

## ●教材報告

# 8ビットマイコンによる市販ラジコンレシーバの信号処理プログラム

宮城障害者職業能力開発校 情報システム科 新妻 幹也

## 1. はじめに

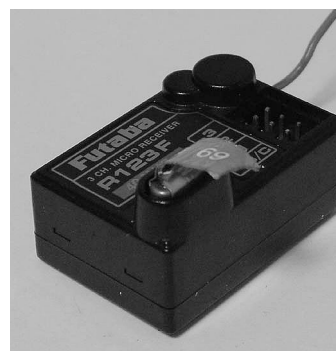
最近、多くのロボットコンテストが各地で開催され、いわゆる「ものづくり」の教育が、職業能力開発施設以外の高校や大学でも注目されるようになりました。そのロボットコンテストの内容も多岐にわたり、戦いあり、パフォーマンスあり、学ぶ人の興味を引くものになっています。中には人を笑わせるパフォーマンスロボットコンテストなるものまであります。そして、そのロボットの制御についても、マイコンを使って自動的に動作するものや、ラジコンで人が操縦するものまでいろいろです。ラジコンで制御するものは、基本的にマイコンを使わずに作るロボットも多いようですが、ラジコンのレシーバ(受信機)からの信号をマイコンで独自に処理することによって、ラジコンであっても、多彩な動きを出すことが容易になります。そこで、今回は市販のラジコンレシーバ(受信機)の信号をマイコンで処理し、独自のコントロールを可能にする信号処理プログラムを作成したので紹介します。

## 2. ラジコンレシーバ(受信機)

市販のラジコン(プロポ)には、各メーカーから送信機と受信機がセットで販売されているものが一般的です。周波数は地上用では27MHz帯12バンド、40MHzでは8バンド用意されています。送信機のスイッチを入れると、プロポの受信機からは、各チャ



ラジコン送信機



ラジコンレシーバ(受信機)

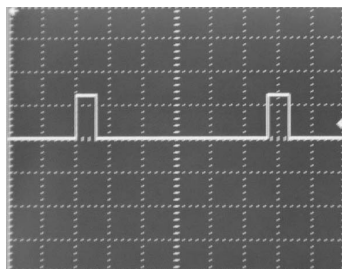


Fig1

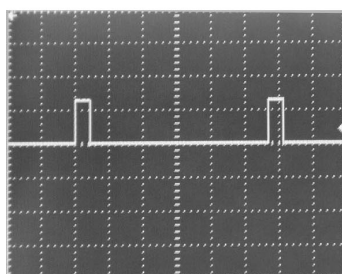


Fig2

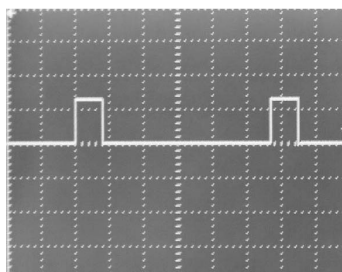
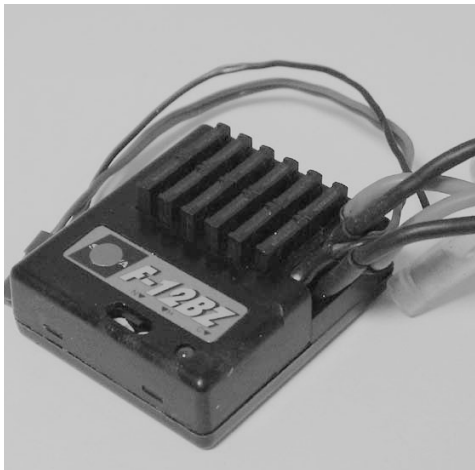


Fig3

ンネル（一般のものは2ch～4ch）ごとに、Fig1（ニュートラル状態）に示すような矩形波が出ていることがわかります。そして、送信機のスティック（ホイール、またはトリガー）を操作することで、この矩形波のデューティが変化するようになっています。Fig2, Fig3はそれぞれ、スティックを右、左に動かしたときの状態です。

一般的な使い方としては、この受信機に市販のサーボや、FETアンプ（モータのスピードコントローラ）を接続してロボットの動作に使います。ただ、その場合、実装スペースやロボットの動作オペレーションの関係で、必ずしも作ろうとしているロボットに合うものとならないこともあります。

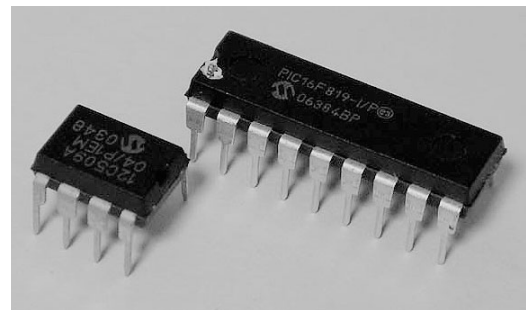


ラジコン用FETアンプ

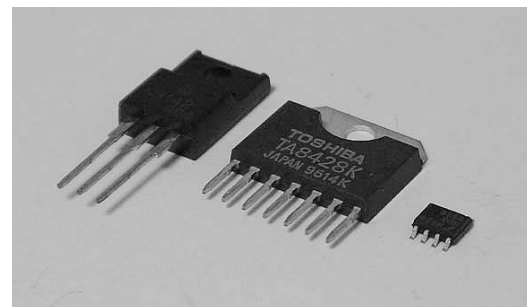


ラジコン用サーボ

そこで、受信機からの信号をマイコンで処理することにより、独自の制御回路を作ることにより、



今回使用したPIC12F629（8ピン）と、PIC16F819（18ピン）



左：TO-220タイプFET  
真ん中：モータドライバIC TA8428K  
右：今回使用したFET 4935

した。これによりラジコンによるロボットの制御にモータ以外のアクチュエータも多彩に利用できるようになります。

### 3. 使用するマイコン

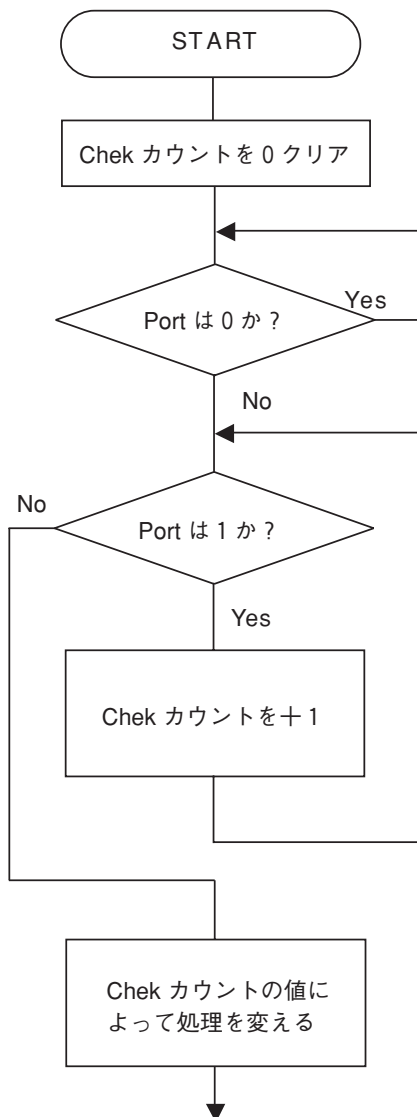
今回使用するマイコンはDIPタイプで使いやすいPICを使ってみました。プログラムはC言語で行いますので、マイコン特有の記述部分を除けば、H8マイコン等でも利用できます。PICには多くの種類がありますが、どのPICマイコンでも基本的には使うことができます。今回使用したPICはPIC12F629で、8ピンのフラッシュタイプです。

より小さなロボットにも、省スペースで組み込みが可能なので、これを選んでみました。プログラムエリアは1K、RAMは64バイトですが、今回プログラムを見ておわかりのように、十分なメモリー容量ですが、I/Oポートをすべて使い切っていますので、他にポートを使う必要があるときは、PIC16F84AやPIC16F819などを使うとよいでしょう。いずれも、使用できる電圧の範囲は2.0V～5.5Vで使いやすいものになっています。

また、PIC12F629の場合は内部クロックで4MHzの動作は可能ですので、高速な処理を必要でなければマイコンの動作スピードを決定するレゾネータも省略できます。このPICは電源ピンを除いて6ピンをI/Oに割り当てできますが、4番ピンだけは入力にしか使えませんので、注意が必要です。

#### 4. 信号処理プログラム

受信機からの信号は前述したように、矩形波のデューティが変化するようにになっています。この形式はかなり昔からこうなっていたように記憶しています。そして、この信号をアナログ専用のIC等で処理していたと思われます。今回は、それをデジタル処



理しようということです。

もちろん、すでにメーカ製の特殊なモータアンプにはデジタルのものも存在します。

今回紹介する矩形波の処理アルゴリズムは上のようないち極簡単なものです。

Portは、レシーバからの信号端子をマイコンのI/Oにつないだ部分です。

まず、矩形波の左端の立ち上がりまで待機し、立ち上がったら、立ち上がっている間Chekカウントを1増加させ、Lowレベルになったらカウントをやめ、その時点の値により処理を分けるというものです。要するに矩形波の幅を数値的に読み取るというものです。

これを各チャンネルごとにやれば、何チャンネルにも対応できます。次のプログラム例は、ホイールタイプ送信機の2ch分の信号を処理して、2個のモータを駆動（今回は単純にON, OFFのみ）して、簡単な車を走らせるものです。ホイールタイプの送信機では、左手の人差し指でトリガーを引くことで、前進、トリガーを押すことで後退、ホイールを右に回すと、右旋回、左に回すと左旋回というような動作をするようにしてみました。

このアルゴリズムでは、矩形波の幅が数値で返ってきますので、その数値を使ってモータのスピードをコントロールするプログラムにすることも可能です。

```

//-----
//レシーバ信号処理制御プログラム
// 2006-11-30 start 1-3
// Programed by Mikiya Niitsuma
//-----
#include <12F629. h >
#fuses INTRC, NOWDT, NOPROTECT
#use delay (clock=4000000)
#byte RA = 5
#bit ch_0 = RA. 2 //受信機からの信号 0
#bit ch_1 = RA. 3 //受信機からの信号 1

void Motor_S (void) //motor STOP関数

```

```

RA = 0x33;//11,0011
delay_ms (10) ;
}
void Motor_F (void) //motor 前進関数
RA=0x21;//10,0001
}
void Motor_B (void) //motor 後退関数
RA=0x12;//01,0010
}
void Motor_L (void) // 左旋回関数
RA=0x11;//01,0001
}
void Motor_R (void) // 右旋回関数
RA=0x22;//10,0010
}

main ()
{
long sig0, sig1;//レシーバからの信号
//sig0 ホイール
//sig1 トリガー

long
high=218, low=178, max=270, mini=150;
set_tris_a (0x0C) ;
delay_ms (100) ;
RA = 0;
//モータストップ
Motor_S () ;
while (1) //1
//ホイールの信号を解析
while (!ch_0) ;
sig0=0;
while (ch_0) {
sig0++;
}
if (sig0<mini || sig0>max) continue;

//トリガーの信号を解析
while (!ch_1) ;
sig1=0;

```

```

while (ch_1) {
sig1++;
}
if (sig1<mini || sig1>max) continue;

//ホイールまたはトリガーでプロポ操作されたとき
if (sig0<low || sig0>high || sig1<low ||
sig1>high) //2
if (sig1>high) {
//LRモータ正転
Motor_F () ;
}
if (sig1<low) {
//LRモータ逆転
Motor_B () ;
}
if (sig0>high) {
//左旋回
Motor_L () ;
}
if (sig0<low) {
//右旋回
Motor_R () ;
}
}
//2
else{ //プロポ操作がないときはモータを
//ブレーキ状態
Motor_S () ;
}
}
//1
}

```

## 5. 使用したCコンパイラ

今回のCプログラムには、CCS-Cコンパイラを使用しましたが、アルゴリズムの部分は同じですので、他のコンパイラでも、多少の手直しをすれば動作すると思います。また、PIC16F84Aを使ってアセンブラで記述したい場合はフローチャートを参考にして

みてください。

## 6. サンプル制御回路

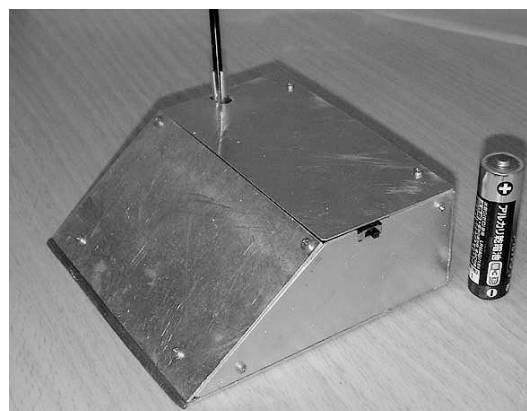
今回、上記プログラムを次のような回路で、2つのモータを制御することを想定しています。

P型FETとN型FETを使った基本的なブリッジ回路です。これによりモータの正回転、逆回転が可能になります。マイコンからのA信号 (GP0), B信号 (GP1) でそれぞれ、正回転、逆回転、そして、A, B両信号で、モータにブレーキがかかります。このロジックをFETの前段で、4011を使って実現しています。

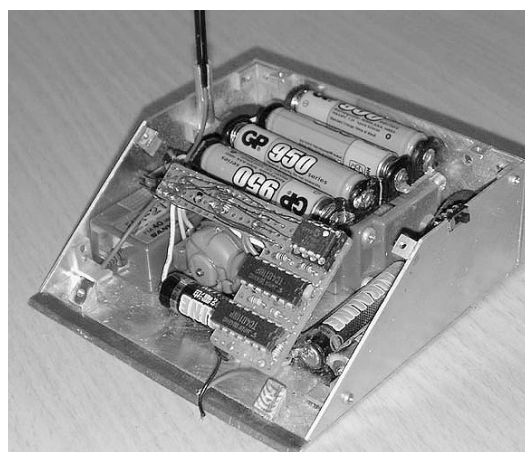
今回の例ではFETにフェアチャイルドのSOP8ピンハーフピッチサイズのP型 (4935) とN型 (9936) のデュアルタイプのFETを使用してみました。1つのチップに2つのFETが入っているので、省スペースに役立ちます。ただ、Maxで5Aまでしか流せませんので、大きなモータを使う場合はもっと、電流容量の大きいFETにする必要があります。模型用FA-130タイプのモータなら全く問題ありません。もちろん、モータドライバICのTA7267や TA8428などを使うと、回路がもっと簡単になりますが、流せる電流は上記のICの場合Max3Aです。また、モータなどに負荷が掛かりすぎると、プロテクト機能が働いて止まってしまうので、注意が必要です。訓練生に作らせるのであればFETで構成した方が基本回路の勉強にもなり、好都合かもしれません。

モータの動きは内部ON抵抗のため、FETの方が専用ドライバよりもよいことがわかります。

また、マイコンの電源とモータの電源については、兼用することも可能ですが、モータやソレノイド等を負荷とした場合、起動電流が大きいため、瞬間的な電圧降下によって、マイコンが誤動作することがありますので分けた方が確実です。どうしても兼用したい場合は、実験して問題ないかどうか検証してみてください。



PIC12F629を使用した10cm×10cmの小型ロボット



PIC12F629を実装した小型ロボットの内部



市販のサーボ、アンプ、レシーバと小型ロボットの比較

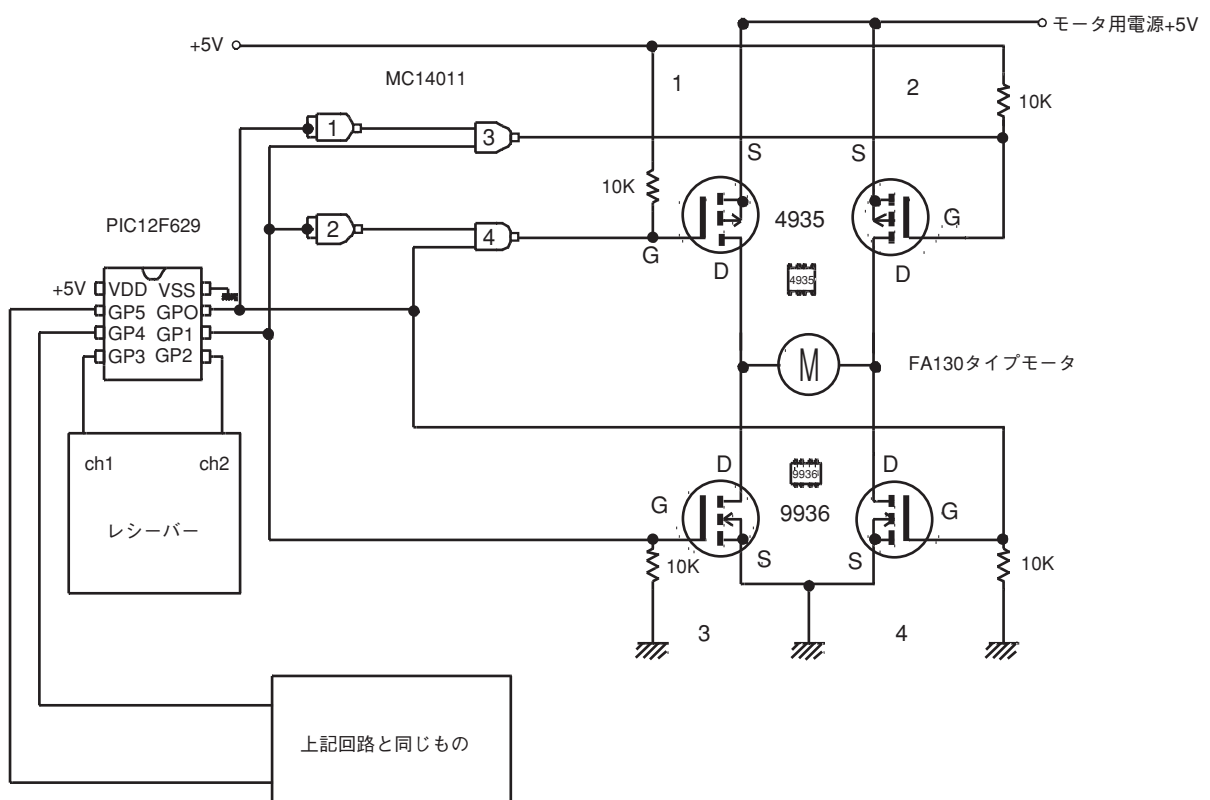
## 7. おわりに

20数年前に一般の人でも比較的手軽にマイコンが

購入できるようになりましたが、それでもその当時は、CPUやメモリー、タイマー、I/Oなどはそれぞれが別々のチップでした。したがって、今回のような簡単な制御ボードでもかなりのスペースを必要とする基板になるのが普通でした。それが、今や120円程度の1チップマイコンで、そのすべてを含み、プログラムを何度も書き換え可能という、当時では想像もつかない状況になっています。訓練生に「もの

づくり」の基礎をじっくり勉強してもらうにはコスト的にも製作の容易性にしても好都合であると思います。電子制御システム科やメカトロニクス科の訓練教材として参考にしていただければ幸いです。

今回のプログラムを参考にして、ラジコン型のロボット作りにもバリエーションが加えられ多くの創造的ロボットが生み出されることを期待したいと思います。



制御回路例