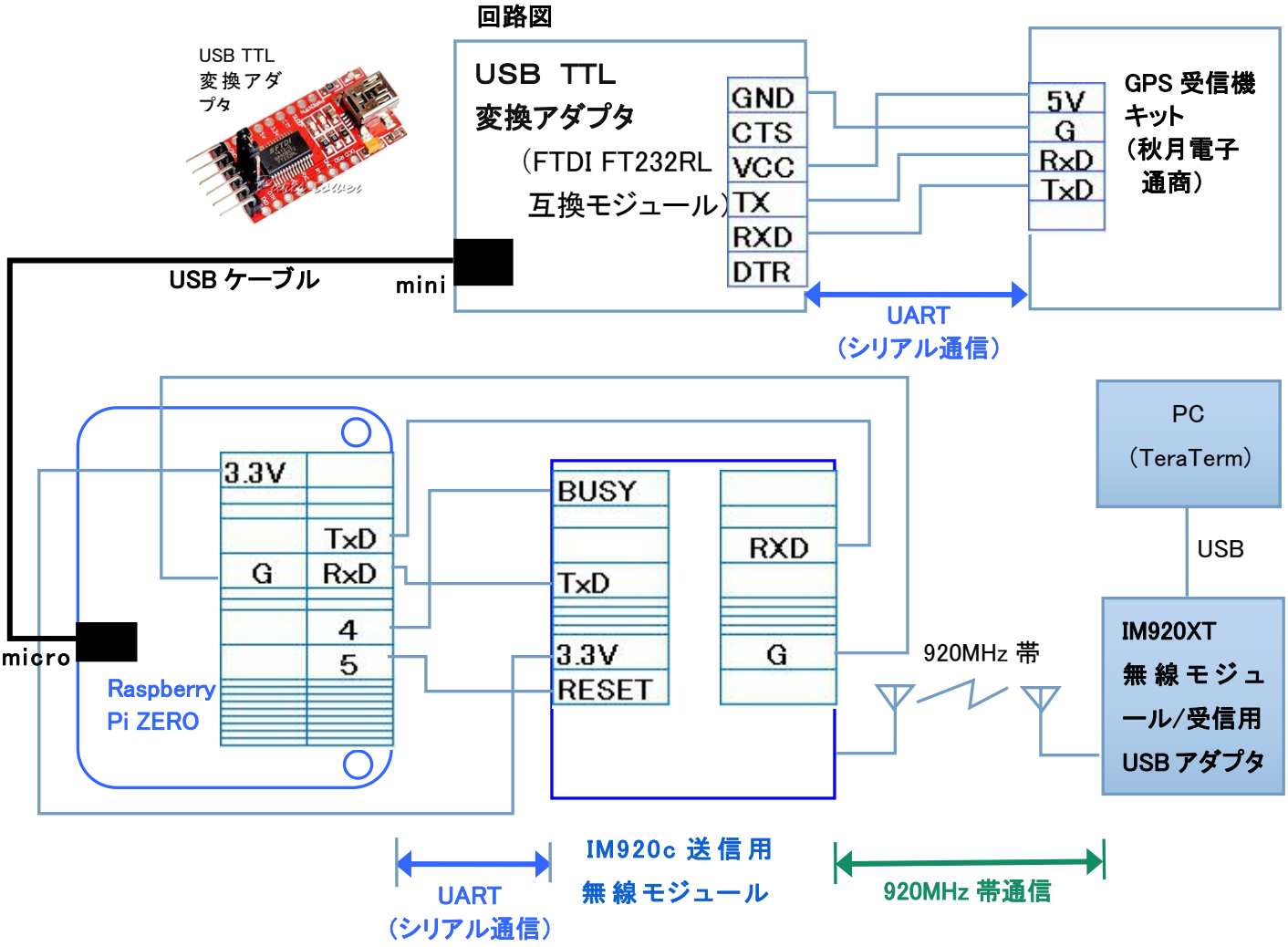
Wiring Pi で Raspberry Pi ZERO の周辺機器を制御する（その8）

2017.02.04 新海

GPSを 920Mhz 帯無線で飛ばす（2つのシリアル通信）

回路図



★今回の概要

＃6のGPSと ＃7の 920MHz 通信を一つにして、**GPS 計測値を 920MHz 通信で送信**します。
シリアル通信（UART）が2つになるので、**片方は USB をシリアル通信に変換**します。
この為のアダプターとして有名なものは **FTDI FT232RL** というモジュールを使ったものですが
互換品なら amazon で**数百円**で買えます。例:http://amzn.asia/bWUq1Vo 244 円
なお VCC は 3.3V と 5V をアダプタ上のジャンパスイッチで切り替えます。（基板上にプリントあり）

★プログラムのキーポイント

- USB をシリアル通信に変換したときの**デバイス名**は/dev/ttyUSB0 になります。
  また、シリアル通信が2つになるので、ファイルハンドラや Termios へのポインタも
  それぞれfd1、fd2 ttyparam1、ttyparam2 のように2つずつあります。
- このドキュメントではプログラムの中の **USB（GPS）側**を**赤文字、赤枠**で、**920MHz 側**を
  緑文字、緑枠で示しました。
- ロジックの追加としては、送信が 1 行単位なので、GPS データを順次配列 str[]に貯めて、
  データの最後のしるしである **CR＋LF が来たら送信**するようにしています。
  CR+LF と比較する必要があるため ループの最初で、**2文字先読み**をしています。

参考資料 IM920Hz がすぐわかる PDF，IM920 ソフトウェア取扱説明書 等
http://www.interplan.co.jp/support/solution/IM315/catalog/IM920_intro.pdf
http://www.interplan.co.jp/support/solution/IM315/manual/IM920_SW_manual.pdf
トランジスタ技術 2016 年 3 月号別冊付録「全ラズパイ対応！楽々I/O関数セット WiringPi 虎の巻」
USB TTL 変換アダプタの販売ページ（ピン配置を記載）http://amzn.asia/bWUq1Vo

920MH 帯で GPS データを送信するプログラム　gpssend

```
#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  */
int main(){
int i;
int fd1,fd2;                    /* UART(シリアル通信)のファイルポインタ  */

/*  通信設定を行う構造体 tarmiios の定義  */
struct termios ttyparam1,ttyparam2;
unsigned char str[255] ={"TXDA "};
/* UART(シリアル通信)のオープン  */
fd1=serialOpen("/dev/ttyUSB0",9600);          /* USB(GPS)側  */
fd2 =serialOpen("/dev/ttyAMA0",19200);        /* 920MHz 無線側  */
/* 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);
```

どちら側のシリアルポートか
という事以外は
通信設定の内容は#06,#07
と同じ

```
/*  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   */
wiringPiSetup();         /* wiringPi  のピン番号を使う Setup  */
pinMode(4,INPUT);        /* 4 番ピンを Busy の検知(入力)に使う  */
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  */
```

←#07と
同じ

```
serialFlush(fd1);                       /*  受信バッファのクリア  */
while('1')                              /*  無限ループ  */
{
   str[5]=serialGetchar(fd1);      /* CR+LF  かどうか調べるために  */
   str[6]=serialGetchar(fd1);      /*  2 文字先読みをする          */
   i=6;                                 /*  次の格納場所を 7 とするための設定  */
while (str[i-1]!=0x0d || str[i]!=0x0a) /* CR+LF  でないなら     */
   {                                    /*  以下を実行           */
        i++;                                        /*  格納場所を1つ先にする  */
        str[i]=serialGetchar(fd1);        /*  読んだ文字を格納場所に入れる  */
   }
      str[i+1]=0x00;                          /*  送信文字列の最後に 0x00 */
      while (digitalRead(4)){}   /* 4 番ピンが Busy でない限り  */
      serialPuts(fd2,str);   /* str が示す文字列をシリアル通信で送る  */
}
return 0;
}
```