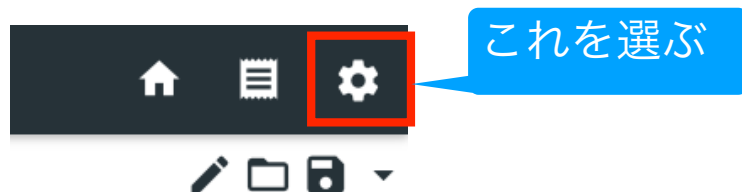


赤外線リモコンで クローラカーを動かす

LEVEL5をBlocklyDuinoで行う場合

(内容はLEVEL5のArduinoのプログラムと一部変わっています)

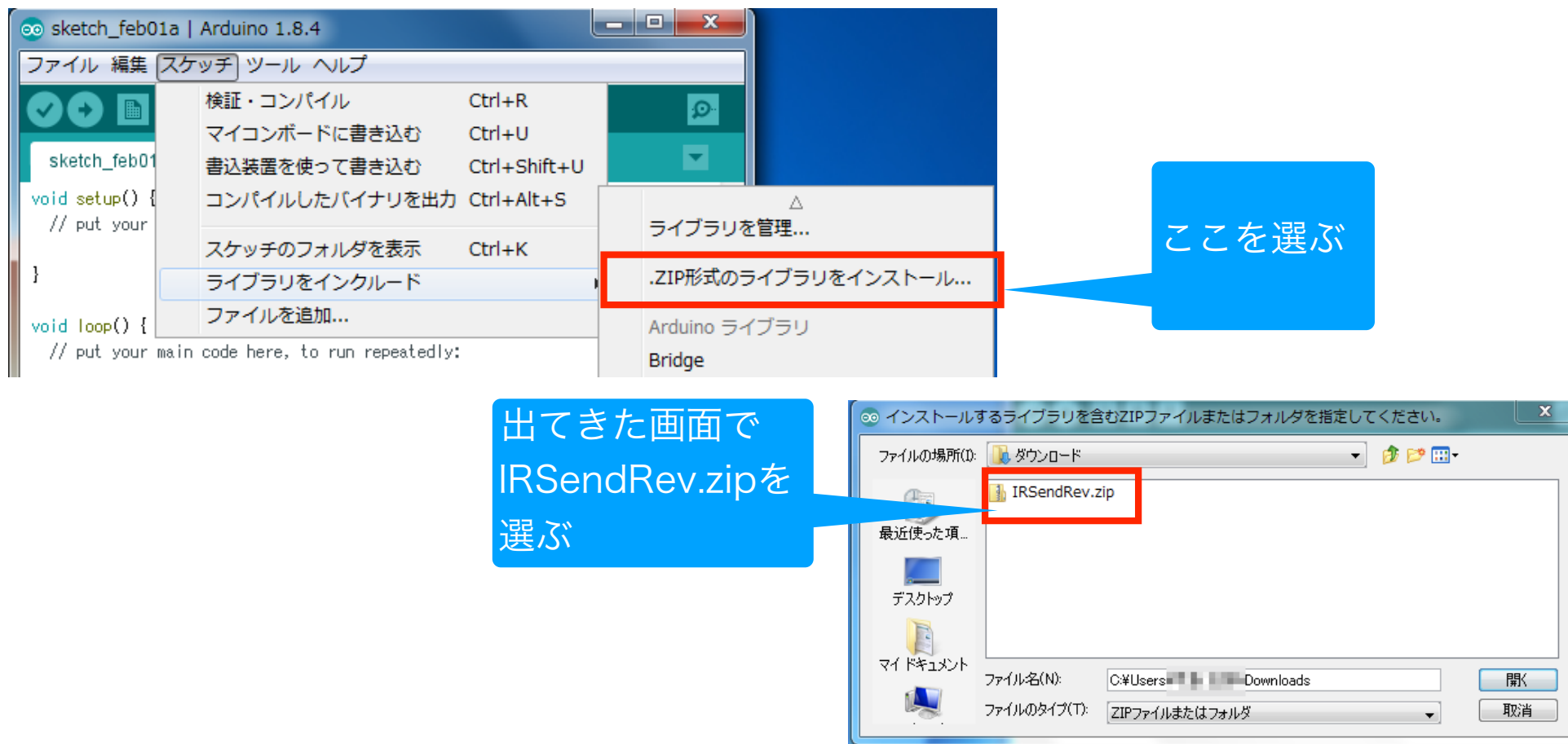
赤外線受信ブロックを追加する



赤外線送受信ライブラリをインストールする

- ① <https://github.com/Seeed-Studio/IRSendRev> から
IRSendRev-mater.zipをダウンロードして、IRSendRev.zipに名前を変える。
※今回はダウンロードしてあるファイルを渡します。

- ② Arduino IDEでダウンロードしたライブラリをインストールします。



The image shows two screenshots from the Arduino IDE. The top screenshot shows the 'Library Manager' menu with the option '.ZIP形式のライブラリをインストール...' highlighted in red. A blue callout box points to this option with the text 'ここを選ぶ'. The bottom screenshot shows a file selection dialog with 'Downloads' as the location and 'IRSendRev.zip' selected. A blue callout box points to the file with the text '出てきた画面で IRSendRev.zipを選ぶ'.

ここを選ぶ

出てきた画面で
IRSendRev.zipを
選ぶ

赤外線リモコンの信号値を確認する

setup()へ記述

赤外線受信 初期化 ピン 3

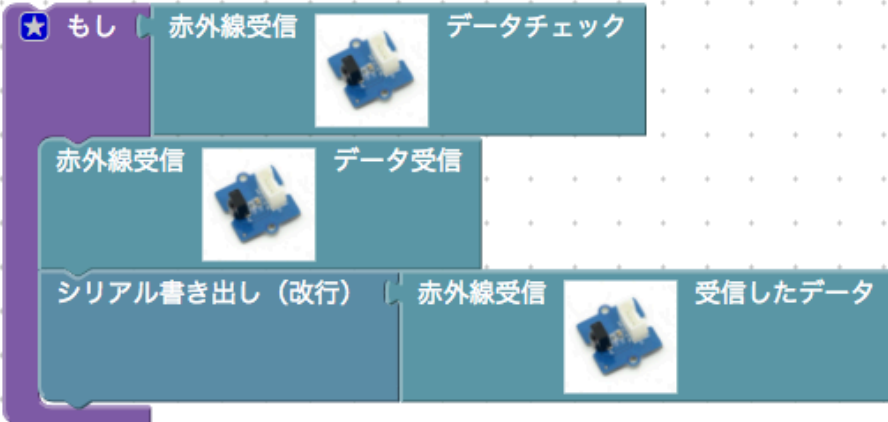


ピン3にしIRレシーバーもArduinoの3ピンに接続する

もし 赤外線受信 データチェック

赤外線受信 データ受信

シリアル書き出し (改行) 赤外線受信 受信したデータ



Arduinoのコード

```
#include <IRSendRev.h>

unsigned char dta[20];

void setup()
{
  IR.Init(3);

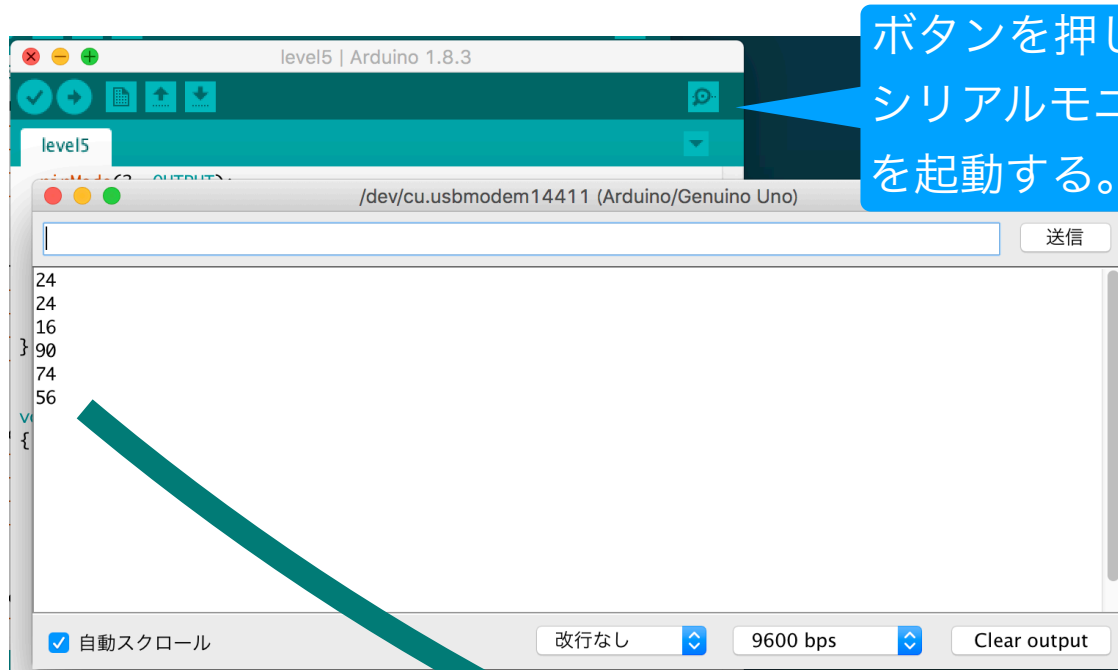
  Serial.begin(9600);
}

void loop()
{
  if (IR.IsDta()) {
    IR.Recv(dta);
    Serial.println(dta[6]);
  }
}
```

dat[6]をdta[8]に替える

リモコンのボタンを押してArudinoのシリアルモニタで数字を確認してメモしてください。

赤外線リモコンの信号値を確認する



ボタンを押して
シリアルモニタ
を起動する。

リモコンのボタンの値をメモする

ボタン	値
上	
下	
右	
左	
OK	

クローラカーをリモコンで動かす

setup0へ記述

赤外線受信 初期化 ピン 3

関数名 front

- デジタル出力 ピン 2 状態 HIGH
- デジタル出力 ピン 4 状態 LOW
- デジタル出力 ピン 7 状態 HIGH
- デジタル出力 ピン 8 状態 LOW

関数名 back

- デジタル出力 ピン 2 状態 LOW
- デジタル出力 ピン 4 状態 HIGH
- デジタル出力 ピン 7 状態 LOW
- デジタル出力 ピン 8 状態 HIGH

関数名 left

- デジタル出力 ピン 2 状態 HIGH
- デジタル出力 ピン 4 状態 LOW
- デジタル出力 ピン 7 状態 LOW
- デジタル出力 ピン 8 状態 HIGH

関数名 right

- デジタル出力 ピン 2 状態 LOW
- デジタル出力 ピン 4 状態 HIGH
- デジタル出力 ピン 7 状態 HIGH
- デジタル出力 ピン 8 状態 LOW

関数名 brake

- デジタル出力 ピン 2 状態 LOW
- デジタル出力 ピン 4 状態 LOW
- デジタル出力 ピン 7 状態 LOW
- デジタル出力 ピン 8 状態 LOW

もし 赤外線受信 データチェック

赤外線受信 データ受信

シリアル書き出し (改行) 赤外線受信 受信したデータ

もし 赤外線受信 受信したデータ = 24

front

もし 赤外線受信 受信したデータ = 74

back

もし 赤外線受信 受信したデータ = 90

left

もし 赤外線受信 受信したデータ = 16

right

もし 赤外線受信 受信したデータ = 56

brake

リモコンから受信した値に合わせて動かす

frontなどの関数は同じなので、LELVE3で作ったコードをコピーしてから作ってください。

調べたボタンの値を条件にしてどの数値だったら何の処理をするかプログラムしてください。

ボタン	値	処理
上		前進
下		後進
左		左回転
右		右回転
OK		ブレーキ

Arduino言語に変換されたコード

```
#include <IRSendRev.h>

void front() {
  digitalWrite(2, HIGH);
  digitalWrite(4, LOW);
  digitalWrite(7, HIGH);
  digitalWrite(8, LOW);
}

unsigned char dta[20];

void back() {
  digitalWrite(2, LOW);
  digitalWrite(4, HIGH);
  digitalWrite(7, LOW);
  digitalWrite(8, HIGH);
}

void left() {
  digitalWrite(2, HIGH);
  digitalWrite(4, LOW);
  digitalWrite(7, LOW);
  digitalWrite(8, HIGH);
}

void right() {
  digitalWrite(2, LOW);
  digitalWrite(4, HIGH);
  digitalWrite(7, HIGH);
  digitalWrite(8, LOW);
}

void brake() {
  digitalWrite(2, LOW);
  digitalWrite(4, LOW);
  digitalWrite(7, LOW);
  digitalWrite(8, LOW);
}
```

つづき

```
void setup()
{
  IR.Init(3);

  pinMode(2, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  if (IR.IsDta()) {
    IR.Recv(dta);
    Serial.println(dta[6]);
    if (dta[6] == 24) {
      front();
    }
    if (dta[6] == 74) {
      back();
    }
    if (dta[6] == 90) {
      left();
    }
    if (dta[6] == 16) {
      right();
    }
    if (dta[6] == 56) {
      brake();
    }
  }
}
```

dat[6]を全て
dta[8]に変える