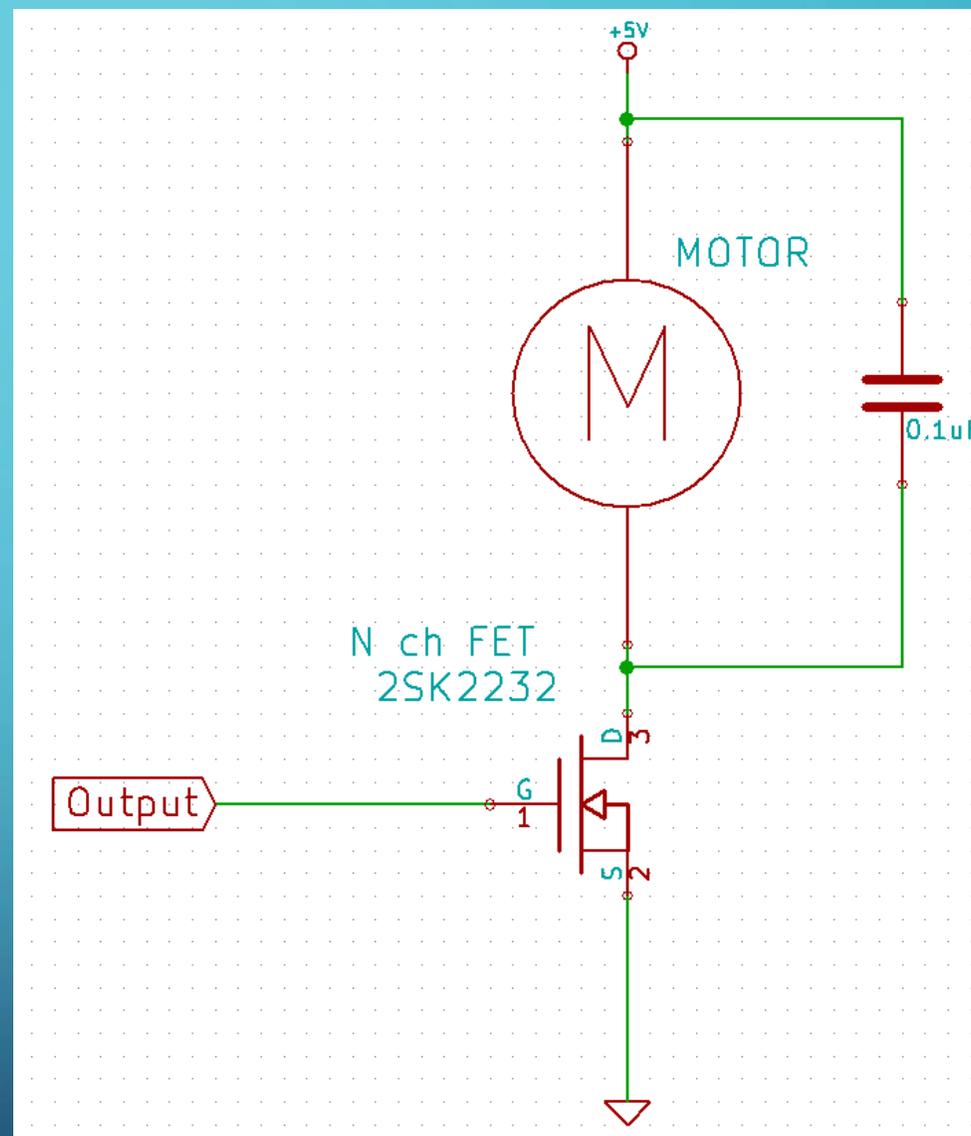
サーボモーター



ステッピングモーター

モーターを動かす

- これだけでもモーターは回る.
- モーターは1方向にしか回らない.
 - 進んだら戻れない...
 - あとで双方向回転回路を解説



第3回 目次

1. [宿題の解説](#)
2. [デジタル回路入門](#)
3. [論理回路](#)
4. [モーター](#)
5. [PWM制御](#)
6. [Hブリッジ回路](#)
7. [モータードライバ](#)
8. [シリアル通信](#)
9. [宿題](#)

目次に戻る

1. PWM制御とは
2. PWM制御

※クリックするとそのページにジャンプします。

PWM制御とは

- PWMとは

- Pulse Width Modulation の略. 日本語訳は「パルス幅変調」.
- パルス波とは, 0と1のみのデジタルな波のこと

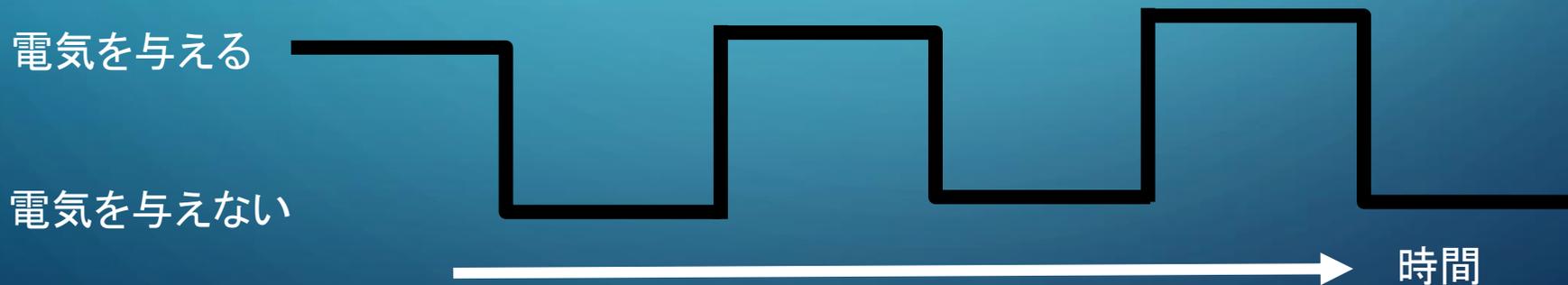
- なにに使うのか

- モーターの速さを調節する.
- LEDの明るさを調節する.



PWM制御

- モーターなどの駆動にパルス波を出力して、電流を断続的に流す.
- こうすることで、相対的に電流を与えてる時間が減るから、モーターはゆっくりと回る.
- モーターがカクカク動きそうだが、周波数を大きくすればなめらかになる.



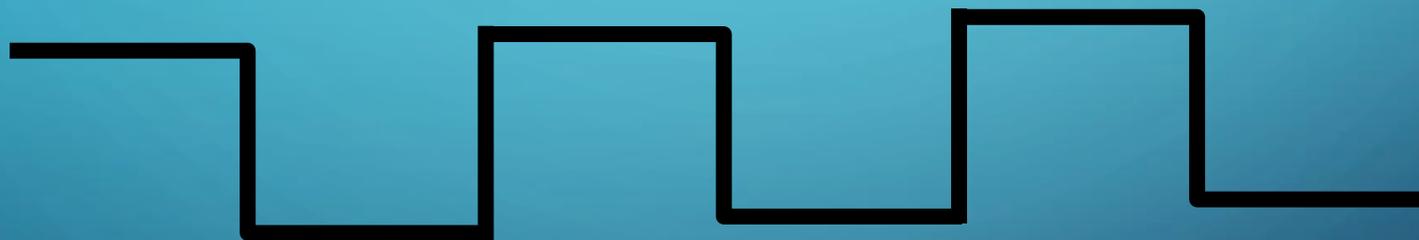
PWM制御

- パルス波のHighの割合をデューティ比(Duty比)という.

Duty比
20%



Duty比
50%



Duty比
80%



第3回 目次

1. [宿題の解説](#)
2. [デジタル回路入門](#)
3. [論理回路](#)
4. [モーター](#)
5. [PWM制御](#)
6. [Hブリッジ回路](#)
7. [モータードライバ](#)
8. [シリアル通信](#)
9. [宿題](#)

目次に戻る

1. Hブリッジ回路
2. Hブリッジの注意①
3. Hブリッジの注意②
4. 回路修正後のモーター駆動表
5. モーター駆動ロジック回路

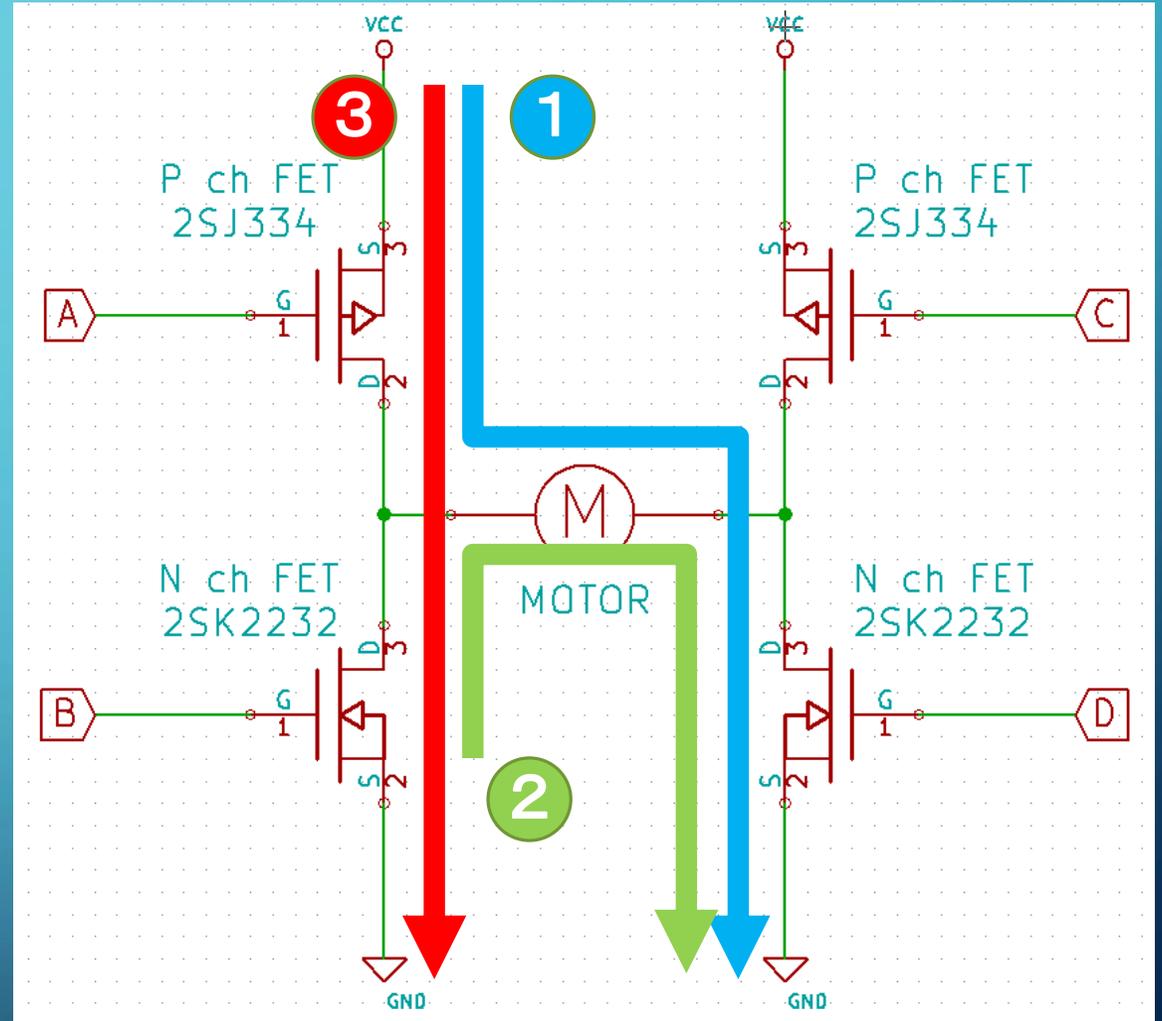
※クリックするとそのページにジャンプします。

Hブリッジ回路

- 双方向回転回路

A	B	C	D	モーター	回路図
0	0	1	1	正転	①
1	1	0	0	逆転	
0	0	0	0	ブレーキ	②
1	1	1	1	ブレーキ	
0	1	X	X	禁止(貫通電流)	③
X	X	0	1	禁止(貫通電流)	
その他				フリー	

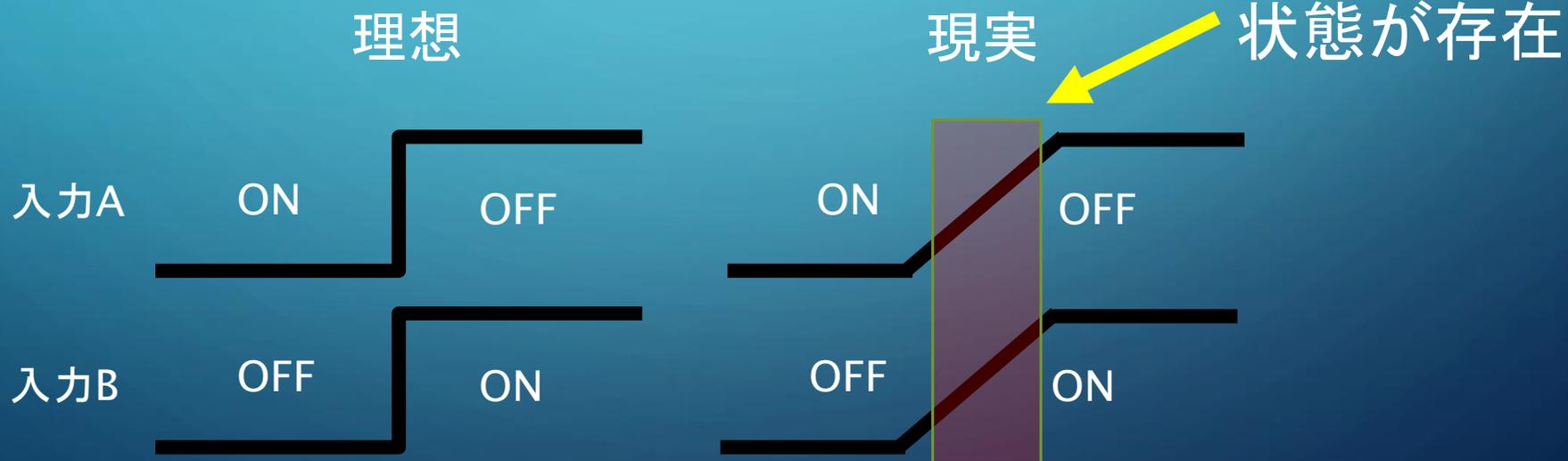
- A,B,C,Dに電位を与える.
- 禁止のように操作すると, 燃える.



Hブリッジの注意①

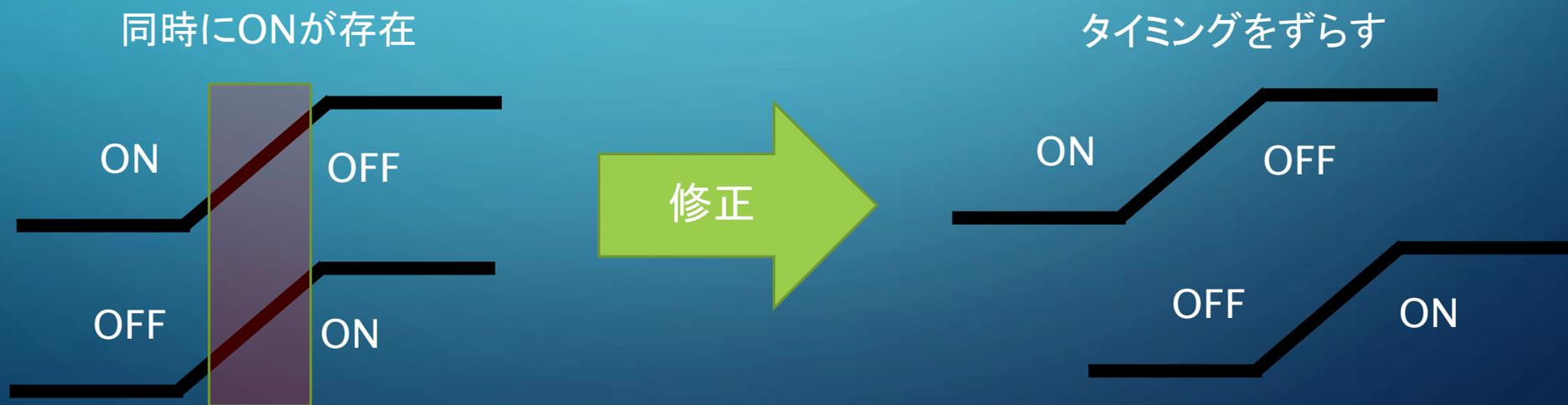
- 絶対に貫通電流を流してはいけない。一瞬で壊れる！
- ロジック回路で0→1や1→0になるのに少し時間がかかる。

→ 貫通電流が流れる可能性



Hブリッジの注意①の対策

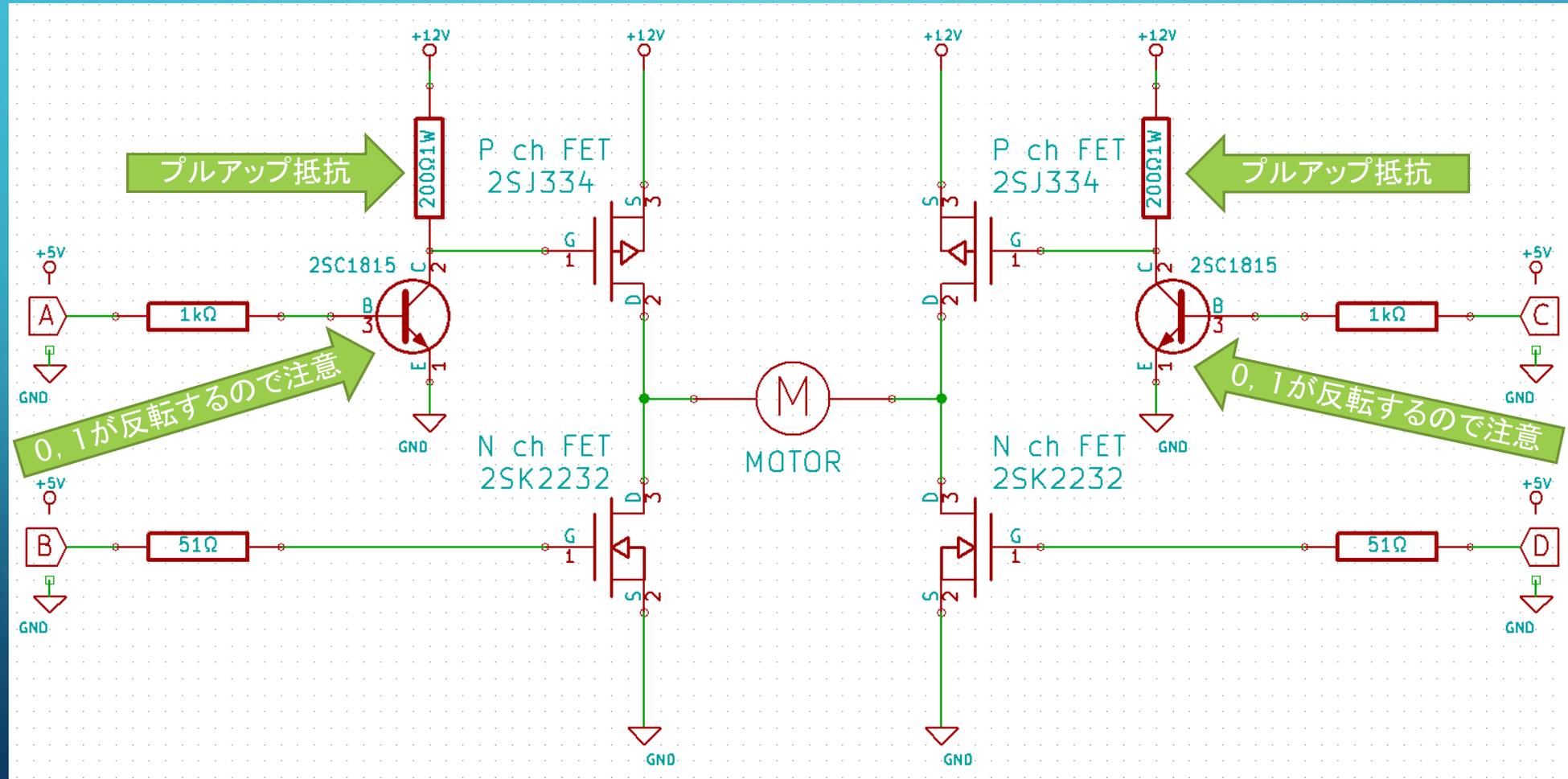
- デッドタイム(両方OFFの時間)を生成する
 - ソフトウェアでデッドタイムを生成
 - ハードウェアでデッドタイムを生成



Hブリッジの注意②

- モーターが12Vで、入力ロジックが5Vの場合、5Vをつないでも12Vの半分以下なので、LOWと認識されてしまう。→誤作動
- モーターの電圧とロジックの電圧が違くと、スイッチングができない。
→部品を少し足して回路を変える。

Hブリッジの注意②の対策



回路修正後のモーター駆動表

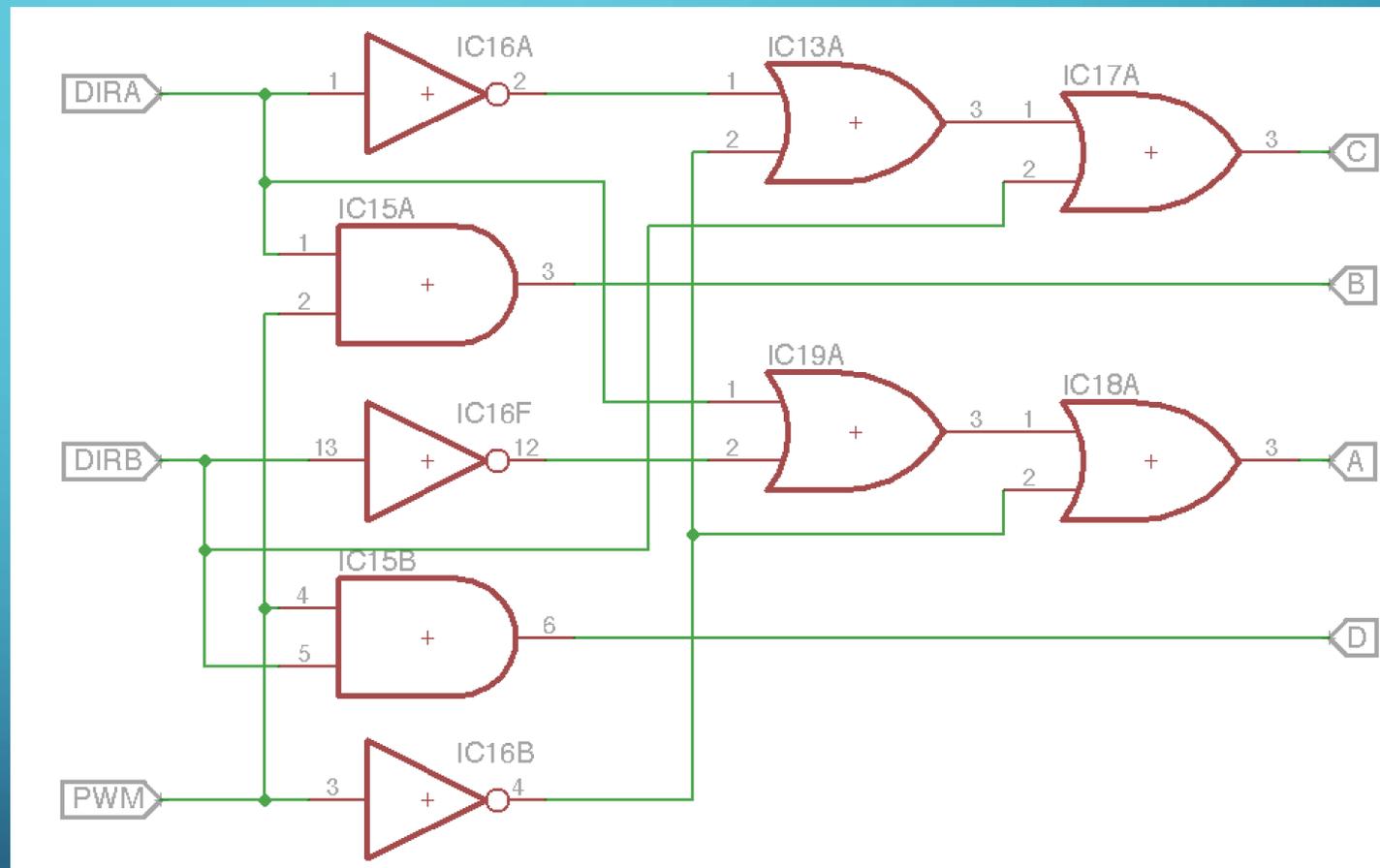
- 実装するときはしっかり0と1を確認しよう.
- マイコンの電源が入っていない状態で禁止パターンにならないことも確認しておくこと.

A	B	C	D	モーター
1	0	0	1	正転
0	1	1	0	逆転
1	0	1	0	ブレーキ
0	1	0	1	ブレーキ
1	1	X	X	禁止
X	X	1	1	禁止
その他				フリー

モーター駆動ロジック回路

- Hブリッジの駆動表をロジック回路で表す.
- これは先ほどの電圧修正**前**のHブリッジ用であることに注意する. (一般のHブリッジ)
- PWM端子にはパルス波を送る

DRI A	DRI B	モーター
0	0	フリー
0	1	正転
1	0	逆転
1	1	ブレーキ



第3回 目次

1. [宿題の解説](#)
2. [デジタル回路入門](#)
3. [論理回路](#)
4. [モーター](#)
5. [PWM制御](#)
6. [Hブリッジ回路](#)
7. [モータードライバ](#)
8. [シリアル通信](#)
9. [宿題](#)

目次に戻る

1. モータードライバICとは
2. モータードライバIC TA7291P
3. TA7291を使う上で
4. サーボモーター

※クリックするとそのページにジャンプします。

モータードライバICとは

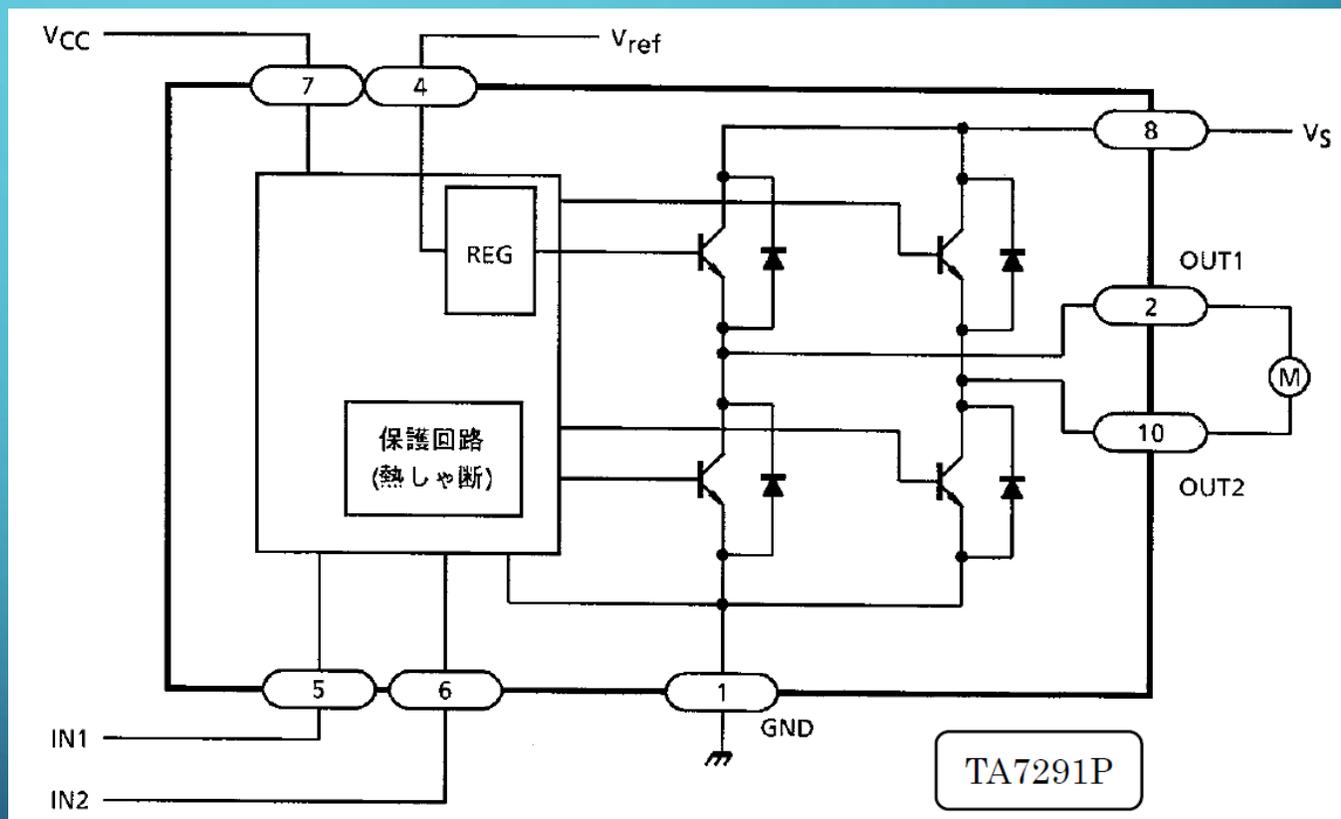
- Hブリッジ回路や駆動ロジック回路を組むのは結構大変.
→ それらの回路をまとめたICが存在する
→ それがモータードライバIC
- 有名なIC: TA7291P (2個で300円)
 - 定格電流: 1A (ピーク2A) 定格を守ろう.
 - 1つのICにHブリッジ, 駆動ロジック, 保護回路が入っている.



モータードライバIC; TA7291P



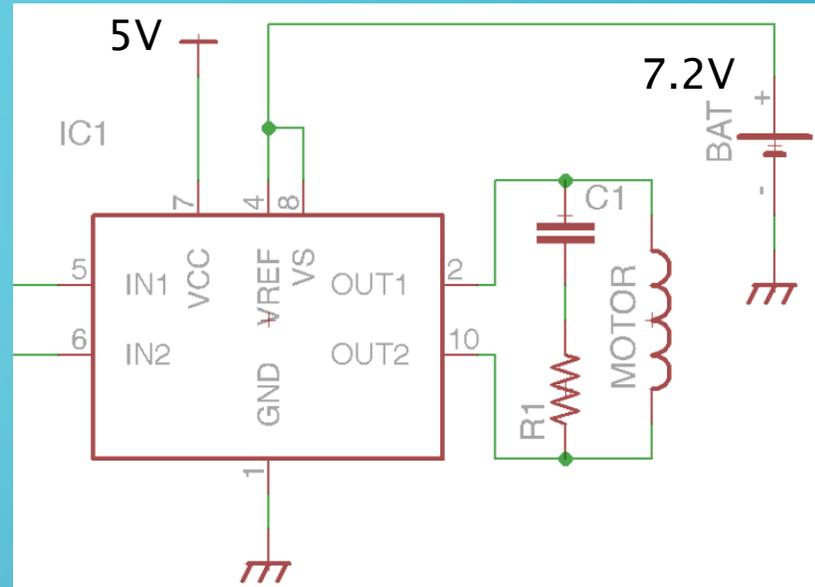
ピン	ピン名	役割
7	Vcc	ロジック電源+
8	Vs	モーター電源+
4	Vref	Vsにプルアップ
1	GND	共通グラウンド
5	IN1	ロジック入力1
6	IN2	ロジック入力2
2	OUT1	モーター出力1
10	OUT2	モーター出力2
3,9	NC	何もつながない



TA7291P内部構造

TA7291P

- 入力ロジックと出力の関係
- ∞ はハイインピーダンス → フリー回転



入 力		出 力		モード
IN1	IN2	OUT1	OUT2	
0	0	∞	∞	ストップ
1	0	H	L	CW / CCW
0	1	L	H	CCW / CW
1	1	L	L	ブレーキ

TA7291Pを使う上で

- TA7291Pは定格1Aである。
 - 普通の小型DCモーターをまわすのに適切な電流
 - 大きなロボットの動力源には、電力が足りない。
- 電流の大きなモータードライバもあるが、値段が高い！
 - やっぱり、Hブリッジを自分で作ろう。



小型DCモーター

サーボモーター

- 角度を自由に制御できるモーター。
 - ロボットの腕, 舵に使う
- 回転し続けて使うわけではなく, ある角度にとめながら使うことが多い。
- +/-90度くらいしか回らないものが多い
- 周期20msくらいのパルス波を送ることで角度を制御 → Duty比が角度に



電源+

パルス波入力

電源-

第3回 目次

1. [宿題の解説](#)
2. [デジタル回路入門](#)
3. [論理回路](#)
4. [モーター](#)
5. [PWM制御](#)
6. [Hブリッジ回路](#)
7. [モータードライバ](#)
8. [シリアル通信](#)
9. [宿題](#)

目次に戻る

1. シリアル通信とは
2. シリアル通信の種類
3. シリアル通信のやり方

※クリックするとそのページにジャンプします。

シリアル通信とは

- 2つのマイコンなどの間でデータを送受信すること.
- シリアル通信とは、通信の総称で、シリアル通信の中にいろいろな種類がある.
 - 例: UART, I2C(I²C), SPI などがある.
 - 上のようにちゃんとした名前がなくても、マイコンの通信ならばシリアル通信と呼べる.
- 実際の使い道
 - マイコンからパソコンに逐次データを送って、正常に動作しているか確認する.
 - センサICから読み取ったデータを、マイコンに送る.
 - マイコン親機が、マイコン子機を操る.

シリアル通信の種類

- UART(ユーアート)
 - 最もよく使われているシリアル通信. 主にパソコンとマイコンの間の通信に使われる.
- SPI(エスピーアイ)
 - 高速の通信が可能. マイコンとICや, マイコン同士で使われる.
- I2C, I²C(アイツーシー, アイスケアードシー)
 - マイコンとICの間で使われる. 通信線の数が少ないのが特長.

シリアル通信のやり方

- 一般的なシリアル通信は、マイコンの機能として存在する。
 - 自分でパルス波を作って送信するのではなく、送りたいデータを送信するプログラムを書けば、**自動で送信**してくれる。
- そんなに難しく考える必要はない。

第3回 目次

1. [宿題の解説](#)
2. [デジタル回路入門](#)
3. [論理回路](#)
4. [モーター](#)
5. [PWM制御](#)
6. [Hブリッジ回路](#)
7. [モータードライバ](#)
8. [シリアル通信](#)
9. [宿題](#)

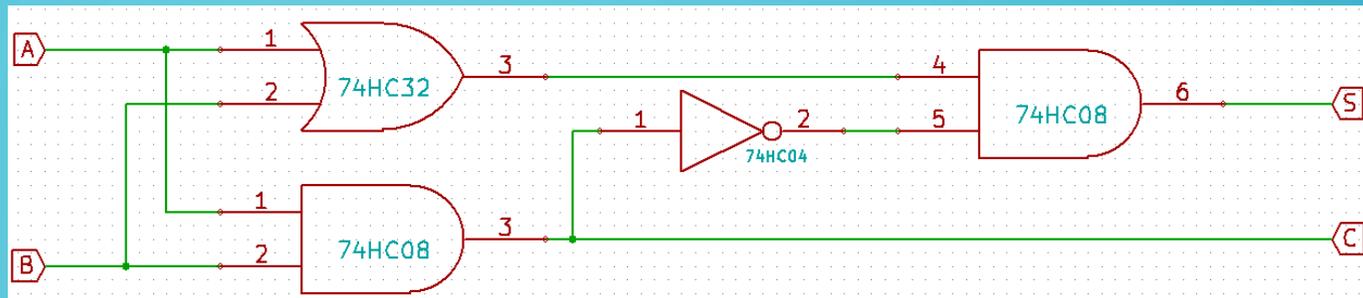
1. 加算器を作ろう

目次に戻る

※クリックするとそのページにジャンプします。

宿題

半加算器→



- 1 bit半加算回路を作ってみよう. 入力はタクトスイッチ, 出力はLED.
- 提出などはありません. 自分で確認してみてくださいね.
- ブレッドボードを使おう. 入力のプルダウン抵抗を忘れずに.
- 余裕のある人は, 1 bit全加算器を作ってみよう.
 - 全加算器と半加算器の違いを考えよう.

↓ 全加算器

52

