

第6章

車輪型ロボットの製作

材料と道具／モータ駆動ICによる移動制御／センサを用いた障害物回避

車輪型ロボットは代表的なロボットで、センサと組み合わせて、様々な研究やロボット・コンテストで使われています。

この章では、車輪型ロボットの製作方法を説明します。車輪型ロボットをゼロから作ろうとすると手間がかかるので、市販の車輪駆動キット(タミヤ模型のブルドーザ工作基本セット)を活用することにしました。これにマイコンH8Tinyによる制御部と、センサを用いた障害物回避部を追加します(図6.1)。これらの例を通じて、センサとモータの組み合わせかたを学びます。

6.1 材料と道具

必要な材料(表6.1)は、秋葉原の「千石電商」と「秋月電子」でほとんど購入できます。どちらもイン



(a) 車輪型ロボット(改造後)



(b) ブルドーザ工作基本セット(改造前)

図6.1 車輪型ロボットと工作基本セット

参考：購入店

- ・秋月電子(<http://akizukidenshi.com/>)
- ・千石電商(<http://www.sengoku.co.jp/>)

表6.1 必要な材料の購入リスト

製品	個数	単価	金額	購入店
TAMIYA 楽しい工作シリーズ No.104 2チャンネル・リモコン・ブルドーザ工作基本セット	1	2,000円	2,000円	千石電商
モータ・ドライブIC TA7279P	1	450円	450円	千石電商
赤外線距離センサ GP2D12	2	720円	1,440円	千石電商
メス・ソケット (Dサブ9ピン)	1	50円	50円	千石電商
電池ソケット	2	20円	40円	千石電商
乾電池 (9V)	2	200円	400円	千石電商
マイコン H8/3664	1	1,600円	1,600円	秋月電子
その他 (LED, ケーブル, ピン・ソケット)	—	—	—	千石電商
合計金額			約6,000円	

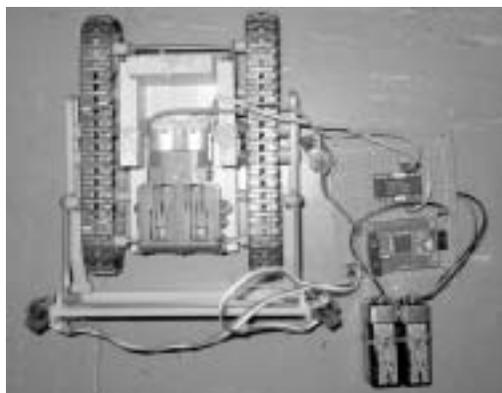


図6.2 2個のDCモータとその接続

ターネット通販も行っています。

6.2 モータ・ドライブICによる移動制御

車輪型ロボットのベースとして、TAMIYA 楽しい工作シリーズ No.104「2チャンネル・リモコン・ブルドーザ工作基本セット」を使います。このセットは、モータにつなぐ4線をモータ・ドライブIC TA7279Pに接続すると、マイコン・リモート・コントロールができますが、ここではこれを使いません。図6.2のように二つのDC制御ができるように改造しました。

全体の回路図および配線を図6.3、表6.2、表6.3に示します。H8Tinyからモータ・ドライブIC(図6.4)には4線が接続されており、各モータの正転・逆転・強弱の制御ができます(表6.4)。また、H8は2個の赤外線距離センサと接続されており、左右斜め前方にある障害物との距離を検出できます。

前進駆動のプログラムをリスト6.1に示します。表6.4に示す設定により各モータの回転方向を決

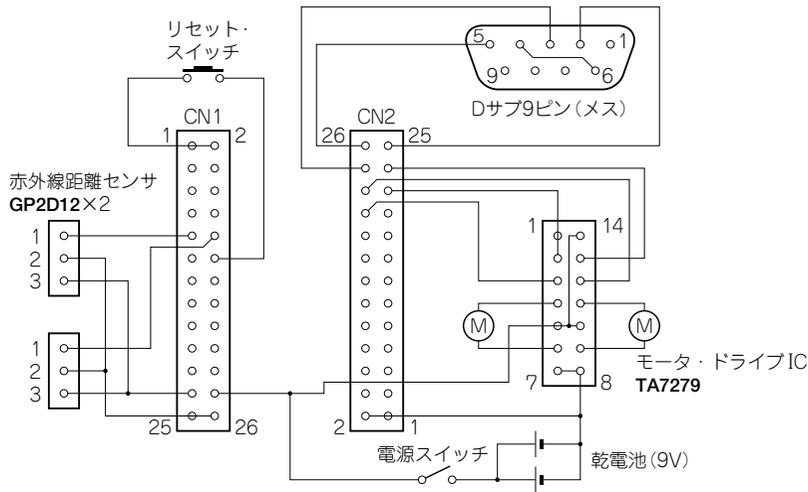


図6.3 車輪型ロボットの回路図

表6.2 H8/3664の配線

ピン配置	H8/3664 機能	内容
CN1-10	AD0	赤外線距離センサ 0ch
CN1-9	AD1	赤外線距離センサ 1ch
CN2-21	P1-5	TA7279-1, 2
CN2-23	P1-7	TA7279-13
CN2-20	P1-4	TA7279-3
CN2-22	P1-6	TA7279-12
CN2-24	RXD	シリアル通信 Dサブ9ピン -3
CN2-25	TXD	シリアル通信 Dサブ9ピン -2
CN2-26	GND	シリアル通信 Dサブ9ピン -5
CN1-6	RESET	リセット・ボタン
CN1-24	電源	乾電池 +9V (2個並列)

表6.3 TA7279の配線

ピン配置	TA7279 機能	内容
1, 2	IN2-A	H8/3664 - CN2-21
3	IN1-A	H8/3664 - CN2-20
4	OUT1-A	モータ 1 (+)
5, 10, 14	V _{CC} , V _S	乾電池 +9V (2個並列)
6	OUT2-A	モータ 1 (-)
7, 8	GND	グラウンド
9	OUT2-B	モータ 2 (-)
11	OUT1-B	モータ 2 (+)
12	IN1-B	H8/3664 - CN2-22
13	IN2-B	H8/3664 - CN2-23

定でき、表6.5に示すように2chのPWM信号により1モータの回転速度が調整できます(第4章 4.2.3「PWM多チャンネル出力プログラム」参照)。

このとき、モータの回転速度はPWM信号のデューティ比の大きさに比例するので、H8においてデューティ比を意味する値in1a, in1b, in2a, in2bの設定が重要となります。そこで、パソコンとH8をシリアル通信させて、ハイパーターミナル上からいずれかのキーを入力し、パルス1周期におけるカウント数(20ミリ秒のcount値)を画面表示してin1a, in1b, in2a, in2bを決定してください。リスト6.1のプログラムにおいては、パルス1周期で8880回(count=8880)カウントされているため、モータの回転するデューティ比を68%(countが6000回時)、モータの回転が停止し

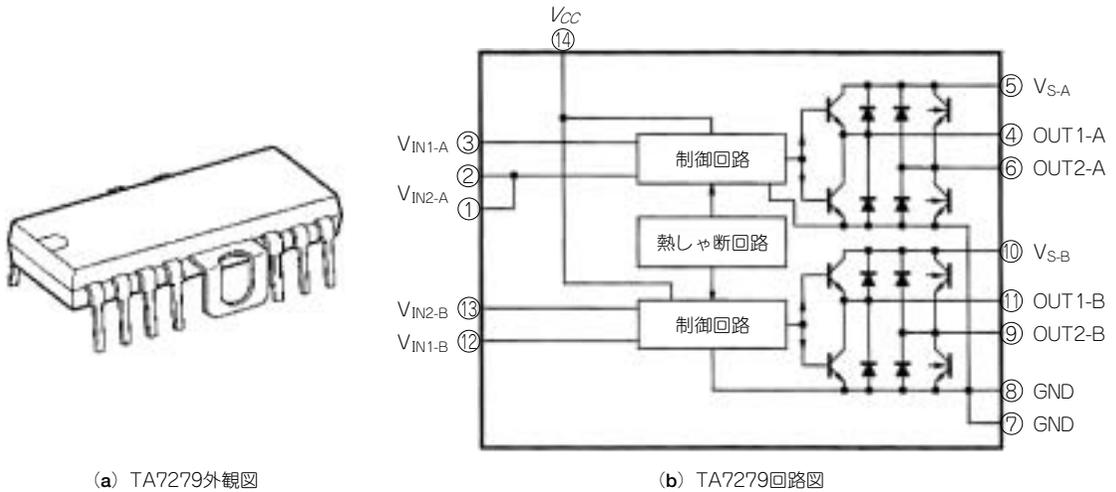


図6.4 モータ・ドライブIC：TA7279

表6.4 TA7279の制御

H8からの信号		モータへの出力		
IN1	IN2	OUT1	OUT2	MODE
High	High	Low	Low	ブレーキ
Low	High	Low	High	正転
High	Low	High	Low	逆転
Low	Low	ハイ・インピーダンス		ストップ

リスト6.1 前進駆動

```
#include <3664f.h>
void init_serial (void);
unsigned char receive (void);
void send (unsigned char ch);
void main(void){
    int count=0;
    int in1a, in1b, in2a, in2b;
    unsigned char buf;

    IO.PMR1.BYTE=0x00; /* ポート1:DIO設定*/
    IO.PCR1=0xff; /* ポート1:DO設定 */
    TW.TCRW.BIT.CCLR=1; /* TCNTがGRAでクリア */
    TW.TCRW.BIT.CKS=3; /* クロックφ/8 */
    TW.GRA=40000; /* パルスの周期 */
    TW.TMRW.BIT.CTS=1; /* カウント・スタート */
    init_serial (); /* シリアル通信の初期設定 */
}
```

リスト 6.1 前進駆動(つづき)

```

while(1){
    count++;
    /* DIO4chをデューティ値でOFF */
    if(count==in1a){IO.PDR1.BIT.B4=0;}
    if(count==in1b){IO.PDR1.BIT.B5=0;}
    if(count==in2a){IO.PDR1.BIT.B6=0;}
    if(count==in2b){IO.PDR1.BIT.B7=0;}

    /* 20ミリ秒間隔で実行 */
    if(TW.TSRW.BIT.IMFA==1){
        TW.TSRW.BIT.IMFA=0;

        /* パソコンからの送信がある場合に実行 */
        if (SCI3.SSR.BIT.RDRF == 1) {
            SCI3.SSR.BIT.RDRF = 0;

            /* 20ミリ秒間のカウント数であるcount値を送信 */
            buf=(unsigned char)(count/10000);
            send(buf+48);
            count=count-(int)buf*10000;
            buf=(unsigned char)(count/1000);
            send(buf+48);
            count=count-(int)buf*1000;
            buf=(unsigned char)(count/100);
            send(buf+48);
            count=count-(int)buf*100;
            buf=(unsigned char)(count/10);
            send(buf+48);
            count=count-(int)buf*10;
            buf=(unsigned char)(count);
            send(buf+48);
            send('\n');
            send('r');
        }

        /* 前進駆動 */
        in1a=6000;    in1b=10;    /* モータ1*/
        in2a=6000;    in2b=10;    /* モータ2*/

        /* DIO4chをすべてON */
        IO.PDR1.BIT.B4=1;
        IO.PDR1.BIT.B5=1;
        IO.PDR1.BIT.B6=1;
        IO.PDR1.BIT.B7=1;
        count=0;    /* 20ミリ秒をカウント開始 */
    }
}/* while end */
}

```

パソコンのハイパーターミナルで、いずれかのキーを入力すると、20ミリ秒間のカウント数が画面表示される

破線枠内のプログラムを書き換えると、違う移動パターンができる