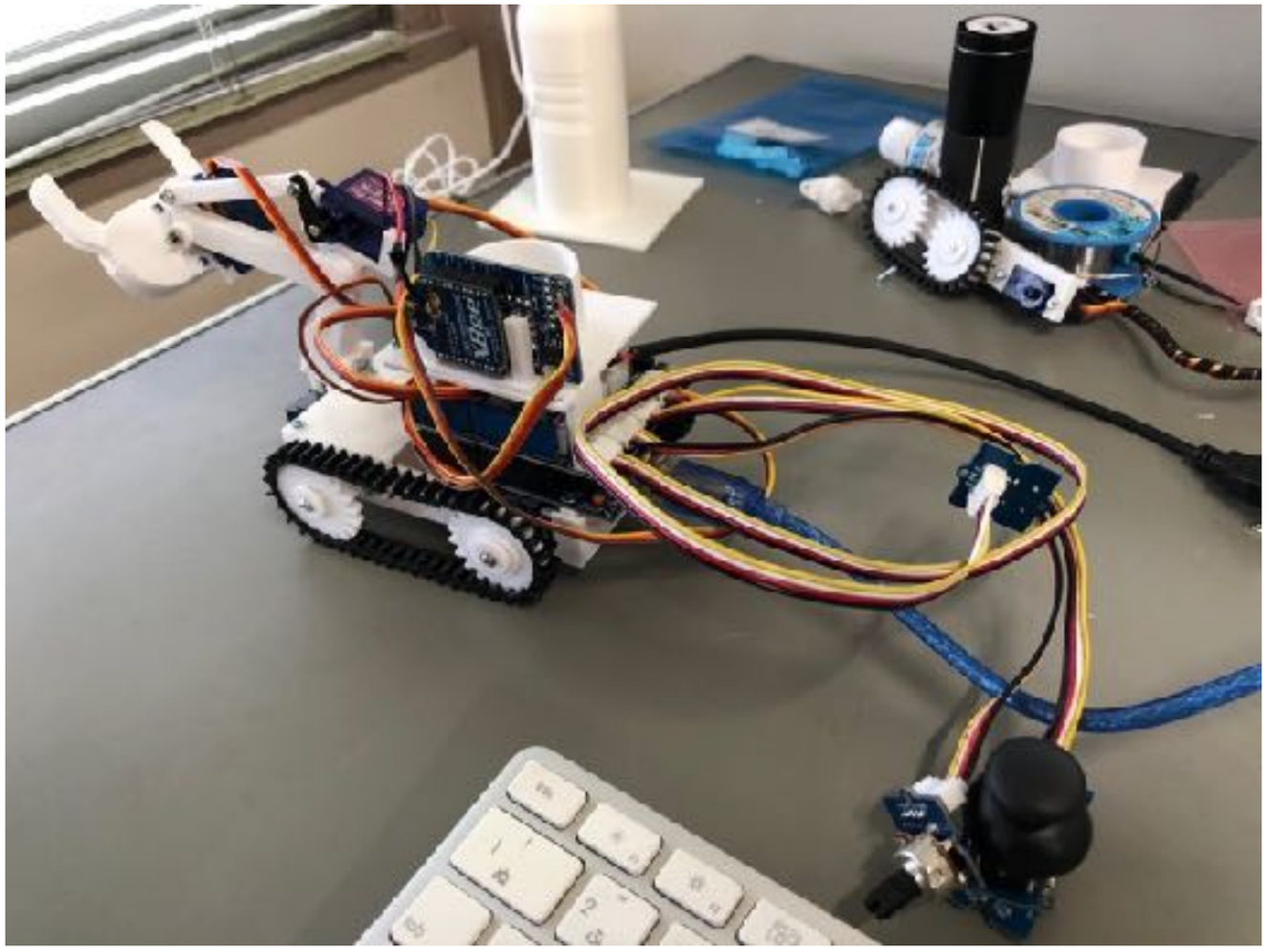
クローラーロボットの無線化

X-Beeを使って無線化してみよう！

- **今のクローラーロボットの操作は？**

  ジョイスティックとボタンで操作をしている。電源はパソコンからUSBケーブルで
  引っ張っているので自由に動けない。



- **無線化を行うのには？**

  ジョイスティックやボタンを操作した時の状態を、送信機からクローラーロボットに送り、
  操作を再現する仕組みが必要。
  また、自由に動くためにはバッテリーで動作できるように改造が必要。

・arduinoで無線通信を簡単に行えるXBeeを使ってみよう

Xbeeは無線通信を行うモジュールで、マイコンの有線通信
（シリアル通信）をそのまま無線に置き換えることができる。

シリアル通信は、これまでパソコンとarduinoをUSBケーブルで
繋いで、虫眼鏡のモニタなどで文字表示をさせてきたが、それが
「シリアル通信」であった。

クローラロボットとコントローラーのarduinoのUARTという
コネクタにそれぞれXBeeを繋ぐことで無線通信をすることが
できる。

注意点

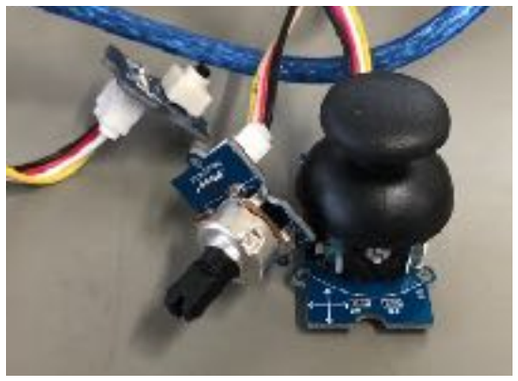
arduinoのUARTはパソコンからプログラムを書き込む時にも使用している。
そのため、Xbeeを繋いだままだと、パソコンとarduinoとXbeeの３つが繋がった状態になり
動作不具合を起こす場合がある。

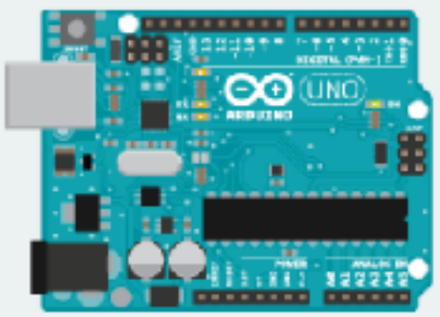
プログラムの書き込みをパソコンから行うときにはUARTに刺したXBeeを外してから行うこと。

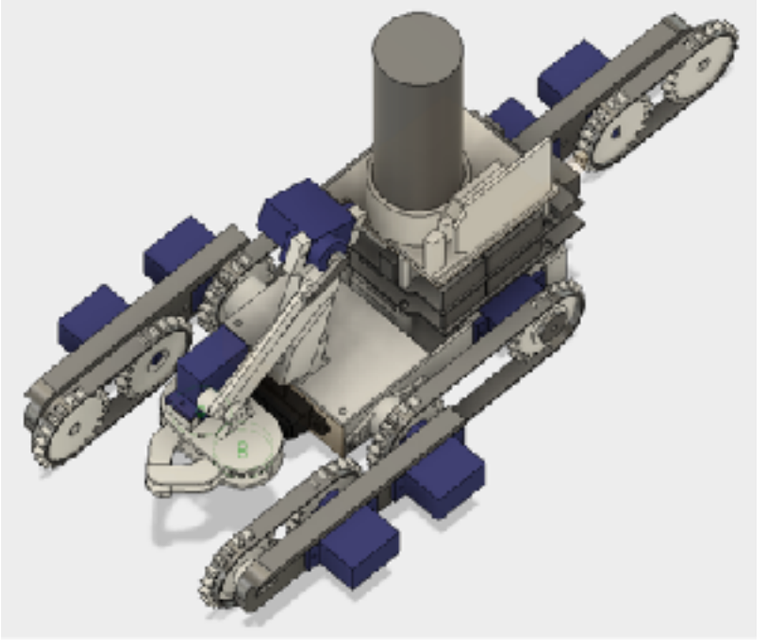
・ジョイスティックやスイッチの状態を無線で送るためには？

テレビやラジオのように情報を変換して相手に伝えて、その情報を元に戻す仕組みが必要。

スタジオの映像を　→ 映像と音声信号にして → テレビ局が放送　→テレビが受信して映像と音声を出す



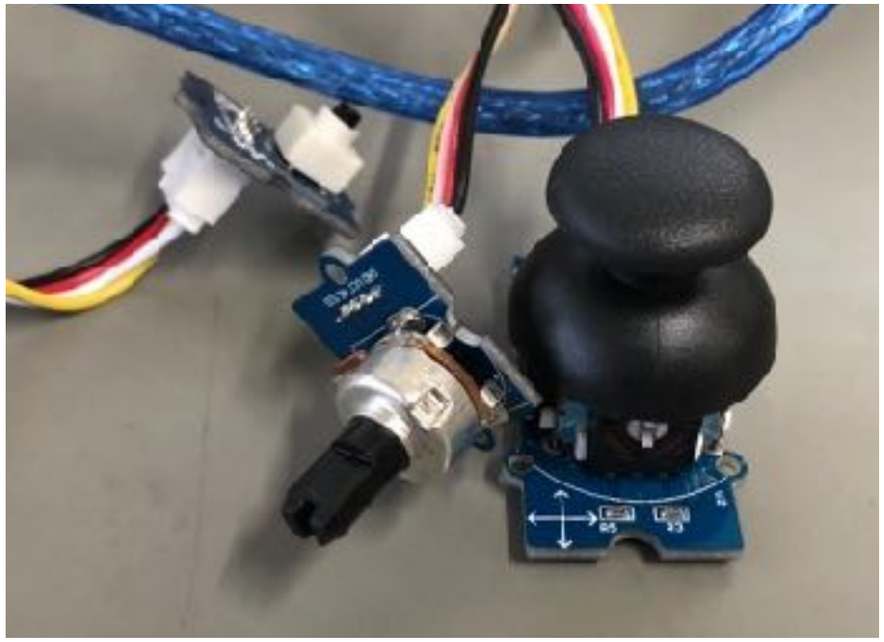




入力装置からの信号を　→ データーに変換して　→ 送信　→ 受信して →クローラーロボットで再現

・どんなふうに変換して送るのか？

ボリュームやスイッチはanalogReadやdigitalReadで数値として読み出すことができる。
その値を「データ列」として送信し、受信したものを数値に変えてサーボモーターや変数に代入する。



０～1023やHIGHかLOWの信号を、サーボモーターの扱いやすい形（-90～+90）などにする。
arduinoのchar型は0~255まで扱えるので、これを利用してみる。

中央値を122として、そこから-90した値（32)から＋90した値（212)までを扱えるようにする。

例えば、12を送るのなら、送信元は122+12=134を受信側に送る。
受信側は受け取った値から122を引くと、134-122=12を受け取れる。

どうして、中央値を122にしたの？

コンピューターのコードは0~31を「制御コード」として特別に扱う。
文字として使えるのが32からなので、32+90=122にした。

・データの終わりをどのように認識するか？

ロボット１つのデータだけを送るなら「122（中央値）」を１つ送るだけでいい。
しかし、３個の場合は「122 122 122」と決められるが、ロボットを拡張すると幾つまで
という決め方が難しい。（都度、プログラムを直さないといけないため）

そこで、受信したデータに「終わり」のコードが出てきたらそこまでが受信しなければ
ならないデータと認識するようにした。

・３個の場合
　122 122 122 終わりのコード
・５個の場合
　122 122 122 122 122 終わりのコード

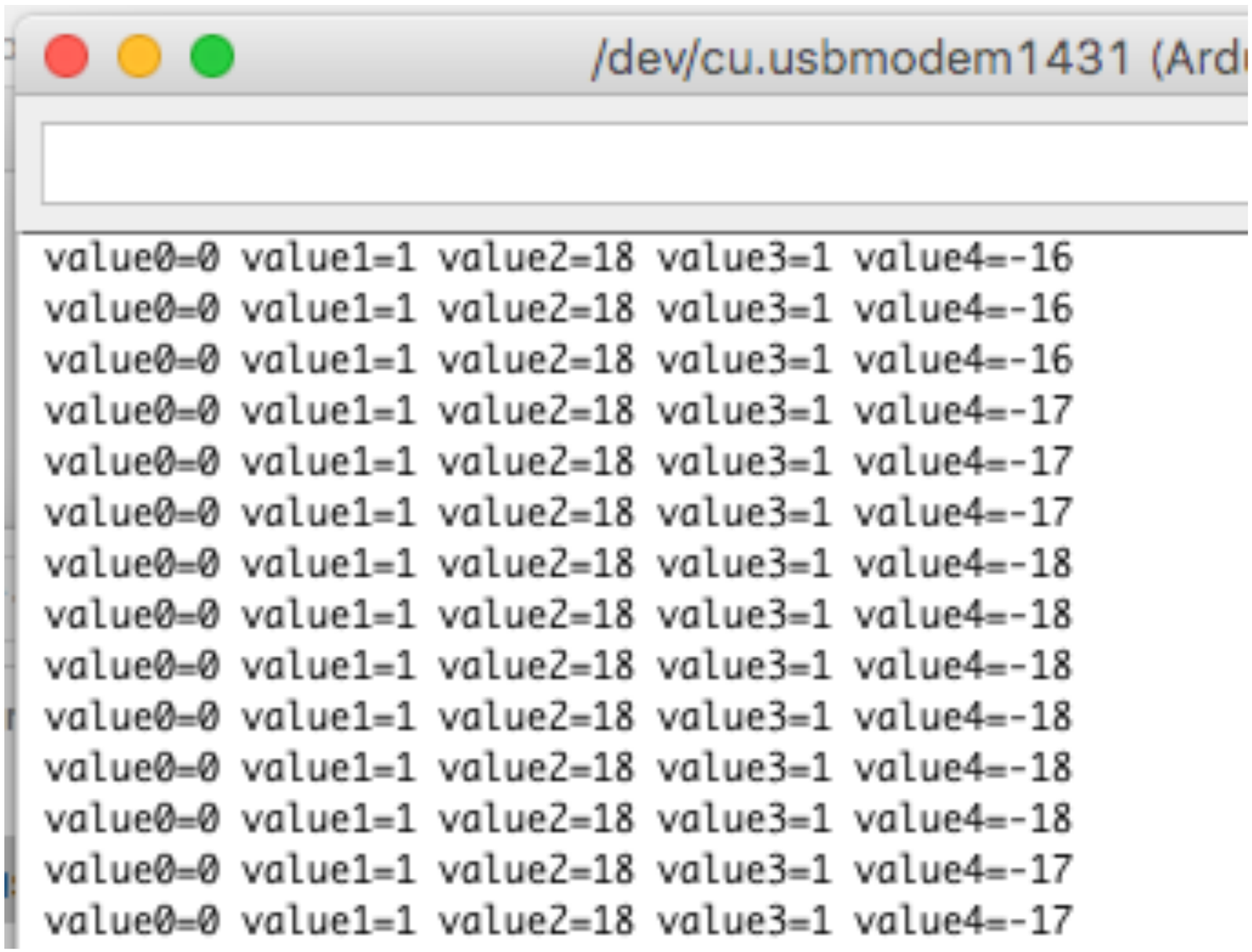
・終わりコードを何にするか？

ポピュラーなのはCR（１３）やLF（１０）。
これはタイプライターなどで次の行を指定するコードがCR（キャリッジリターン）や、
先頭桁に移動するコードがLF（ラインフィード）だったため。
今回は、CRを使うことにした。

・無線による通信を実験してみよう

Fig1をクローラーロボットに、tank-controller-miniを送信用arduinoに入れてみよう。
送信用arduinoのA1にジョイスティック、A3にボリュームを、D2にプッシュスイッチを繋いで、両方のarduinoのUARTにXBee繋いでみよう。

そして、虫眼鏡アイコンをクリックして、ジョイスティックやボリューム、スイッチを長押しして操作してどのように数値が変わるか見てみよう！



・tank-controller-miniは何をしているの？

クローラーロボットのコントロールに特化して、１つのジョイスティックと２つのボリューム、１つのプッシュスイッチに対応した制御値を送信しています。

移動用のジョイスティックの値を受けて、value0とvalue1に駆動用サーボモータの動作値をアーム用ボリュームの値を受けて-90~+90までの値を、プッシュスイッチの長押しでHIGHとLOWを切り替えるデータを送ります。

プログラムソースは時間をかければ、これまでのセミナー内容で充分読み解けると思います。

・fig-1は何をしているの？

tank-controller-miniのプログラムから送られてきた情報を、専用の配列変数「value[]」に格納して、その値を表示させています。

先頭の#includeでSerialModule.hというクローラーロボット用の通信ライブラリを呼び出すことで、簡単に無線通信で値を受け取ることができます。

・fig-1を改造してクローラーロボットのアームをボリュームで動かしてみよう！

クローラーロボットのD4にアームのサーボモータを繋いで、fig1を以下のように書き換えてみよう。

```
#include <Servo.h>
#include "SerialModule.h"//通信モジュールを追加

const int ARM_SERVO_PIN=4;
const int ARM_ZERO_POS=90;

Servo arm_servo;
//----------------------初期化-----------------------------
void setup() {
  Serial.begin(9600);
  arm_servo.attach(ARM_SERVO_PIN);

  //通信用変数初期化
  write_ptr=0;
  read_ptr=0;
  command_ptr=0;
}
//------------------メイン変数-----------------------------------
void loop() {

  Serial_check();//通信チェック
  arm_servo.write(ARM_ZERO_POS+value[2]);
}
```

・クローラーロボットにtank-wirelessを書き込んでみよう

クローラーロボットのD2に駆動用サーボの左、D3に駆動用サーボの右、Ｄ４にアームのサーボ、
D5にハンドのサーボを繋いでから書き込むこと。
その時にパソコンのUSBからの給電では電力不足になるのでモバイルバッテリーをI2Cの
コネクタに刺しておこう。

・クローラーロボットの調整をしてみよう

直線距離を長く走らせると前進後退で左右のどちらかに曲がっていくと思います。
それは駆動系のサーボや組み付けの個体差、抵抗のばらつきによるものです。
tank-controllerの以下の数値を変更してまっすぐ走るように調整をしよう。

```
const int MOTOR_L_FWD=25;//左サーボ用前進時固定値
const int MOTOR_R_FWD=40;//右サーボ用前進時固定値
const int MOTOR_L_BACK=28;//左サーボ用後退時固定値
const int MOTOR_R_BACK=40;//右サーボ用後退時固定値
```

上の数値は、前進と後退の時に、左右のサーボモーターが停止状態からどのくらいの速度で
モータを回すかの値になります。
数値が大きいと速く、小さいと遅く回ります。

・クローラーロボットに自由に走らせてみよう

バッテリーベースを取り付けて、無線操縦でクローラーロボットを操作してみよう。
取り外したロケットユニットは、ハンドで掴むにはちょうどいいサイズになっています。

持ち上げて移動したりしてロボットの操作に慣れてみよう。

・クローラーロボットでゲームをしてみよう

セミナーの参加者を２グループに分けて、サッカーをしてみよう。

今回のセミナーはこれで終了です。

クローラーロボットの操作について感想を交換してみよう