

第4章 ホストコンピュータからプロポの制御

成岡 優

2006年10月15日

全般の説明

全体説明で示したとおり、室内飛行機の機体側は、ラジコン無線インフラを通じて制御情報や外界の情報を取り込んでいる。つまり送信側はラジコンで用いられるプロポという装置になる。プロポには外部機器からプロポを制御することができるトレーナー機能という機構が備わっており、本プロジェクトではそれを活用することにした。トレーナー機能はトレーナー端子に制御量に応じたパルス幅をもつパルス信号を入力することによって有効となる。従って、このパルス信号を生成できれば様々な外部機器からプロポを制御できることになる。

トレーナー端子に入力すべき信号は次の図のようなものである。

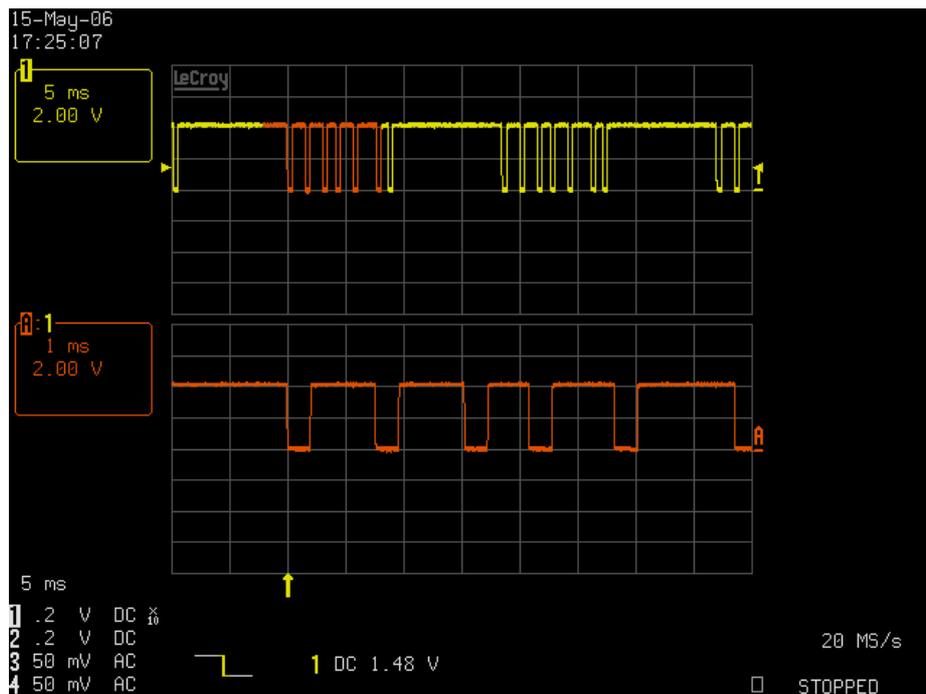


図1 トレーナー端子に入力する信号

また、コードであらわすと次のようになる。

```
1 if(時刻が 18ms で割り切れたら){  
2   for(i=1; i<=(チャンネル数); i++){  
3     Low 0.4ms;  
4     High (Ch.i の操縦量、最小=0.8ms、最大=1.6ms);  
5   }  
6   Low 0.4ms;  
7 }
```

本プロジェクトではプロポを制御する外部機器として PC を想定している。PC はこのようなパルス信号を直接生成するのは困難であるので、PC とプロポの間にマイコンを挿入し、そのマイコンがトレーナー端子へパルス信号を送ることにした。マイコンと PC はシリアルポートで通信するものとし、マイコンは Microchip 社の PIC 16F648A(秋月で購入できる)を採用した。

以下、ソフトウェア、ハードウェアについて詳細を掲載する。

ソフトウェア

PC とプロポを接続するマイコンのためのプログラムが ropo2pc.c[1] である。これは CCS 社の C コンパイラでコンパイルすると実行形式に変換されるので、それを適切な PIC プログラマ(プログラム書込み機のこと)を利用して書込む。書込みには秋月電子で販売している『AKI - PIC プログラマー Ver. 3.5 キット』と『AKI - PIC プログラマー Ver4 パージョンアップキット』を利用した。また CCS 社のコンパイラはバージョンが古いと PIC 16F648A 用のコードを正しく処理できないというバグが存在する。Ver.3.249 で処理すると正常なプログラムが生成された。

ソースコード 1 ropo2pc.c

```
1 #include <16F648A.h>
2 #fuses HS,NOWDT,NOPROTECT,NOLVP
3 #use delay(clock=20000000)
4 #use rs232(baud=9600, rcv=PIN_B1, xmit=PIN_B2, parity=N, bits=8)
5 #use fast_io(A)
6
7 ///priority rda,timer1
8
9 #define INIT_WIDTH 400
10
11 #define PROPO_CHANNEL 6
12 #define SCHEDULE_SIZE ((PROPO_CHANNEL << 1) + 1)
13
14 #define TMR1L 0x0E
15 #define TMR1H 0x0F
16 #define T1CON 0x10
17
18 unsigned char schedule_H[SCHEDULE_SIZE];
19 unsigned char schedule_L[SCHEDULE_SIZE];
20
21 void set_schedule(unsigned char target, unsigned long width){
22     unsigned long temp_width;
23     temp_width = ~((width * 5) >> 1);
24     schedule_H[target] = (unsigned char)(temp_width >> 8);
25     schedule_L[target] = (unsigned char)(temp_width & 0xFF);
26 }
27
28 void set_width(unsigned char buf){
29     static unsigned char byte_count = 0;
30     static unsigned char target_channel;
31     static unsigned long new_width;
32
33     switch(byte_count){
34         case 0:
35             if(buf < PROPO_CHANNEL){
36                 target_channel = buf;
37                 break;
38             }
39             putc(' $ ');
40             return;
41         case 1:
42             new_width = buf;
```

```

43     new_width <<= 8;
44     break;
45 default:
46     new_width |= buf;
47     if((new_width >= 400) && (new_width <= 1600)){
48         set_schedule((target_channel << 1) + 1, new_width);
49         putc('!');
50     }
51     byte_count = 0;
52     return;
53 }
54 byte_count++;
55 }
56
57 /*
58 #INT_RDA
59 void invoked_by_sci_rx(){
60     set_width(getc());
61 }
62 */
63
64 #INT_TIMER0
65 void invoked_by_timer0(){
66     static unsigned char toggle = 0;
67     if(toggle == 0){
68         set_timer0(0); // tick=(0.2 * 256)us * 256 = 13.1072 ms
69         toggle = 1;
70     }else{
71         set_timer0(161); // tick=(0.2 * 256)us * 95 = 4.864 ms
72         toggle = 0;
73         (*TMR1H) = 0xFA;
74         (*TMR1L) = 0x00;
75         (*T1CON) |= 0x01;
76     }
77 }
78
79 #INT_TIMER1
80 void invoked_by_timer1(){
81     static unsigned char target = 0;
82     static unsigned char *p_schedule_H = schedule_H;
83     static unsigned char *p_schedule_L = schedule_L;
84
85     if(target & 0x01){
86         output_high(PIN_A0);
87     }else{
88         output_low(PIN_A0);
89     }
90     if(target >= SCHEDULE_SIZE){
91         (*T1CON) &= ~(0x01);
92         target = 0;
93         p_schedule_H = schedule_H;
94         p_schedule_L = schedule_L;
95         return;
96     }
97
98     if((*p_schedule_L) > ((*TMR1L) += (*p_schedule_L))){
99         (*TMR1H) += ((*p_schedule_H) + 1);
100    }else{
101        (*TMR1H) += (*p_schedule_H);
102    }
103
104    target++;
105    p_schedule_H++;
106    p_schedule_L++;

```

```

107 }
108
109 void main() {
110     unsigned char i;
111
112     set_tris_a(0x00);
113     setup_timer_0(RTCC_INTERNAL | RTCC_DIV_256);
114
115     (*TMR1L) = (*TMR1H) = 0;
116     (*T1CON) = 0x10;
117
118
119     for(i = 0; i < SCHEDULE_SIZE; i++){set_schedule(i, INIT_WIDTH);}
120
121     // setup interrupts
122     enable_interrupts(INT_TIMER0);
123     enable_interrupts(INT_TIMER1);
124     enable_interrupts(GLOBAL);
125
126     set_timer0(0xA0);
127
128     while(TRUE){set_width(getc());}
129 }

```

また、PC とマイコンで通信する必要があるが、それには `propo_host.h`[2] を利用する。使用方法は `example.cpp`[3] も参照のこと。

ソースコード 2 `propo_host.h`

```

1 /**
2  * プロポをコントロールする基板に指令をあたえるための
3  * PC 側ソフトウェア。
4  *
5  * 使い方:
6  *
7  * #include "propo_host.h"
8  *
9  * PropoHost propo(com_port_no = 1); //引数なしの場合はCOM1 が設定される
10 * propo.set(int channel, double ratio);
11 *
12 * 同じディレクトリのexample.cpp も参考のこと。
13 *
14 * coded by fenrir(M.Naruoka) 2006.
15 */
16
17 #ifndef _PROPO_HOST_H_
18 #define _PROPO_HOST_H_
19
20 #include <string>
21 #include <exception>
22
23 #include <Windows.h>
24 #include <stdio.h>
25 #include <stdlib.h>
26 #include <math.h>
27 #include <iostream>
28
29 using namespace std;
30
31 #if _MSC_VER >= 1400
32 #define sprintf sprintf_s
33 #endif
34
35 #define concat_str(str1, str2) str1 ## str2

```

```

36
37 #define COM_STR "COM"
38 #define COM_DEFAULT 1
39 #define PROPO_HOST_BANDRATE CBR_9600
40
41 //トレーナーケーブルを流れる PWM 信号の最大値、最小値
42 #define PROPO_HIGH 1600 //us]
43 #define PROPO_LOW 800 //us]
44
45 #define PROPO_PACKET_SIZE 3
46
47 class PropoHostException : public exception{
48     private:
49         const string what_str;
50     public:
51         PropoHostException(const string &what_arg) : what_str(what_arg){}
52         ~PropoHostException() throw(){}
53         /**
54          * エラー内容を取得します。
55          *
56          * @return (chr *) エラー内容
57          */
58         const char *what() const throw(){
59             return what_str.c_str();
60         }
61 };
62
63 class PropoHost{
64     private:
65         HANDLE hComm;
66         DCB dcb;
67
68     void writeToContoller(const unsigned char packet[PROPO_PACKET_SIZE]) const{
69         DWORD size_TX, size_RX;
70         char buf[16];
71         WriteFile(hComm, packet, PROPO_PACKET_SIZE, &size_TX, NULL);
72         if(ReadFile(hComm, buf, sizeof(buf), &size_RX, NULL)){
73             buf[size_RX]=0;
74         }else{buf[0] = 0;}
75 #if DEBUG
76     printf("%x_%x_%x:_%d_byte_TXed._->_RX:_%s\n", packet[0], packet[1], packet[2], size_TX, buf);
77 #endif
78     }
79     private:
80     void init(const int &com_port) throw(PropoHostException){
81
82         char com_str[16];
83         sprintf(com_str, "%s%d", COM_STR, com_port);
84         cout << "TARGET_PORT: \t" << com_str << endl;
85
86         hComm = CreateFile(
87             com_str,
88             GENERIC_READ | GENERIC_WRITE,
89             0,
90             0,
91             OPEN_EXISTING,
92             FILE_ATTRIBUTE_NORMAL,
93             0
94         );
95
96         if(hComm == INVALID_HANDLE_VALUE){
97             throw PropoHostException("Couldn't open port.");
98         }
99

```

```

100     GetCommState(hComm, &dcb); //DCB を取得
101     dcb.BaudRate = PROPO_HOST_BANDRATE;
102     dcb.ByteSize = 8;
103     dcb.Parity = NOPARITY;
104     dcb.fParity = FALSE;
105     dcb.StopBits = ONESTOPBIT;
106     dcb.fBinary = TRUE; //バイナリモード
107     dcb.fNull = FALSE; //NULL バイトは破棄しない
108     dcb.fOutX = FALSE; //XON なし
109     dcb.fInX = FALSE; //XOFF なし
110     //dcb.fOutxCtsFlow = FALSE; //CTS フロー制御なし
111     //dcb.fOutxDsrFlow = FALSE; //DSR フロー制御なし
112     //dcb.fDsrSensitivity = FALSE; //DSR 制御なし
113     SetCommState(hComm, &dcb); //DCB を設定
114 }
115
116 public:
117     PropoHost(const int &com_port) throw(PropoHostException){
118         init(com_port);
119     }
120     PropoHost() throw(PropoHostException){
121         init(COM_DEFAULT);
122     }
123
124     ~PropoHost(){
125
126         CloseHandle(hComm);
127     }
128
129     /**
130     * プロポをコントロールします。
131     *
132     * @param channel チャンネル (0からはじめることに注意)
133     * @param ratio 指令値 (0.0 ~ 1.0、1.0がフル)
134     */
135     void set(const unsigned char &channel, double ratio) const{
136         unsigned char packet[PROPO_PACKET_SIZE];
137
138         packet[0] = (unsigned char)channel;
139
140         //指令値の計算
141         if(ratio < 0){ratio = 0;}
142         else if(ratio > 1){ratio = 1;}
143         unsigned int command_width = (unsigned int)(ratio * (PROPO_HIGH - PROPO_LOW)) + PROPO_LOW;
144         //printf("%d\n", command_width);
145
146         packet[1] = (unsigned char)((command_width >> 8) & 0xFF);
147         packet[2] = (unsigned char)(command_width & 0xFF);
148
149         writeToContoller(packet);
150     }
151 };
152
153 #endif /* __SERVO_HOST_H__ */

```

ソースコード 3 example.cpp

```

1  //#include <cstdlib>
2  #include <Windows.h>
3  #include <stdio.h>
4  #include <stdlib.h>
5  #include <limits.h>
6  #include <time.h>
7

```

```

8 using namespace std;
9
10 #define DEBUG 1
11 #include "propo_host.h"
12
13 #define PERIOD 500 // [ms]
14 #define INTERVAL 10 // [ms]
15
16 #ifndef PI
17 #define PI 3.14159265358979323846
18 #endif
19
20 int main(){
21
22     PropoHost propo(1);
23
24     srand((unsigned int)time(NULL));
25
26     //余弦的往復運動をする
27     for(int loop = 0; true; loop++){
28         double ratio = (cos(PI * (loop * INTERVAL) / PERIOD) + 1) / 2;
29         printf("%d_passed_>_%f\n", loop, ratio);
30         propo.set(0, ratio);
31         propo.set(2, ratio);
32         //Sleep(INTERVAL);
33     }
34 }

```

ハードウェア

ハードウェアは図 2 にある回路のものを手作業で作成した。完成したハードウェアを図 3 に示す。採用しなかったものの、その後この回路を持つ基板を作る機会があったが、これも合わせて図 4 に掲載する。必要部品は表 1 に示す。

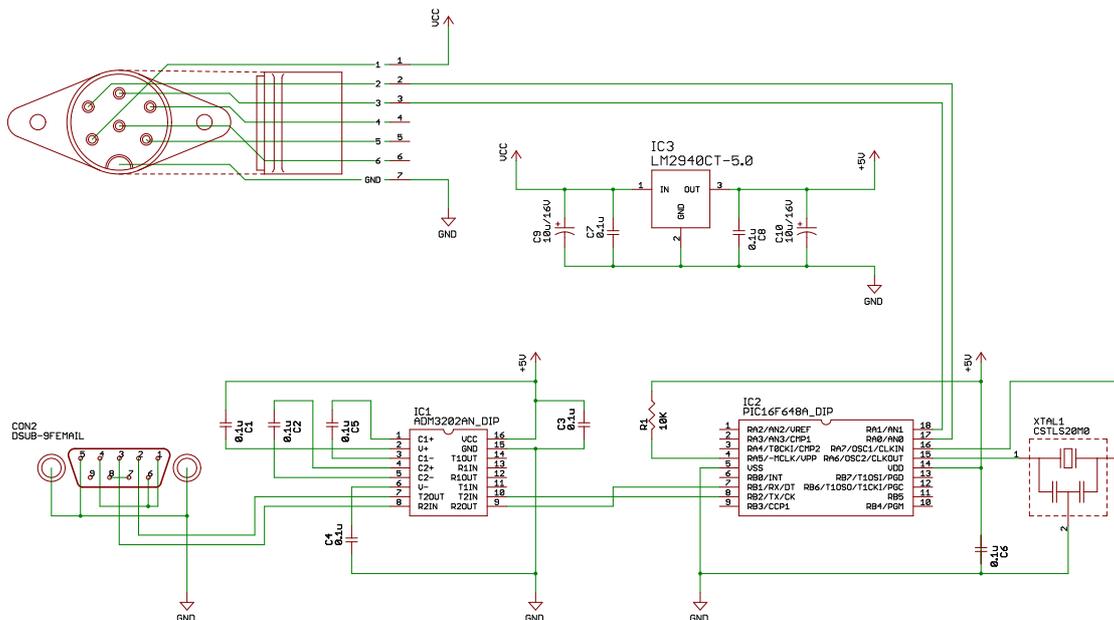


図 2 プロポ I/F 基板 回路図



図3 プロポ I/F 基板 外観

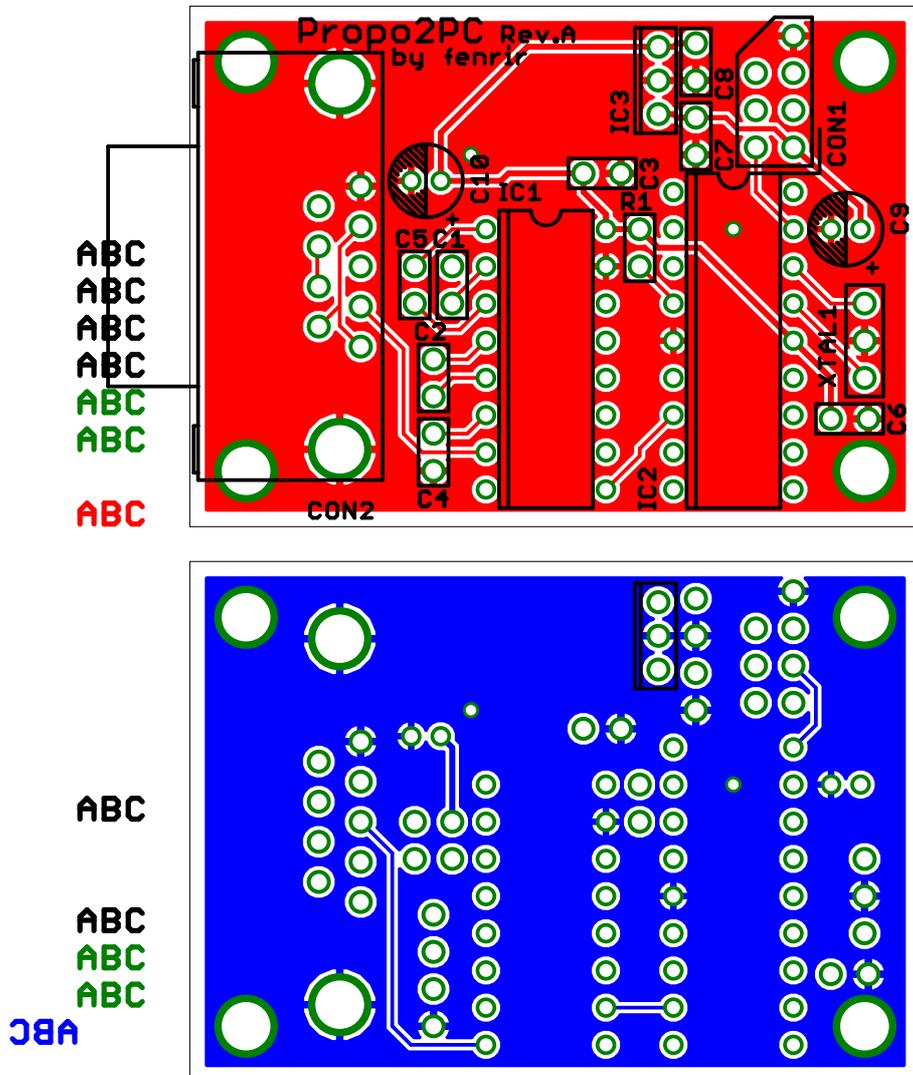


図4 プロポ I/F 基板 レイアウト (表・裏)

表1 プロポ I/F 部品表

名前	値	説明
C1	0.1 μ	
C2	0.1 μ	
C3	0.1 μ	
C4	0.1 μ	
C5	0.1 μ	
C6	0.1 μ	
C7	0.1 μ	
C8	0.1 μ	
C9	10 μ /16V	OS コン、千石地下1階
C10	10 μ /16V	OS コン、千石地下1階
CON1	DIN6	ヒロセ地下1階
CON2	DSUB-9FEMAIL	DSUB-9メス
IC1	ADM3202AN DIP	PDIP16パッケージ、秋月
IC2	PIC16F648A DIP	PDIP18パッケージ、秋月
IC3	LM2940CT-5.0	3端子レギュレータ、秋月
R1	10K	
XTAL1	CSTLS20M0	セラロック 20MHz、秋月
	ケース	適当に、千石地下1階
	トレーナーケーブル	ラジコンショップで Futaba 用の旧タイプのもの
	シリアルケーブル	