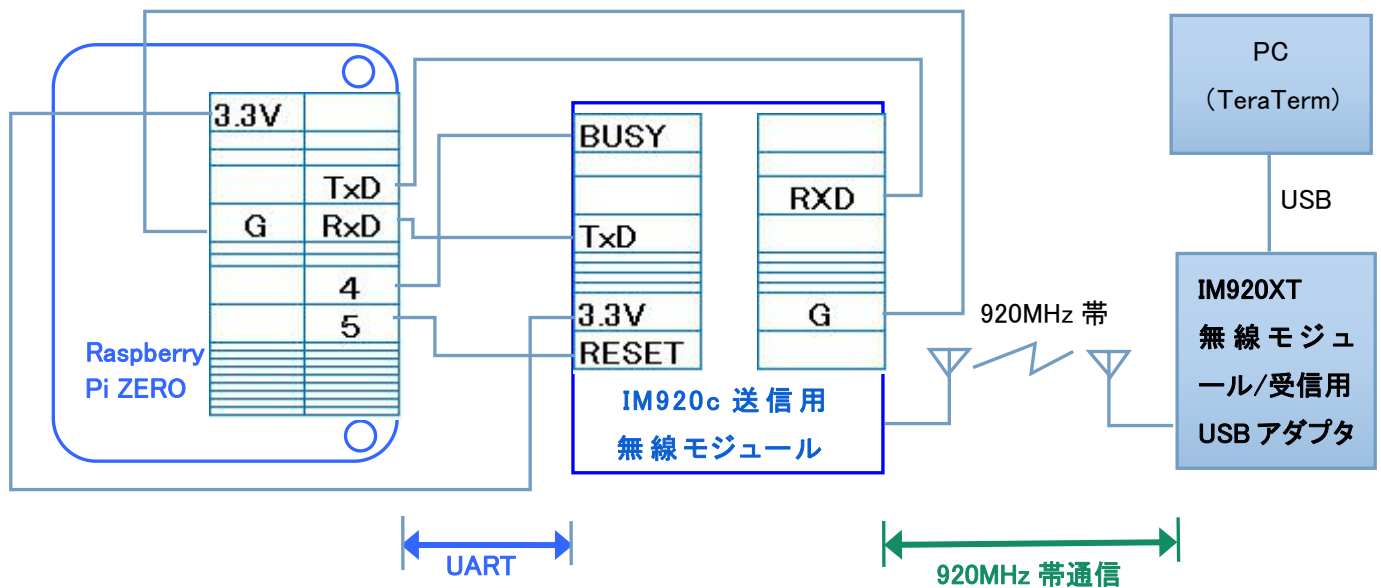


UART(シリアル通信)で 920MHz 帯無線通信

(UART: Universal Asynchronous Receiver Transmitter)

## 回路図



## ★今回のキーポイント:

無線モジュールと無線モジュールの間は **920MHz 帯通信**だが、RaspberryPiZERO と無線モジュールの間は **UART(シリアル通信)**。なので、**920MHz 帯通信のコマンドやデータ**はいったん **UART(シリアル通信)**に載せて無線モジュールに送る必要がある。

## ・920MH 帯通信について

920MH 帯通信は、免許が不要な特定小電力無線にもかかわらず、見通しで最大7kmという通信範囲の広さが特徴です。欠点としては通信速度が遅いこと、連続的な通信ができない事があり、多数のセンサーからの情報収集など、IoT向きを想定した通信です。

ここでは、秋月電子で販売されているインタープラン社の IM920 シリーズ無線機器を使用します。

## ・ Wiring Pi の関数(その8)

`serialPuts`(ファイルハンドラ, 文字列ポインタ)

文字列ポインタで示される文字列を 0x00 の手前まで UART(シリアル通信)で送信する。

## ・ 920MH 帯通信コマンドの例

**TXDA**: データ送信コマンド(可変長)

形式 TXDA ...データ... <CR><LF>

TXDA の後に一つ以上の空白を空け、それ以降、<CR><LF>が来るまでの間のデータを920MHz 帯で送信する。

## データの形式

**DCIO** (HEX 入出力モード) 0~9,A~Fの16進数でデータを表す。デフォルト

**ECIO** (キャラクタ入出力モード) <CR><LF> 以外の文字なら文字で送れる。

デフォルトではないので **ECIO コマンド**(ECIO<CR><LF>)で明に指定する必要がある。

## 受信形式

aa,bbbb,cc: 受信データ

**aa** : ノード番号 (ユーザ設定可 省略値 00)

**bbbb** : 送信モジュールの固有 ID(今回の送信モジュールは 5543)

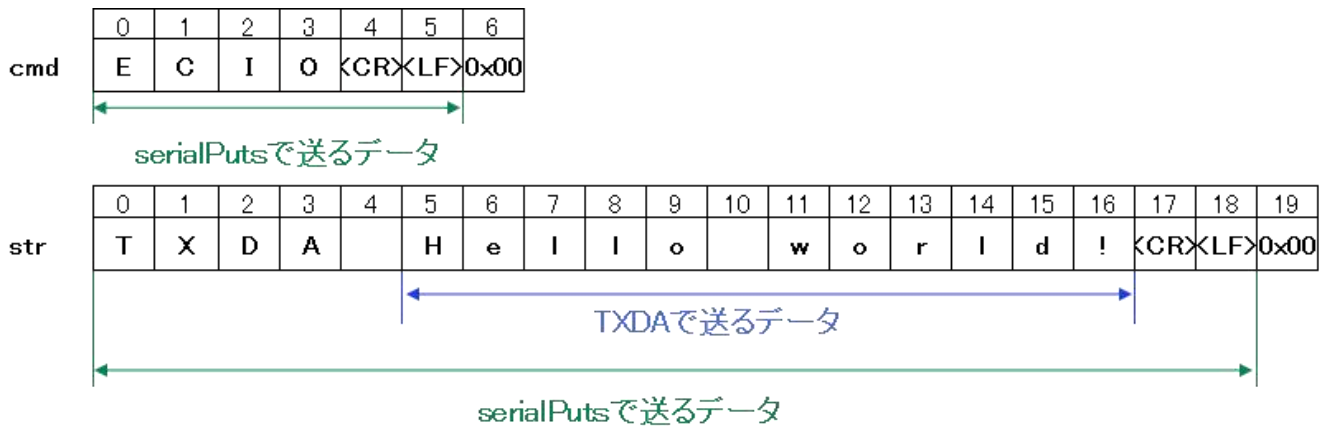
**cc** : データ受信時の RSSI(Received Signal Strength Indicator:受信強度) 値

例: TXDA hello World! ⇒ **00,5543,DD**:hello World!

## 参考資料① 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/catalog/IM920_intro.pdf)

[http://www.interplan.co.jp/support/solution/IM315/manual/IM920\\_SW\\_manual.pdf](http://www.interplan.co.jp/support/solution/IM315/manual/IM920_SW_manual.pdf)



### 920MH 帯で”Hello World!”を送信するプログラム send

```

/* 以下、↑までは通信速度以外前回と同じなので、前回は参照のこと */

#include<wiringPi.h>
#include<wiringSerial.h>
#include<termios.h>
int main()
{
int fd;
struct termios ttyparam;
fd=serialOpen("/dev/ttyAMA0",19200); /* 通信速度 19200bps */
togetattr(fd,&ttyparam);
ttyparam.c_iflag &=~IXON;
ttyparam.c_iflag &=~IXOFF;
ttyparam.c_cflag &=~CSIZE;
ttyparam.c_cflag |=CS8;
tcsetattr(fd,TCSANOW,&ttyparam);

/* ↑以上、通信速度(19200bps)以外前回と同じ。 */

/* 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 */

/*IM910 のコマンド TXDA で ”Hello world!” を送る為の文字列 */
char str[20]={”TXDA Hello world!”}; /* コマンド TXDA と本文 */
str[17]=0x0d; /* <CR> IM910 コマンドの終わり(CR+LF) */
str[18]=0x0a; /* <LF> IM910 コマンドの終わり(CR+LF) */
str[19]=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(fd, cmd); /* cmd が示す文字列をシリアル通信で送る */
}

while (digitalRead(4)){ /* 4 番ピンが Busy でない限り */
serialPuts(fd, str); /* str が示す文字列をシリアル通信で送る */
}

return 0;
}

```