Wiring Pi で Raspberry Pi ZERO の周辺機器を制御する（その 10）

2017.08.05 新海

★今回の概要

＃9で述べた**「GPS 計測値を 920MHz 通信で送信する」**を実際に**缶サット**（モデルロケットから放出される人工衛星のようなもの。とても衛星軌道までは行かないので、パラシュートで降りてきます）に実装します。
プログラムの機能はほとんど＃9と同じですが、実装にあたってプログラムを変えている部分について解説します。

★ハードソフトの追加部分

1．追加した機能

（1） GPSデータをファイルに書き出す（ログ）機能

（2） （1）を書き出すファイル名を、立ち上げのつど変更する機能
関数 incrementfopen()で新しいファイル名を作っています。
ディスク上の number.txt というファイルに番号を入れておき、起動するたびにそれに1を加えて、
ファイル名の末尾にします。数字は number.txt に書き戻しておきます。
RaspberryPi とは特に関係ないロジックだけの話なので、これ以上の説明は省略します。
この機能の理由は、外からファイルを操作できないので、起動毎にファイルを分けておきたいことと、
電源の瞬断等による再起動時に、ファイルが上書きされないようにする為です。

（3） プログラム立ち上げ完了を示すために LED（右目）を点灯

2．ハードウェアの変更点

（1）実装（アニメ「けものフレンズ」の「ボス」のような外観、直径6cmの大きさに実装）

（2）電源をリチウムポリマー電池に変更

（3）動作していることを表すLEDを6番ピンに追加。

★プログラムのキーポイント

1. エラーの把握とリトライの追加
今までのプログラムでは、命令の機能がよくわかるように、エラー処理を省略していました。
しかし、実際のプログラムでは、その命令がエラーになったかどうかを判断して、
もし、エラーなら対策をしなければなりません。
ロケットや缶サットの場合、コンソールに表示しても、空の上まではだれも見に来てくれないので、
エラーの場合は、ひたすらエラーが出なくなるまで処理を繰返すことになります。

例1：wiringPiSetup

① 従来の記述

```
wiringPiSetup();
```

② エラーの把握とリトライの追加

```
int e;
e=wiringPiSetup();
  while ( e != 0 ) {   e=wiringPiSetup();   }
```

おなじみの wiringPiSetup ですが、エラーかどうかを知るため、新たに e という変数を定義して、エラーがあれば、wiringPiSetup を繰返すようにしています。

例2：シリアル通信のオープン

①従来の記述

```
serialOpen("/dev/ttyUSB0",9600);
```

②エラーの把握とリトライの追加

```
Int fd1;
fd1=serialOpen("/dev/ttyUSB0",9600);
while ( fd1 < 0 ) {
   Fd1=serialOpen("/dev/ttyUSB0",9600); }
```

fd1 という変数は従来からあるファイルポインタですが、シリアル通信オープンの結果がうまく行ったかどうかも表しています。（値が0より小さいときはエラー）
うまく行っていなければ、シリアル通信のオープンをうまく行くまで繰返します。

以上

缶サットプログラム boss

```
#include<wiringPi.h>
#include<wiringSerial.h>                    /*UART(シリアル通信)を使うためのライブラリ*/
#include<termios.h>                         /* 通信設定を行う構造体 tarmios のライブラリ */
#include<string.h>
#include<stdio.h>
#include<stdlib.h>
char *incrementfopen();
char fname[20];
int main(){
int i;
FILE *fp;
int fd1,fd2;                                /* UART(シリアル通信)のファイルポインタ */
int e;                                          /* エラー識別子 */
*fname=*incrementfopen();                       /* 新しいファイル名を取得する関数 */
/* 通信設定を行う構造体 tarmiios の定義 */
struct termios ttyparam1,ttyparam2;
unsigned char str[255] ={"TXDA "};
unsigned char savestr[255] ;
unsigned char sendstr[255] ={"TXDA "} ;
/* UART(シリアル通信)のオープン */
fd1=serialOpen("/dev/ttyUSB0",9600);                /* USB(GPS)側 */
while ( fd1 < 0 ) {                                 /* もしエラーならば */
   fd1=serialOpen("/dev/ttyUSB0",9600); }           /* OK になるまでリトライ */
fd2 =serialOpen("/dev/ttyAMA0",19200);              /* 920MHz 無線側 */
while ( fd2 < 0 ) {                                 /* もしエラーならば */
   fd2=serialOpen("/dev/ttyAMA0",19200); }          /* OK になるまでリトライ */
/* USB(GPS)側の通信設定 */
tcgetattr(fd1,&ttyparam1);
ttyparam1.c_iflag &=~IXON;
ttyparam1.c_iflag &=~IXOFF;
ttyparam1.c_cflag &=~CSIZE;
ttyparam1.c_cflag |=CS8;
tcsetattr(fd1,TCSANOW,&ttyparam1);
/* 920MHz 無線側の通信設定 */
tcgetattr(fd2,&ttyparam2);
ttyparam2.c_iflag &=~IXON;
ttyparam2.c_iflag &=~IXOFF;
ttyparam2.c_cflag &=~CSIZE;
ttyparam2.c_cflag |=CS8;
tcsetattr(fd2,TCSANOW,&ttyparam2);
/* キャラクタ入出力モードにする IM910 のコマンド"ECIO" 発行文字列 */
char cmd [7]={"ECIO"};                              /* ECIO コマンド */
cmd[4]=0x0d;                                        /* <CR> IM910 コマンドの終わり(CR+LF) */
cmd[5]=0x0a;                                        /* <LF> IM910 コマンドの終わり(CR+LF) */
cmd[6]=0x00;                                        /* serialPuts データの終わり 0x00 */
e=wiringPiSetup();                                  /* wiringPi のピン番号を使う Setup */
while ( e != 0 ) {   e=wiringPiSetup();  }          /* エラーなら OK になるまでリトライ */
pinMode(4,INPUT);                                   /* 4 番ピンを Busy の検知(入力)に使う */
pinMode(6,OUTPUT);                                  /* 6 番ピンを立上検知(出力)に使う */
digitalWrite(6,1);                              /* 6 番ピン(右目 LED)を ON にする */
while (digitalRead(4)){}                            /* 4 番ピンが Busy でない限り */
serialPuts(fd2, cmd);                           /* cmd が示す文字列をシリアル通信で送る */
serialFlush(fd1);                                   /* 受信バッファのクリア */
unsigned char chkstr[7];                        /* "$GPGGA と比較するための作業エリア */
```

```
chkstr[6]=0x00;                 /*  エリアの最後に文字列の終わりを示す 0x00 をセット  */

pinMode(4,INPUT);                               /* 4 番ピンを Busy の検知(入力)に使う  */
pinMode(6,OUTPUT);                              /* 6 番ピンを立上検知(出力)に使う  */
digitalWrite(6,1);                          /* 6  番ピン(右目 LED)を ON  にする  */
while (digitalRead(4)){}                        /* 4 番ピンが Busy でない限り  */
serialPuts(fd2, cmd);                       /* cmd  が示す文字列をシリアル通信で送る  */
serialFlush(fd1);                               /*  受信バッファのクリア  */
unsigned char chkstr[7];                    /* "$GPGGA  と比較するための作業エリア  */
chkstr[6]=0x00;                 /*  エリアの最後に文字列の終わりを示す 0x00 をセット  */
while('1')
{
   str[5]=serialGetchar(fd1);
   str[6]=serialGetchar(fd1);
   i=6;
   while (str[i-1]!=0x0d || str[i]!=0x0a)
   {
         i++;
         str[i]=serialGetchar(fd1);
   }
      str[i+1]=0x00;
      strncpy(chkstr,str+5,6);
      if (strcmp(chkstr,"$GPGGA")==0){
         while (digitalRead(4)){}                   /* 4 番ピンが Busy でない限り  */
         strcpy(savestr,str+5);
         strcpy(sendstr+5,str+12);
         fp=fopen(fname,"a");
            while (fp==NULL){fp=fopen(fname,"a");}    /*エラーなら OK になるまでリトライ*/
         fprintf(fp,"%s",savestr);
         e=fclose(fp);
            while (e < 0){e=fclose(fp);}                /*エラーなら OK になるまでリトライ*/
         serialPuts(fd2,sendstr)          ;        /* str をシリアル通信で送る*/
       }
}
return 0;
}
```

起動のたびに新しいファイル名を生成する関数 incrementfopen()

```
/*   起動のたびに新しいファイル名を生成する関数
       同じディレクトリに、末尾の数字を入れる number.txt が必要 */
char *incrementfopen()
{
FILE *fp2,*fp3;
int i=0,j,k,l;
int e;

char num[4]={0};
fname[0]='D';
fname[1]='A';
fname[2]='T';
fname[3]='A';
char txt[4] = {".txt"};
fp2=fopen("number.txt","r");
   while (fp2==NULL){fp2=fopen("number.txt","r");}
fscanf (fp2,"%d",&i);
e=fclose(fp2);
   while (e < 0){e=fclose(fp2);}
i=i+1;
fp3=fopen("number.txt","w");
   while (fp3==NULL){fp3=fopen("number.txt","w");}
fprintf (fp3,"%d",i);
e=fclose(fp3);
   while (e < 0){e=fclose(fp3);}
sprintf(num,"%d",i);
for (j=0;num[j] != 0x00;j++){
   fname[4+j]=num[j];
}
k=4+j;
for (l=0;l<4;l++){
   fname[k]=txt[l];
   k++;
}

return fname;
}
```