面白プログラムの発表

1126100
前川大輝

発表の流れの説明

1．今回の学習目的
2．プログラム概要説明
3．実機によるデモンストレーション
4．OutPutCompareモジュールの概要
5．制御に必要な回路知識
6．プログラムソース解説
7．質疑応答

１．今回の学習目的

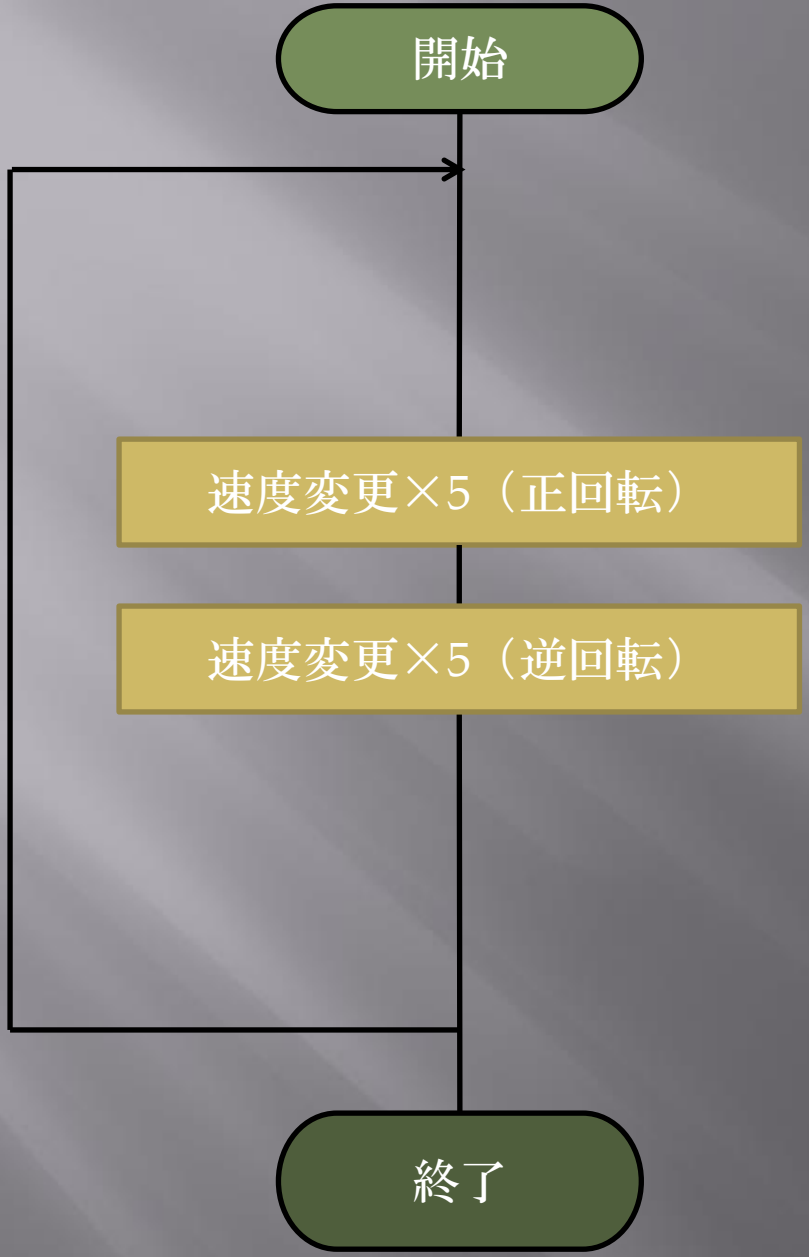
DCモータの速度制御ができるようになること

2.プログラム概要説明

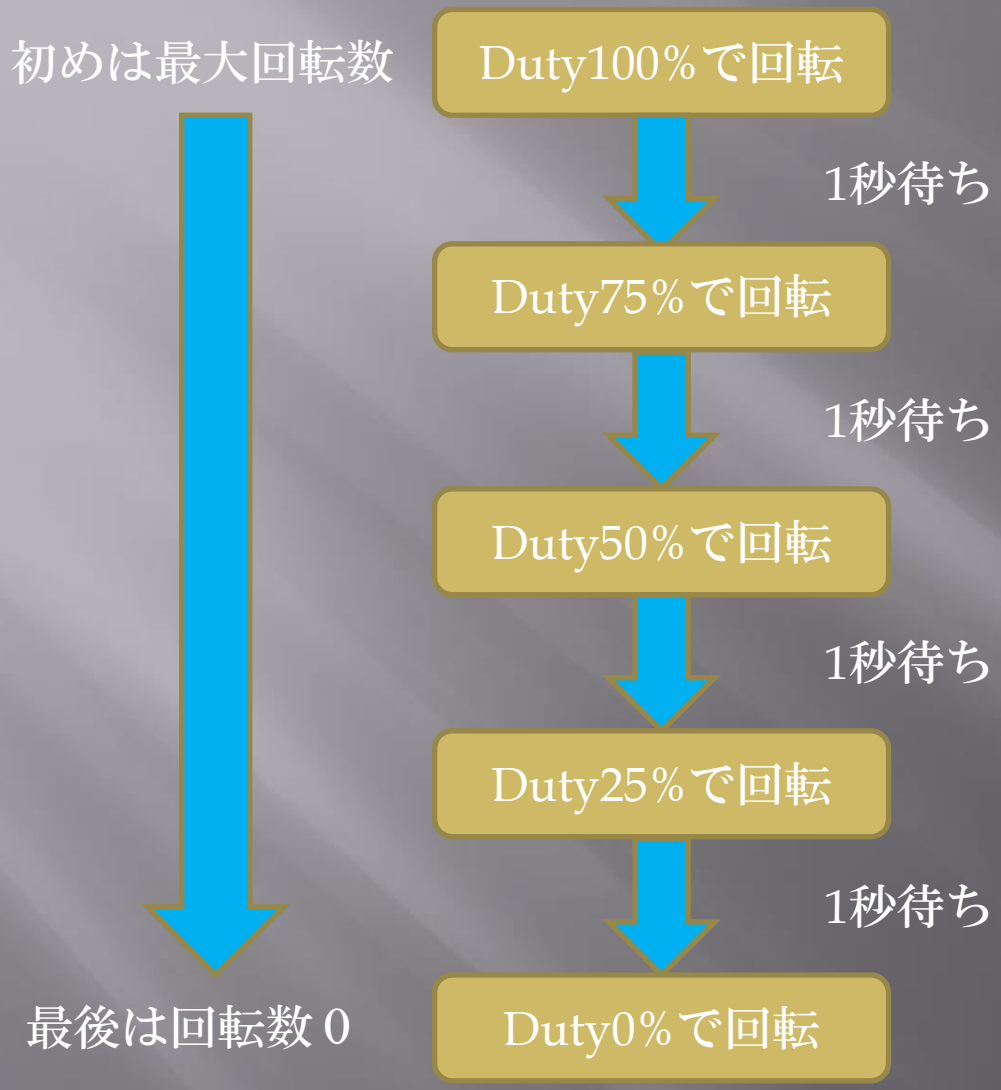
OutPutCompareモジュールを使用し、モータの速度制御を行うプログラム

速度変化はデューティー比を100%,75%,50%,25%,0%と順番に変化させることにより実現する

また、正回転と逆回転切り替える


開始
速度変更×5（正回転）
速度変更×5（逆回転）
終了

速度変更×５　の説明



ON・OFF
切り替え周期は
変更していません

３．実機によるデモンストレーション

それではさっそくデモンストレーションします。

4．OutPutCompareモジュールの概要

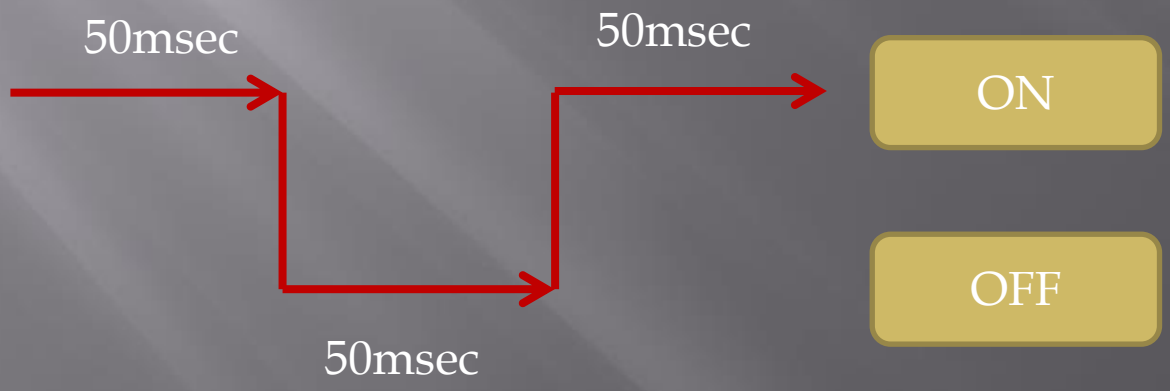
OutPutCompareモジュールを簡単に言い表すなら
高速でON・OFFを繰り返すことができる機能である

一定時間内におけるON・OFFの比率を調整することにより速度制御することができる（これがデューティー比）

OutPutCompareの動作方法

ON・OFFの周期はTimer2またはTimer3に依存します

Timerによるカウントを100msecに設定し、
デューティー比を50パーセントの場合



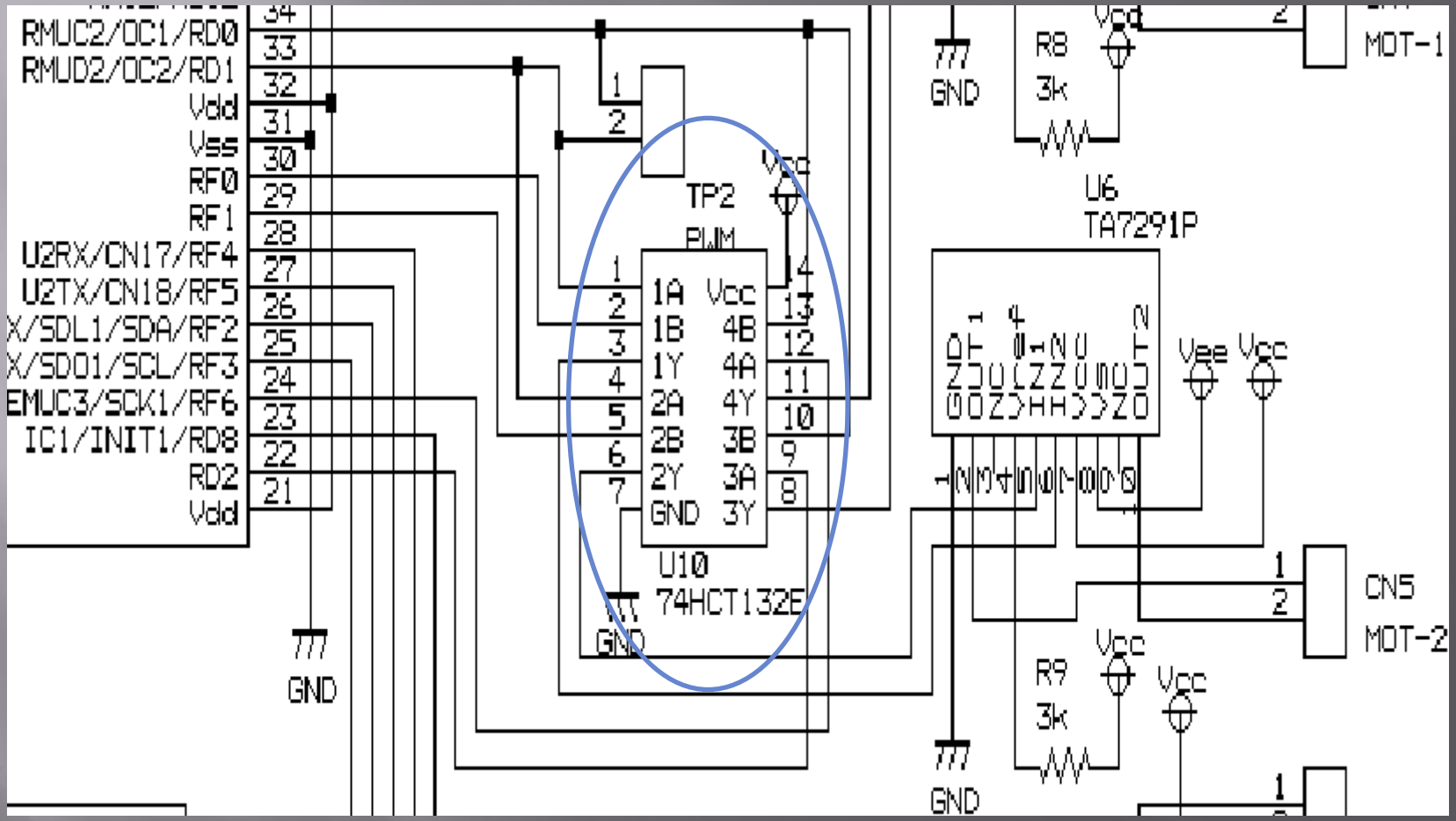
例としてカウント周期はそのままでデューティー比だけを変更する

■デューティ比２５パーセントの場合

■デューティ比100パーセントの場合

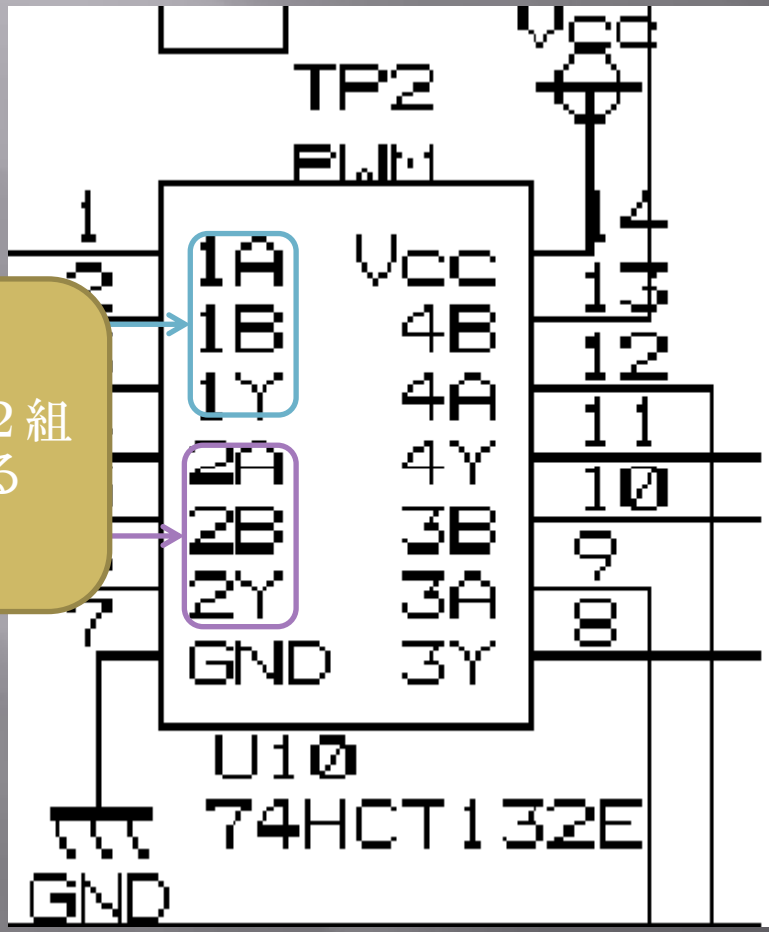
デューティ比を変化させることで速度を
調整できることがわかりました

5. 制御に必要な回路知識



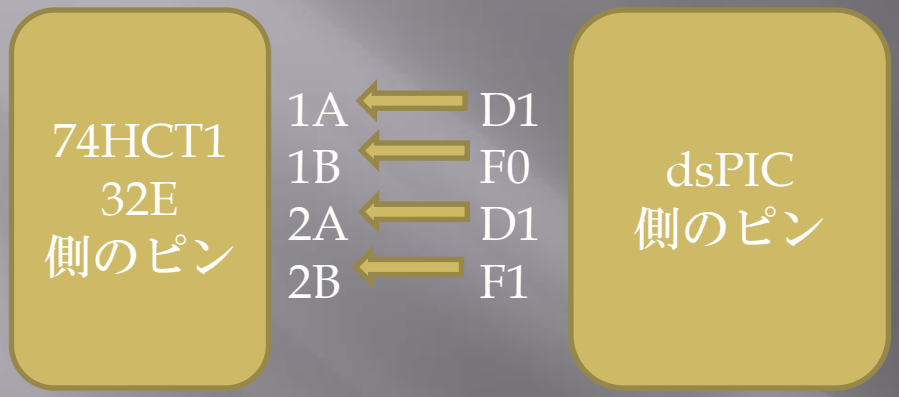
74HCT132E
から解説します

今回はこの２組
を使用する



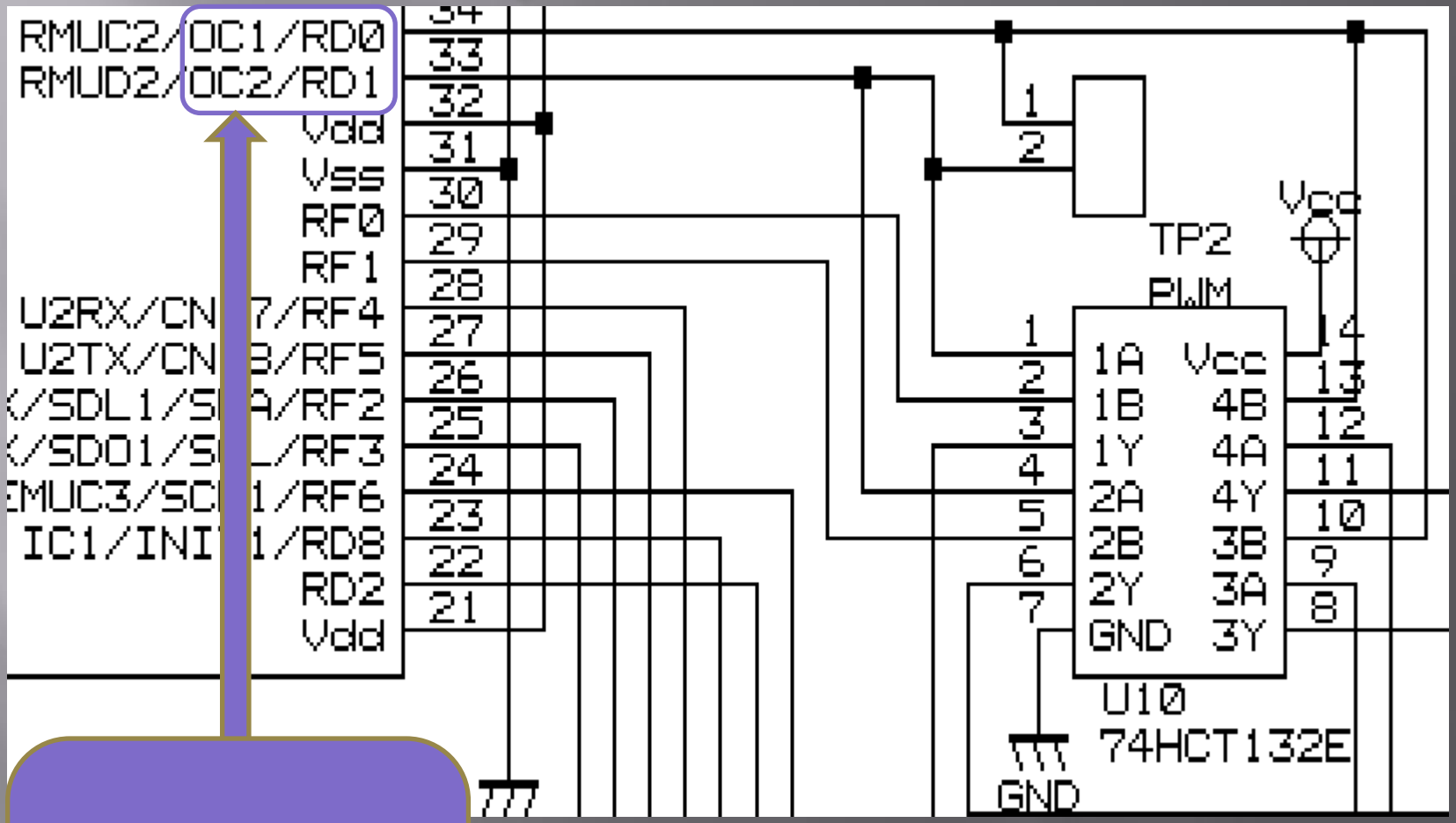
74HCT132E
の回路内は
A:入力
B:入力
Y:出力
の
NAND回路に
なっている

74HCT132EとPICとのつながり



※１Aと２AがPICのD1ピンを共有している点がポイント

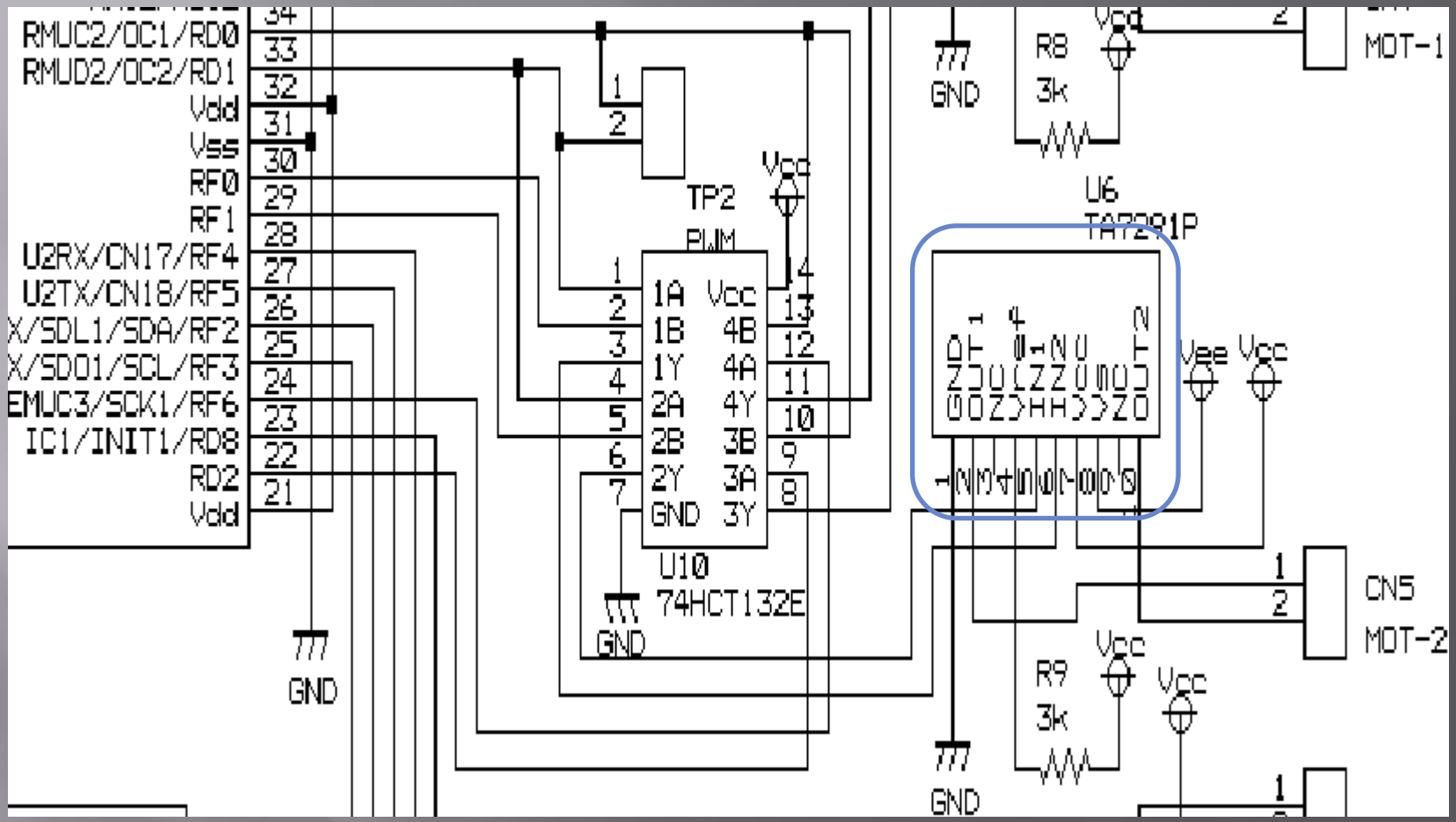
OutPutCompareモジュールの回路



RD1は
OutputCompare2
を使用した際のピン
と同じだった

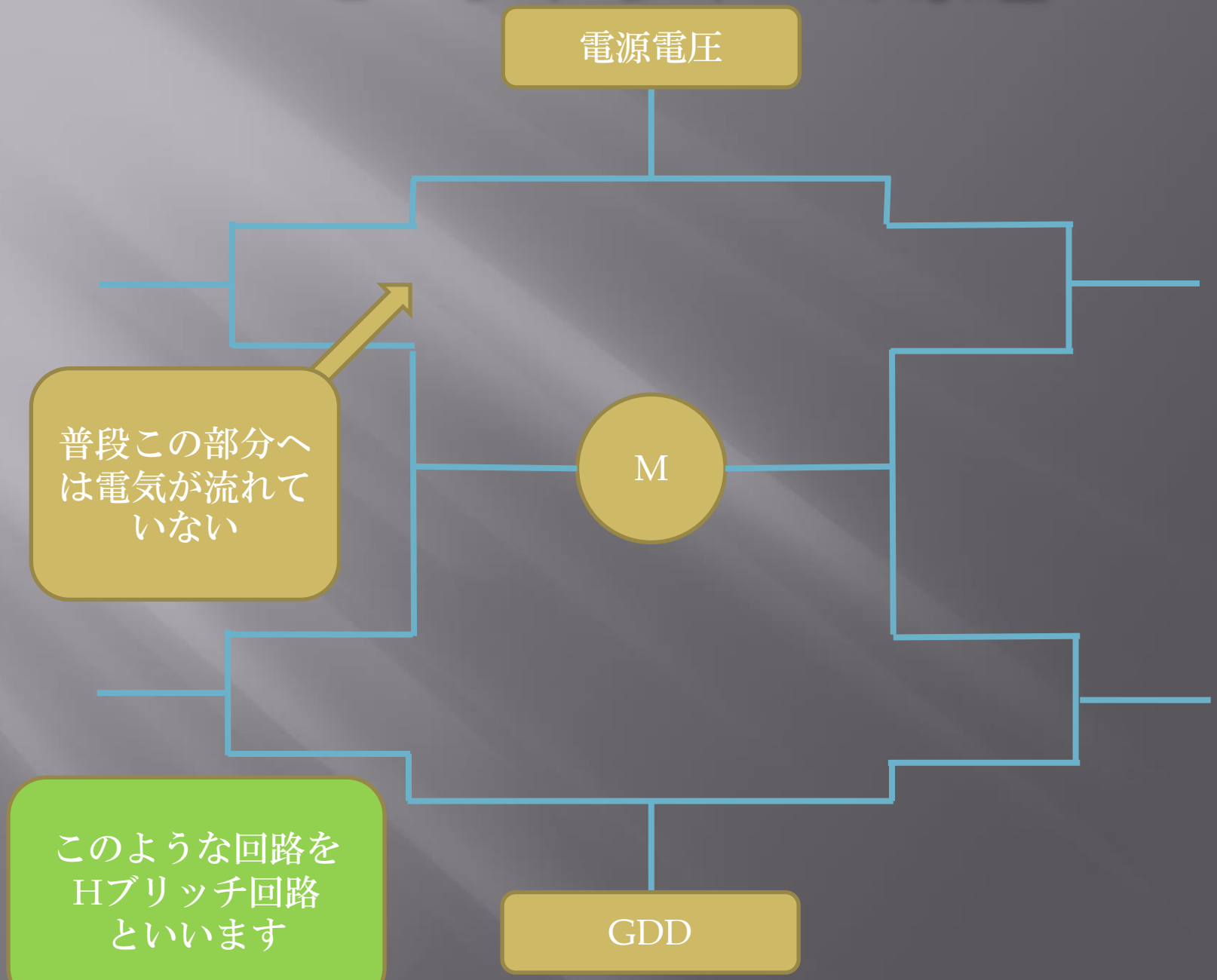
つまりOutputCompare2を使用したとき
1Aと２Aが同時に
1→0→1→0
を一定周期で繰り返すということ

モータドライバ

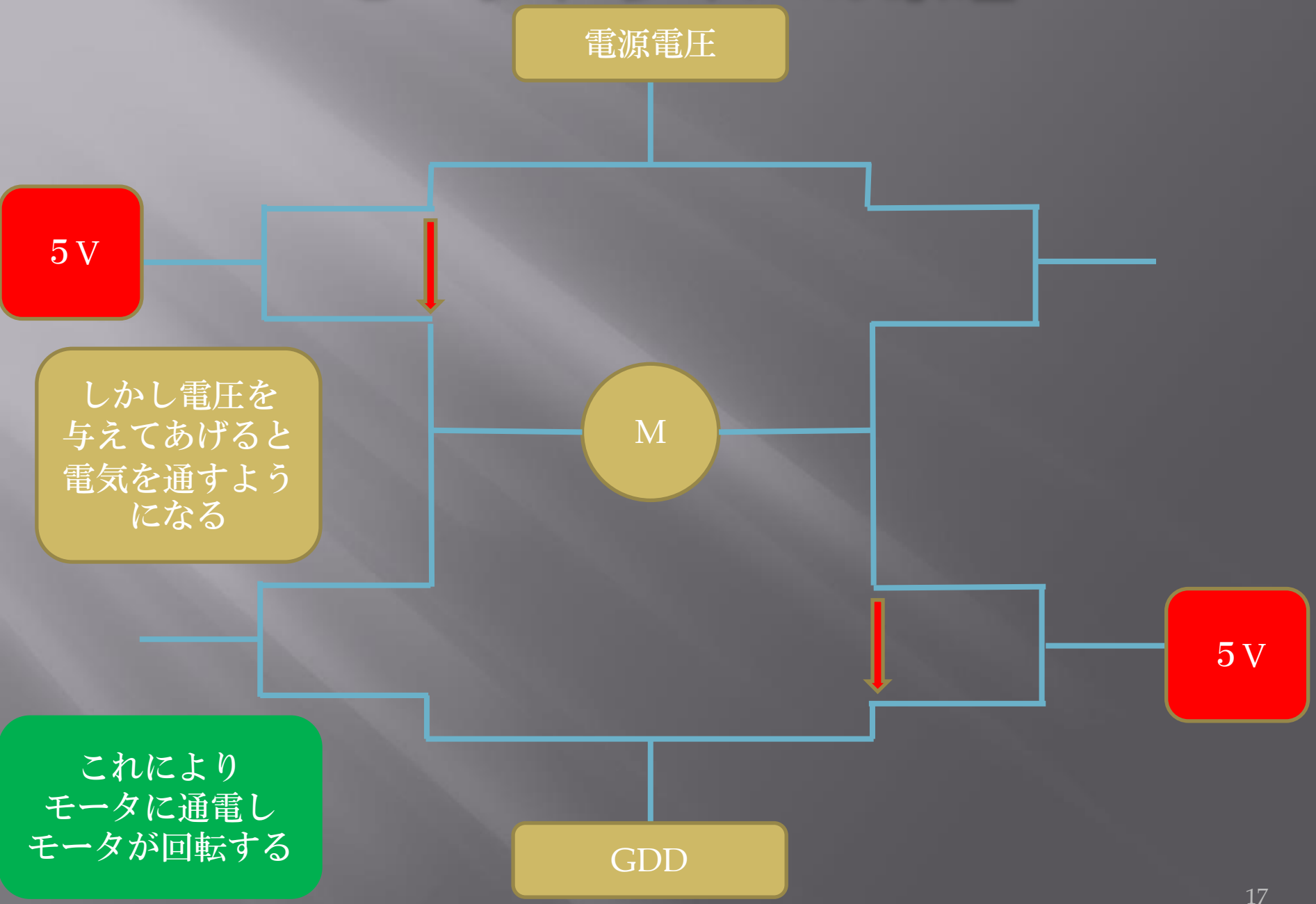


次はモータドライバを
解説します

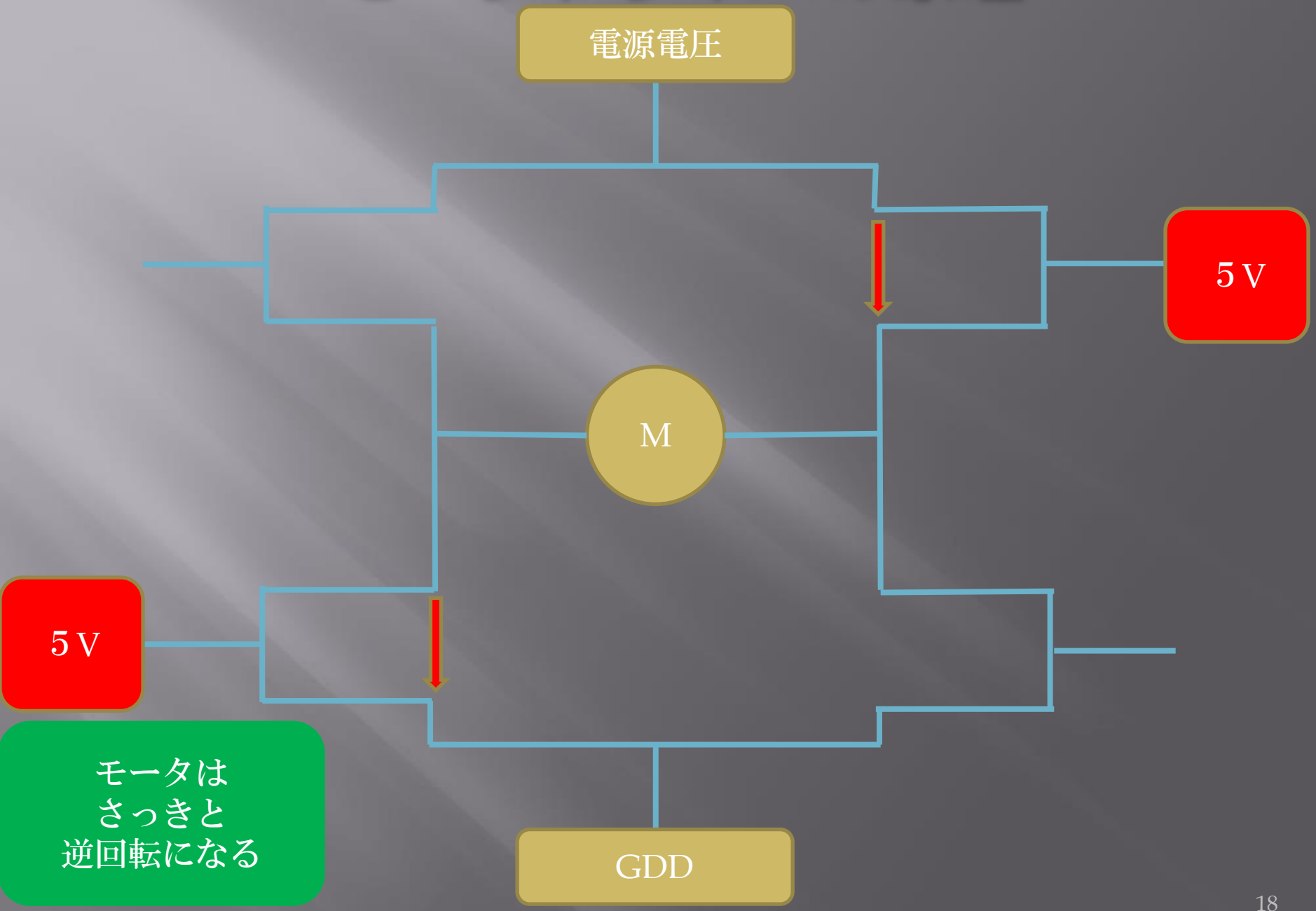
モータドライバの原理



モータドライバの原理

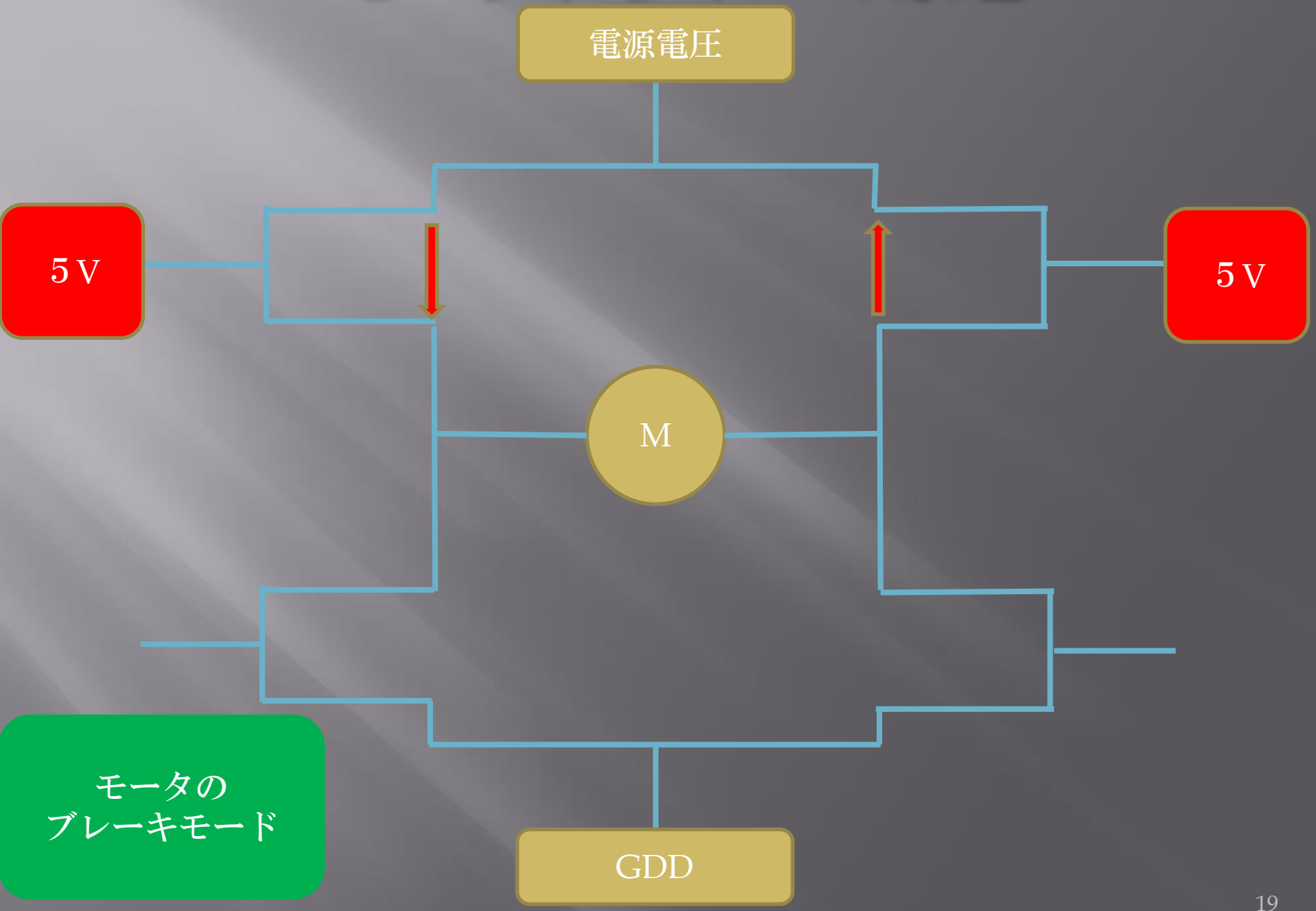

電源電圧
5 V
しかし電圧を
与えてあげると
電気を通すよう
になる
M
5 V
これにより
モータに通電し
モータが回転する
GDD

モータドライバの原理



モータは
さっきと
逆回転になる

モータドライバの原理



しかし、すべてのモータドライバが同一とは限りません

例）種類によっては、ブレーキモードがないものも存在しますので注意すること。

今回使用するTA7291Pはどうなっているのか、仕様書を見ていきましょう。

STEP1　端子の確認

端子説明

端子記号	端子番号			端　子　説　明
	P	S/SG	F/FG	
V_{CC}	7	2	11	ロジック側電源端子
V_S	8	6	15	出力側電源端子
V_{ref}	4	8	5	制御電源端子
GND	1	5	1	GND
IN1	5	9	7	入力端子
IN2	6	1	9	入力端子
OUT1	2	7	4	出力端子
OUT2	10	3	13	出力端子

P タイプ　：　③⑨ピンは NC 端子
S/SG タイプ：　④ピンは NC 端子
F/FG タイプ：　②③⑥⑧⑩⑫⑭⑯ピンは NC 端子
なお F タイプの FIN は、GND にショートすることを推奨します。

STEP2　ファンクションの確認

ファンクション

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

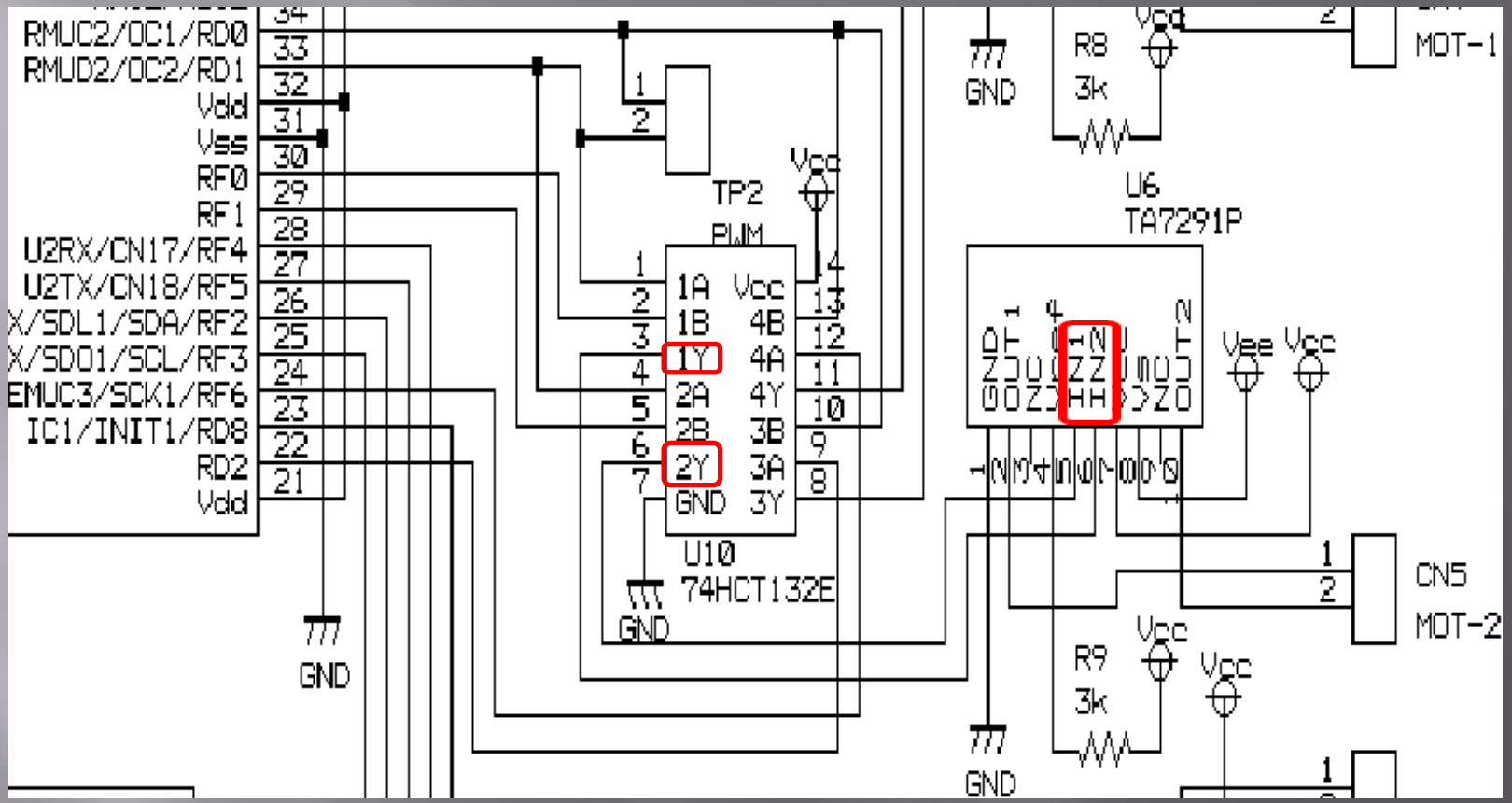
∞:　ハイインピーダンス

注:　入力は" H" アクティブ

ブレーキモード
IN1　5 V
IN2　5 V

正回転
IN 1　5 V
IN 2　0 V

逆回転
IN 1　0 V
IN 2　5 V



モータドライバ

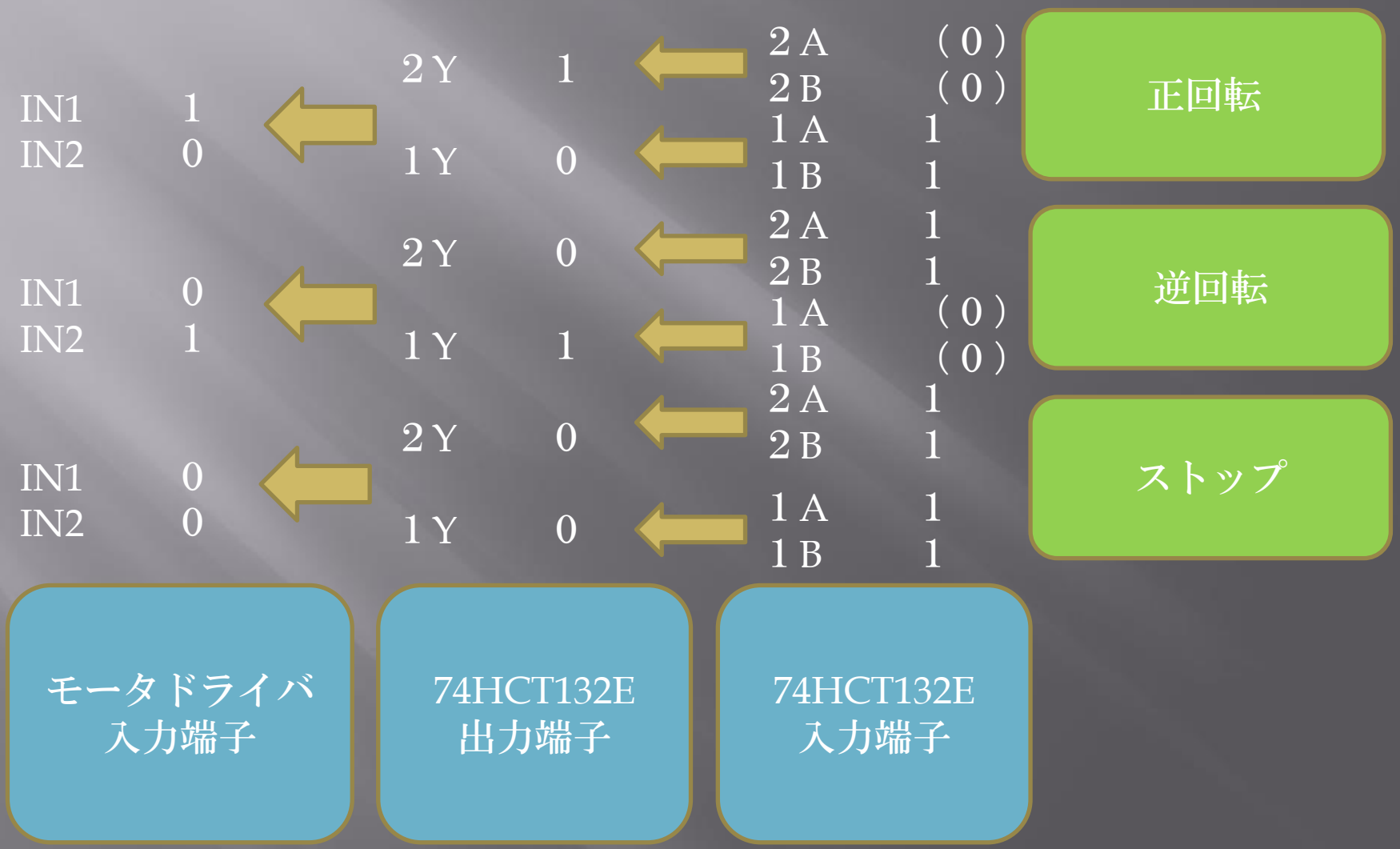
IN 1 ⇐ 2 Y

IN 2 ⇐ 1 Y

74HCT132E

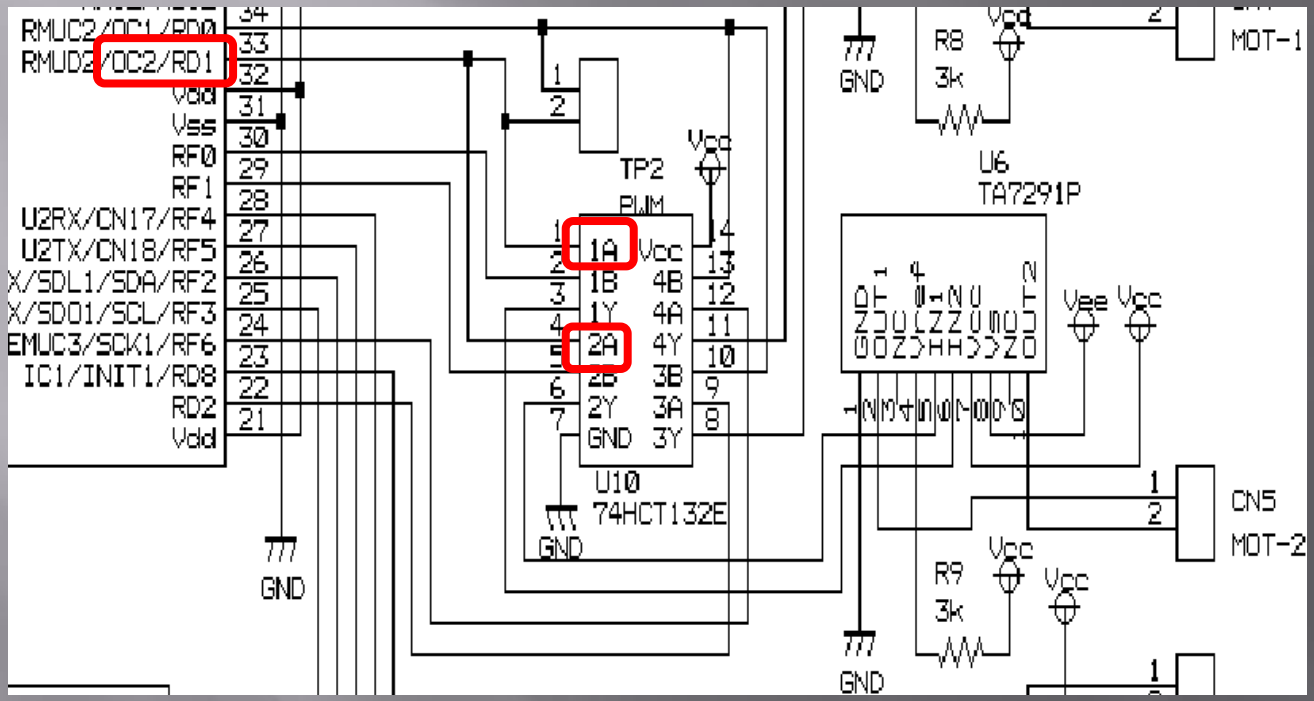
74HCT132Eからモータドライバまでの関係性

※　（）はそれ以外の組み合わせが存在することを示している



dsPICから74HCT132Eの関係性

前述したとおり、OutPutCompare２を使用した際
１Ａと２Ａが1→０を繰り返す



よってモータの回転を制御するために
調節すればいいのは１Ｂと２Ｂ

以上で回路の
説明は終了

６．プログラムソース解説

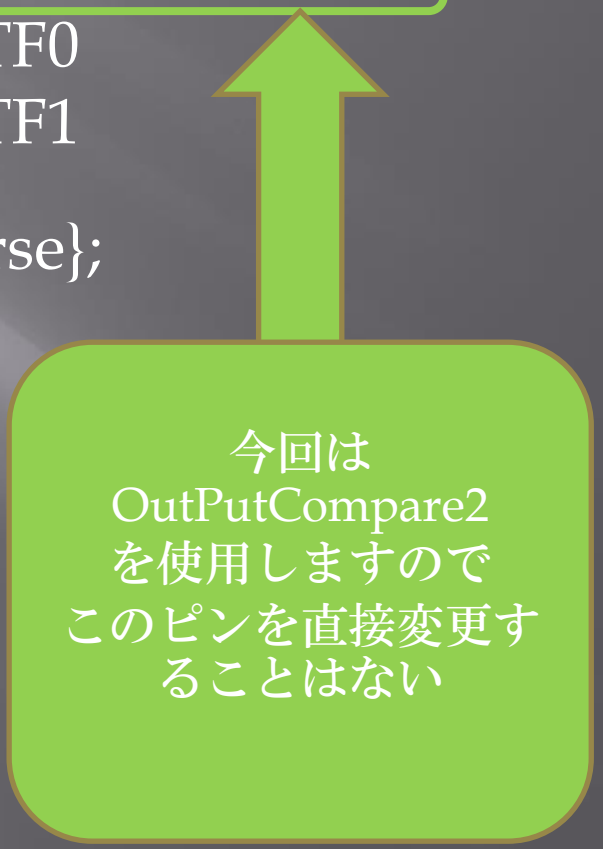
必要なヘッダーファイルをinclude

```
#include "p30F3014.h"
#include "timer.h"
#include "outcompare.h"
#include "lcd.h"
```

必要な定数を定義

```
#define PWM_1A_AND_2A LATDbits.LATD1
#define PWM_1B LATFbits.LATF0
#define PWM_2B LATFbits.LATF1

enum MotionType{normal,reverse};
```



コンフィギュレーション設定

_FOSC(CSW_FSCM_OFF & XT_PLL8);

8×10MHz(外部発振機)=
80MHz(クロック周波数)

CSW（クロック切り替え）FSCM(クロックのエラー検出)ともにOFF
（※クロック停止の際の処理→監視もしないし、内部の別クロック源に切り替えもしない）

_FWDT(WDT_OFF);

ウォッチドッグタイマ無効
(※特にシステムの監視を行わない場合は、これをOFFとする)

_FBORPOR(PBOR_ON & BORV_20 & PWRT_64 & MCLR_EN);

電源ON直後に動作リセット
電源OFF直後に動作停止
電源OFFを検知する電圧→2.0V
電源ON直後のリセットパルス幅→64msec
MCLRピン→有効

_FGS(CODE_PROT_OFF);

コードプロテクト→読み出し・書き込みともにOFF

main文内の説明

```
short MotionType;
int i;
float Duty_set[4] = {1,0.75,0.5,0.25,0};

TRISD = 0x0000;
TRISF = 0x0000;

OpenOC2(OC_IDLE_CON & OC_TIMER2_SRC &
        OC_PWM_FAULT_PIN_DISABLE   ,0   ,0);
```

デューティ比
100%
75%
50%
25%
0%
をそれぞれ準備

アイドルループ内
動作継続

タイマー2を選択

出力モードを
PWMに設定

OCxRSレジスタ
に設定する値

OCxRレジスタ
に設定する値

OpenOC関数の詳しい解説

OutPutCompareモジュールはタイマ２かタイマ3のカウンタと、
OCxRレジスタかOCxRSレジスタの値を常時比較していて、一致したとき、何らかの出力をOCx端子にするというものです。

１）今回はOCxRレジスタ、OCxRSレジスタともに0に設定しました

２）そしてカウンタはタイマー2を使用します。

３）PWMモードを指定しているので、OCxRレジスタが比較対象のレジスタとなり、一致によりOCxレジスタはLowに制御されます。
さらにタイマが0クリアされるとき、Ocx出力がHighに制御される。

OCx出力とは
OutputCompare出力の略
つまり今回ではdsPICの
OC2端子から出力するものを指します

今回はPWMモードで使用するため
OCxRSレジスタの値は
意味を持ちません

main文内の説明（２）

```
//4/80Mhz×64×3125=10msec
OpenTimer2(T2_ON & T2_GATE_OFF & T2_PS_1_64 &
                                    T2_SOURCE_INT,3125-1);
```

これによりOC２出力周期（ON・OFF周期）が0.01秒に決定した

```
MotionType = normal;        //正回転にセット
PWM_1B = 1;
PWM_2B = 0;
```

OC2出力が１の時

1A=1
2A=1

NAND
1A=1
1B=1
→ 1Y=0 → IN2=0

NAND
2A=1
2B=0
→ 2Y=1 → IN1=1

モータは
正回転する

一応ファンクションを参照

ファンクション

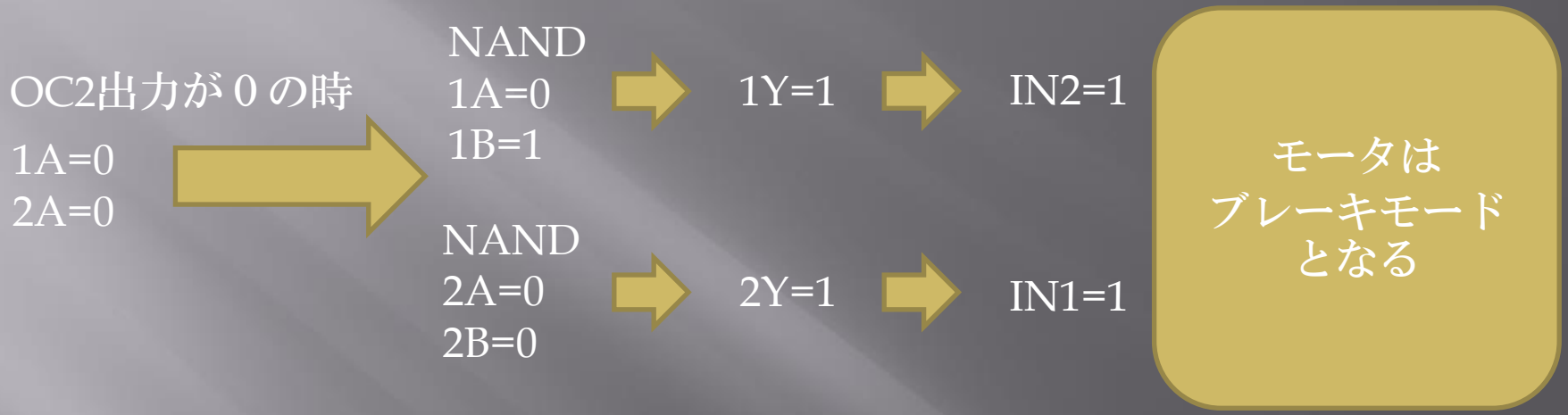
入　力		出　力		モード
IN1	IN2	OUT1	OUT2	
0	0	∞	∞	ストップ
1	0	H	L	CW / CCW
0	1	L	H	CCW / CW
1	1	L	L	ブレーキ

∞:　ハイインピーダンス

注:　入力は" H" アクティブ

正回転している

ではOC2出力が0の時モータは、ちゃんと停止しているのか
見ていきましょう



一応ファンクションを参照

ファンクション

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

∞:　ハイインピーダンス

注:　入力は" H" アクティブ

ブレーキモードになっている

main文内の説明（３）

```
while(1){
        for(i=0;i<5;i++){
                SetDCOC2PWM(3125*Duty_set[i]);          //デューティ比の設定
                machi_msec(1000);
        }

        if(MotionType == normal){
                ///逆回転にセット
                PWM_1B = 0;
                PWM_2B = 1;
                MotionType = reverse;
        }else{                                          //つまり逆回転の時
                ///正回転にセット
                PWM_1B = 1;
                PWM_2B = 0;
                MotionType = normal;
        }
}
```

デューティ比を
100%~0%まで
段階的に変更する

本当に逆回転になってる？

PWM_1B = 0;
PWM_2B = 1;

が逆回転となる仕組みを解説します

OC2出力が1の時

1A=1
2A=1

NAND
1A=1
1B=0

1Y=1

IN2=1

NAND
2A=1
2B=1

2Y=0

IN1=0

モータは
逆回転する

OC2出力が0の時

1A=0
2A=0

NAND
1A=0
1B=0

1Y=1

IN2=1

NAND
2A=0
2B=1

2Y=1

IN1=1

モータは
ブレーキモード
となる

一応ファンクションを参照

ファンクション

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

∞:　ハイインピーダンス

注:　入力は" H" アクティブ

逆回転している

以上によりプログラムの解説を終わります

7. 質疑応答

質問はなんでもいいです。

回路図に間違いを見つけたので報告します！

以上にて発表を終了させていただきます。

ご清聴ありがとうございました。