

赤外線ラジコン スケッチ

```
#define IR_PIN          2 // 赤外線受信モジュール接続ピン

#define LPWM           3 // モーター(左)接続ピン
#define LIN1           4
#define LIN2           5
#define RPWM           6 // モーター(右)接続ピン
#define RIN1           8
#define RIN2           7

#define DATA_POINT    3 // 受信したデータから読取る内容のデータ位置

void GO() {
    digitalWrite(LIN1, HIGH);
    digitalWrite(LIN2, LOW);
    analogWrite (LPWM, 255);
    digitalWrite(RIN1, LOW);
    digitalWrite(RIN2, HIGH);
    analogWrite (RPWM, 255);
}

void RIGHT() {
    digitalWrite(LIN1, LOW);
    digitalWrite(LIN2, LOW);
    analogWrite (LPWM, 255);
    digitalWrite(RIN1, LOW);
    digitalWrite(RIN2, HIGH);
    analogWrite (RPWM, 255);
}

void LEFT() {
    digitalWrite(LIN1, HIGH);
    digitalWrite(LIN2, LOW);
    analogWrite (LPWM, 255);
    digitalWrite(RIN1, LOW);
    digitalWrite(RIN2, LOW);
    analogWrite (RPWM, 255);
}

void BACK() {
    digitalWrite(LIN1, LOW);
    digitalWrite(LIN2, HIGH);
    analogWrite (LPWM, 255);
    digitalWrite(RIN1, HIGH);
    digitalWrite(RIN2, LOW);
```

```

    analogWrite (RPWM, 255);
}
void STOP() {
    digitalWrite(LIN1, LOW) ;
    digitalWrite(LIN2, LOW) ;
    analogWrite (LPWM, 255);
    digitalWrite(RIN1, LOW);
    digitalWrite(RIN2, LOW) ;
    analogWrite (RPWM, 255);
}

void setup()
{
    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) ;    // 赤外線受信モジュール接続ピンを入力に設定
    pinMode (LIN1, OUTPUT);
    pinMode (LIN2, OUTPUT);
    pinMode (RIN1, OUTPUT);
    pinMode (RIN2, OUTPUT);
}

void loop() {
    int ans , l , r ;
    ans = IRrecv(); // 赤外線リモコンのデータを受信する
    if (ans != 0) {
        switch (ans) {
            case 0x16: // [>]ボタン：右回転
                {
                    RIGHT();
                }
                break ;
            case 0x12: //前進
                {
                    GO();
                }
                break;
            case 0x14: //左回転
                {
                    LEFT();
                }
                break;
            case 0x17: //後進
                {

```

```
    BACK();
}
break;
//ストップ
case 0x15:
{
    STOP();
}
break;
case 0x25:
{
    STOP();
}
break;
case 0x2A:
{
    STOP();
}
break;
case 0xA9:
{
    STOP();
}
break;
case 0xDF:
{
    STOP();
} break;
case 0x51:
{
    STOP();
} break;
case 0xAC:
{
    STOP();
} break;
case 0x54:
{
    STOP();
} break;
}
Serial.println(ans, HEX);
delay(300);
```

```
}  
}
```

```
/******
```

IRrecv - 赤外線リモコンの送信データを受信する関数

DATA_POINT で指定した位置のデータ(8ビット)を読み取ります。

```
* * * * *
```

戻り: 読み取った整数値を返します

```
*****/
```

```
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 ) ;  
  
}
```