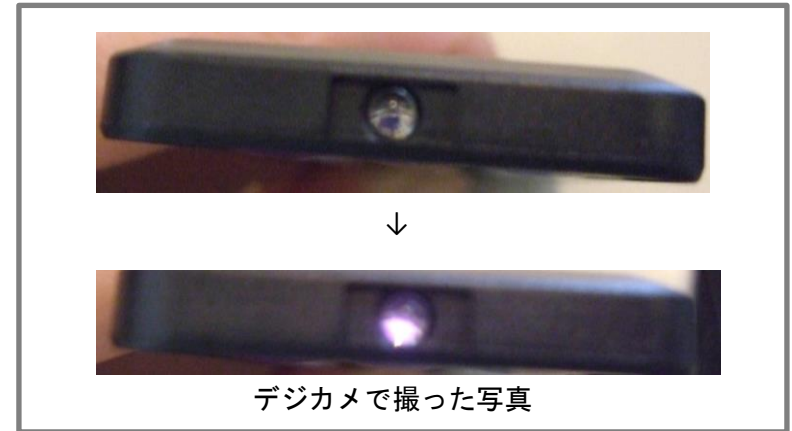
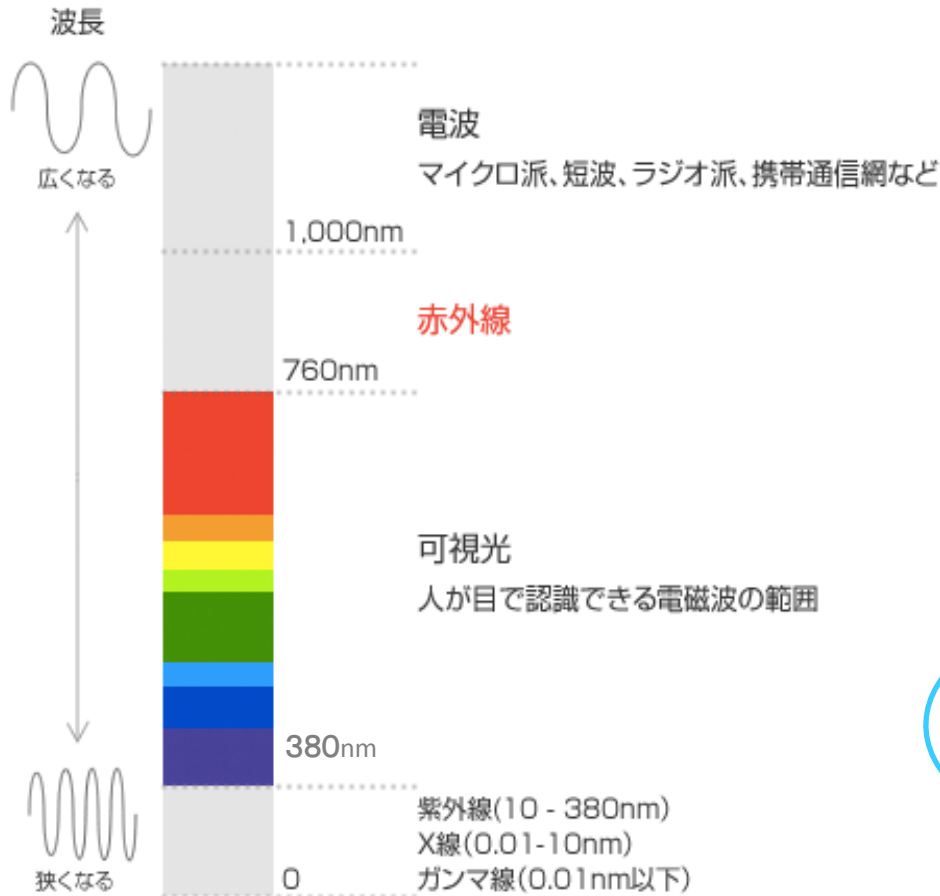


レベル5 赤外線リモコンで操作できるようにする

赤外線 = IR (infrared)とは？

赤外線は、可視光(ヒトの目で見ることができる光) 赤色より波長が長く(周波数が低い)、電波より波長の短い電磁波のことです。ヒトの目では見ることができない光ですが、デジカメ、スマホなどのカメラを使うと見ることができます。また、赤外線は太陽光にも含まれているので、このリモコンでは発光する周波数を38KHzと設定し、誤作動しにくくしています。屋外で使うときは注意が必要です。



<ついでの豆知識>

『どうして空は青いの?』

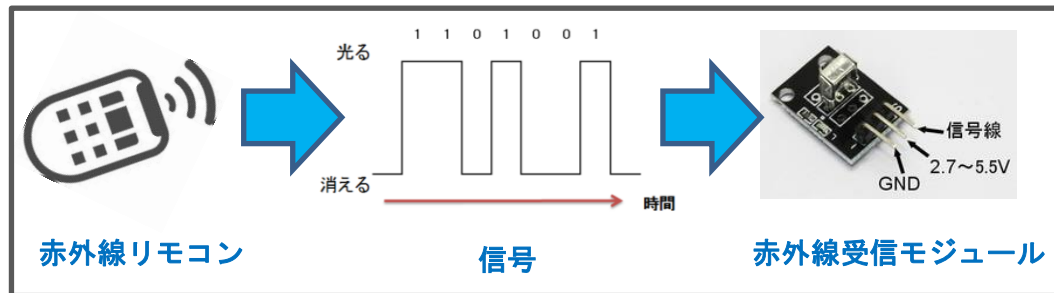
太陽の光にはいろいろな色が含まれています。太陽の光が空気(の分子)にあたると、青い方の色から多く散らばります。そのため空は青く見えます。夕方になると、太陽が遠くなり、途中で青い色はなくなってしまうので、太陽は赤く見えるのです。

では、「どうして雲は白いの?」と思ったするどいキミ! ぜひ考えてから調べてみようね。

赤外線リモコンのデータを送受信する

リモコンのボタンを押すと、赤外線的光がチカチカ点滅(てんめつ)します。この光ったり消えたりする時間が、ボタンによって違います。それを 受信モジュールが "1" と "0" の信号として受け取ります。

この例の場合は「1101001」です。
これは2進法と言われます。



Q. 2進法って何？

A. 時計の60分は1時間と言いますよね。それは60進法です。同じように考えたらわかりますね。

<自分のリモコンの信号を確認する方法>

1. Arduinoに「level5_IR」を書き込みます。
2. USBケーブルをつないだまま画面右上のボタンを押してください。
3. リモコンの 上↓→←OK ボタンを順に押すと、**図1**のように表示されます。
4. あれ？ “1”と“0” ではないですね。

これは、今回のプログラムは `Serial.println(ans.HEX);` と書いてあって、HEX = 16進数 で表示されるようにしているからです。
ちなみに、16進数の場合は、 `case 0xb9: // 上` のように、値としては前に “0x” をつける必要があります。

図2、**図3**はそれぞれ BIN = 2進数 DEC = 10進数での表示です。

図2 2進法



図3 10進法

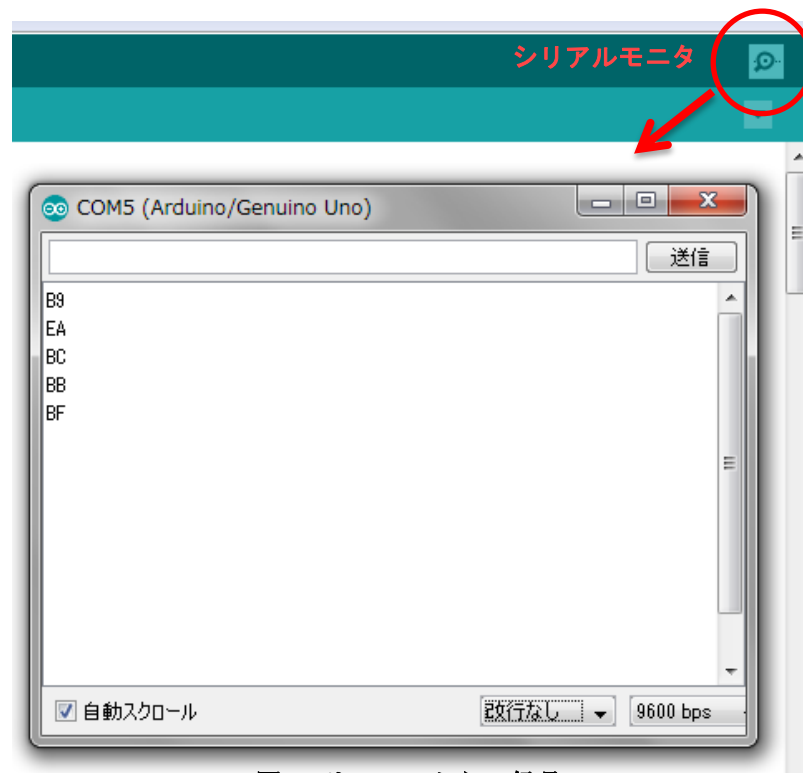
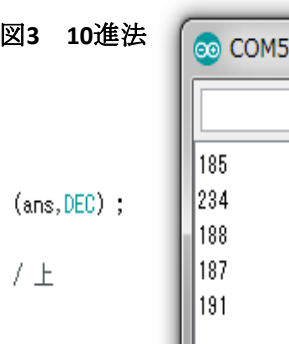
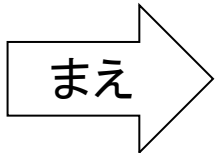
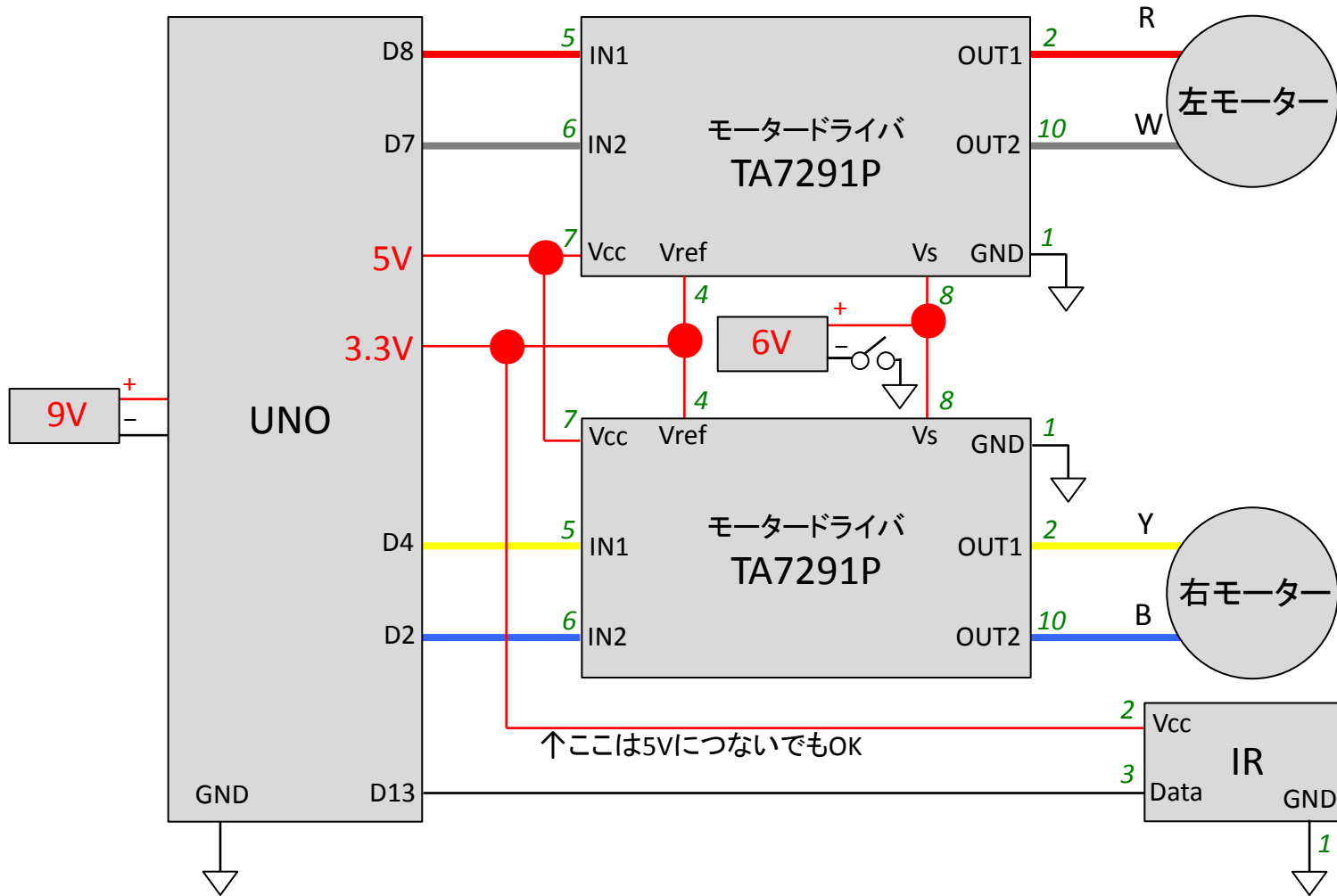


図1 リモコンからの信号

レベル5 赤外線リモコン利用の回路図

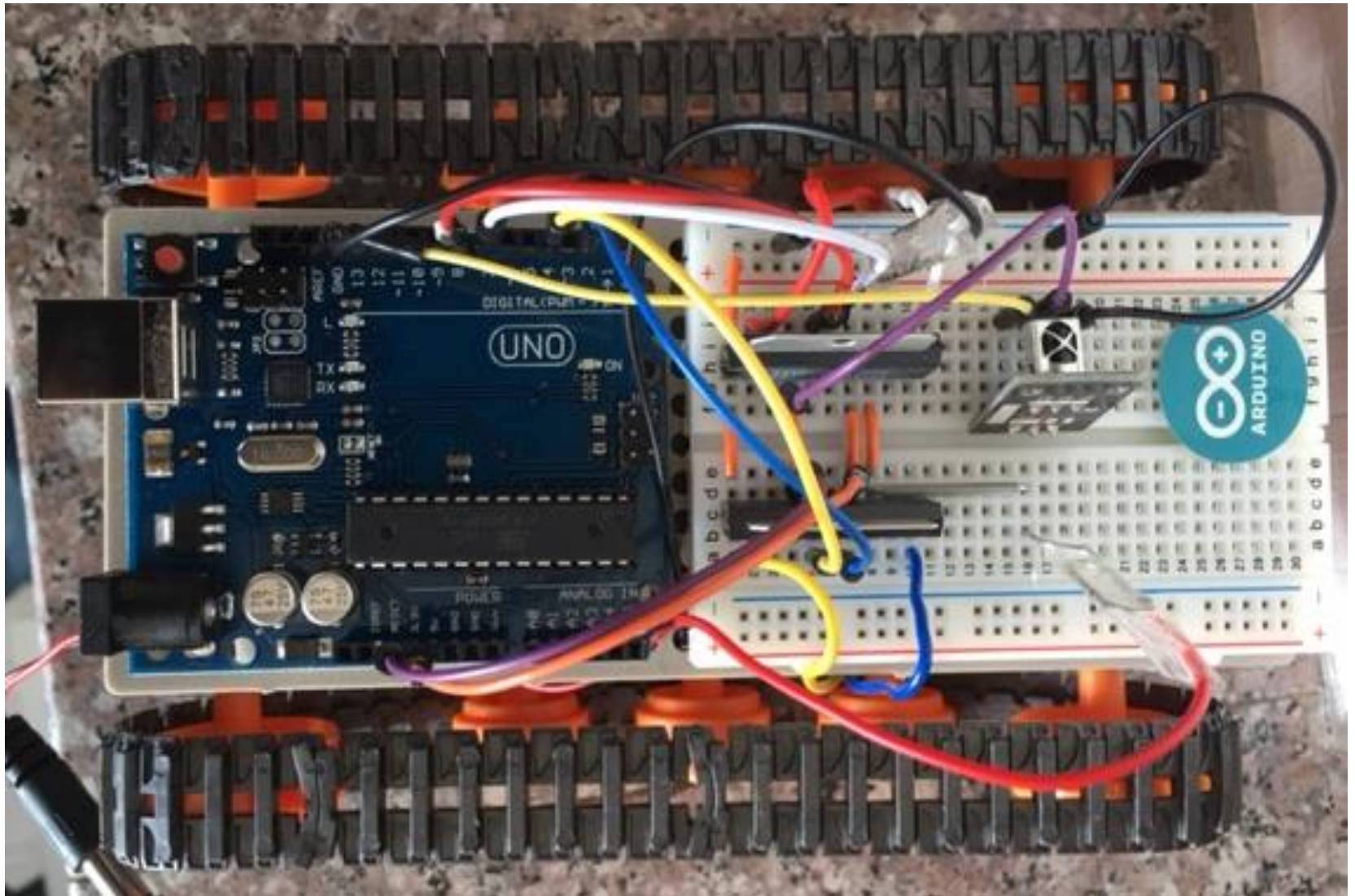
- ・実験中は、ショート(=電池のプラスとマイナスを直接つなぐこと)をしない。
- ・実験中は目を離さない。
- ・使い終わったら電池を抜く。
- ・熱くなったり、変な匂いがしたら実験を中断する。
- ・USBコードと、9Vの電池からのジャックは同時に繋がらない



IRの番号は黒い部品を見て左から123です。

レベル5 赤外線リモコンでの完成例

※あくまでも例ですので、この写真と同じ配線になるとは限りません



ここまで頑張ったね！

クローラー製作を通してみんなに伝えたかったことがあります。

**うまくいかないところからが、(エンジニアとして)始まり。
ここで問題に向き合えるか、向き合えないかが大切。
安易に答えを求めてはいけません。**

※安易に: 努力しないで楽に

今、AI(人工知能)やIoT(モノのインターネット)でどんどん社会が変わりつつあります。
「無線で操縦できるクローラー」というのはその入り口です。

また、福島原発では人が入れないところでもロボットが活躍しています。このように世の中、
人のために役に立つクローラーの使い方も考えてもらえたらうれしいです。

何かいいアイデアが思いついたら、一緒に作り上げていきましょう！
また遊びに来てくださいね！

赤外線リモコンで操作するプログラム

```
//IR
#define IR_PIN 13 // 赤外線受信モジュール接続ピン番号
#define DATA_POINT 4 // 受信したデータから読取る内容のデータ位置

//右車輪
const int p1 = 2;//Blue
const int p2 = 4;//Yellow
//左車輪
const int p3 = 7;//White
const int p4 = 8;//Red

void front() //前進
{
digitalWrite(p1,HIGH);
digitalWrite(p2,LOW);
digitalWrite(p3,HIGH);
digitalWrite(p4,LOW);
}
void back() //後退
{
digitalWrite(p1,LOW);
digitalWrite(p2,HIGH);
digitalWrite(p3,LOW);
digitalWrite(p4,HIGH);
}
void right() //右旋回
{
digitalWrite(p1,LOW);
digitalWrite(p2,HIGH);
digitalWrite(p3,HIGH);
digitalWrite(p4,LOW);
}
void left() //左旋回
{
digitalWrite(p1,HIGH);
digitalWrite(p2,LOW);
digitalWrite(p3,LOW);
digitalWrite(p4,HIGH);
}
void brake() //停止
{
digitalWrite(p1,HIGH);
digitalWrite(p2,HIGH);
digitalWrite(p3,HIGH);
digitalWrite(p4,HIGH);
}
```

```
// 赤外線リモコンのデータを受信する処理関数
int IRrecv()
{
unsigned long t ;
int i , j ;
int cnt , ans ;
char IRbit[64] ;

ans = 0 ;
t = 0 ;
if (digitalRead(IR_PIN) == LOW) {
// リーダ部のチェックを行う
t = micros() ; // 現在の時刻(us)を得る
while (digitalRead(IR_PIN) == LOW) ; // HIGH(ON)になるまで待つ
t = micros() - t ; // LOW(OFF)の部分をはかる
}
// リーダ部有りなら処理する(3.4ms以上のLOWにて判断する)
if (t >= 3400) {
i = 0 ;
while(digitalRead(IR_PIN) == HIGH) ; // ここまでがリーダ部(ON部分)読み飛ばす
// データ部の読み込み
while (1) {
while(digitalRead(IR_PIN) == LOW) ; // OFF部分は読み飛ばす
t = micros() ;
cnt = 0 ;
while(digitalRead(IR_PIN) == HIGH) { // LOW(OFF)になるまで待つ
delayMicroseconds(10) ;
cnt++ ;
if (cnt >= 1200) break ; // 12ms以上HIGHのままなら中断
}
t = micros() - t ;
if (t >= 10000) break ; // ストップデータ
if (t >= 1000) IRbit[i] = (char)0x31 ; // ON部分が長い
else IRbit[i] = (char)0x30 ; // ON部分が短い
i++ ;
}
// データ有りなら指定位置のデータを取り出す
if (i != 0) {
i = (DATA_POINT-1) * 8 ;
for (j=0 ; j < 8 ; j++) {
if (IRbit[i+j] == 0x31) bitSet(ans,j) ;
}
}
}
return( ans ) ;
}
```

次のページへ続く

```
void setup()
{
  pinMode(p1, OUTPUT);
  pinMode(p2, OUTPUT);
  pinMode(p3, OUTPUT);
  pinMode(p4, OUTPUT);

  ///IR
  Serial.begin(9600); // パソコン(ArduinoIDE)とシリアル通信の準備を行う
  pinMode(IR_PIN, INPUT_PULLUP); // 赤外線受信モジュールに接続ピンをデジタル
  入力に設定
}
```



```
void loop()
{
  int ans;
  ans = IRecv(); // 赤外線リモコンのデータを受信する

  if (ans != 0) {
    Serial.println(ans, HEX);
    switch(ans) {
      case 0xb9: // 上
        //前進
        front();
        delay(2000);
        brake();
        break;

      case 0xea: // 下
        //後退
        back();
        delay(2000);
        brake();
        break;

      case 0xbc: // 右
        //右旋回
        right();
        delay(500);
        brake();
        break;

      case 0xbb: // 左
        //左旋回
        left();
        delay(500);
        brake();
        break;

      case 0xbf: // OK
        //停止
        brake();
        break;

    }

  } else {
    brake();
  }
}
```