

```

/*
リモコンの押したボタンで LED の点き方が変わる
今回は ONKYO 製 RC-601 使用
*/
/*
押したスイッチの信号データの確認可能
調べたい位置の信号データの確認可能
*/
/*
データポイントの目安 16 進数で表されるようにしている
DATA_POINT1,DATA_POINT2,DATA_POINT3,DATA_POINT4
11,22,33,44
DATA_POINT の数だけ増設可能
今回は DATA_POINT3 を IRrecv1,ans1、 DATA_POINT4 を IRrecv2,ans2 にしている
*/

#define IR_PIN          11 // 赤外線受信モジュール接続ピン
#define DATA_POINT3    3 // 受信したデータから読取る内容のデータ位置 3,4 のどちらか
#define DATA_POINT4    4

//切り替えスイッチの ON,OFF
#define ON    LOW        //プルアップのため HIGH,LOW が逆になる。(私の好み 楽だから)
#define OFF  HIGH       //プッシュスイッチは使用せずジャンパなどで接続する (好みでスライドスイッチにしても良い)

//LED の ON,OFF
#define LON   HIGH
#define LOFF  LOW

//左から 1~9
#define LED1   2
#define LED2   3
#define LED3   4
#define LED4   5
#define LED5   6
#define LED6   7
#define LED7   8
#define LED8   9
#define LED9   10

//切り替えスイッチ
#define IN1     12
#define IN2     13

```

```

void setup() {
  Serial.begin(9600);
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
  pinMode(LED5, OUTPUT);
  pinMode(LED6, OUTPUT);
  pinMode(LED7, OUTPUT);
  pinMode(LED8, OUTPUT);
  pinMode(LED9, OUTPUT);
  pinMode(IR_PIN, INPUT);
  pinMode(IN1, INPUT_PULLUP);
  pinMode(IN2, INPUT_PULLUP);
}

void loop() {
  int ans1 , l1 , r1 , ans2 , l2 , r2 ;
  if (IN1 == ON) {
    ans1 = IRrecv1() ; // 赤外線リモコンのデータを受信する
    if (ans1 != 0) { // データを受信したら処理する
      Serial.println(ans1, HEX); // (ans?, HEX)指定した DATA_POINT の値を 16 進数で表示する
      switch (ans1) { // case 0x?? ??←スイッチ番号 (16 進数) ans=スイッチ番号
        case 0x8E: { //ON
          LED01();
        }
        break;
        case 0x8F: { //STANBY
          LED02();
        }
        break;
        case 0xE: { //1
        }
        break;
        case 0xF: { //2
        }
        break;
        case 0x10: { //3
        }
        break;
        case 0x11: { //4

```

```

    }
    break;
case 0x12: { //5
    }
    break;
case 0x13: { //6
    }
    break;
}
delay(300); // 300ms リモコンチャタリング防止
}
}
else if (IN2 == ON) {
    ans2 = IRrecive2();
    if (ans2 != 0) { // データを受信したら処理する
        Serial.println(ans2, HEX);
        switch (ans2) { // case 0x?? ??←スイッチ番号 (16 進数) ans=スイッチ番号
            case 0x11: {
                LED01();
            }
            break;
        }
        delay(300); //300ms リモコンチャタリング防止
    }
}
else if (IN1 == ON && IN2 == ON) { // 両方 ON
    LED01();
}
else if (IN1 == OFF && IN2 == OFF) { // 両方 OFF
    LED03();
}
}
}

```

//以下 LED 点滅 (任意)

```

void LED01() { //全灯
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, HIGH);
    digitalWrite(LED5, HIGH);
    digitalWrite(LED6, HIGH);
    digitalWrite(LED7, HIGH);
    digitalWrite(LED8, HIGH);
}

```

```

    digitalWrite(LED9, HIGH);
}

void LED02() { //全消
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, LOW);
    digitalWrite(LED5, LOW);
    digitalWrite(LED6, LOW);
    digitalWrite(LED7, LOW);
    digitalWrite(LED8, LOW);
    digitalWrite(LED9, LOW);
}

void LED03() {
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, LOW);
    digitalWrite(LED5, HIGH);
    digitalWrite(LED6, LOW);
    digitalWrite(LED7, HIGH);
    digitalWrite(LED8, LOW);
    digitalWrite(LED9, HIGH);
}

//以下赤外線受信部分（重要）
/*****
    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。
    IRrecv?()
*
    戻り: 読み取った整数値を返します
    return
*****/

int IRrecv1() //赤外線受信部分 1 DATA_POINT3
{
    unsigned long t1 ;
    int i1 , j1 ;
    int cnt1 , ans1 ;
    char IRbit[64] ;

```

```

ans1 = 0 ;
t1 = 0 ;
if (digitalRead(IR_PIN) == LOW) {
    // リーダ部のチェックを行う
    t1 = micros() ; // 現在の時刻(us)を得る
    while (digitalRead(IR_PIN) == LOW) ; // HIGH(ON)になるまで待つ
    t1 = micros() - t1 ; // 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 (t1 >= 3400) {
    i1 = 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
        t1 = micros() ;
        cnt1 = 0 ;
        while (digitalRead(IR_PIN) == HIGH) { // LOW(OFF)になるまで待つ
            delayMicroseconds(10) ;
            cnt1++ ;
            if (cnt1 >= 1200) break ; // 12ms 以上 HIGH のままなら中断
        }
        t1 = micros() - t1 ;
        if (t1 >= 10000) break ; // ストップデータ
        if (t1 >= 1000) IRbit[i1] = (char)0x31 ; // ON 部分が長い
        else IRbit[i1] = (char)0x30 ; // ON 部分が短い
        i1++ ;
    }
    // データ有りなら指定位置のデータを取り出す
    if (i1 != 0) {
        i1 = (DATA_POINT3 - 1) * 8 ;
        for (j1 = 0 ; j1 < 8 ; j1++) {
            if (IRbit[i1 + j1] == 0x31) bitSet(ans1, j1) ;
        }
    }
}
return ( ans1 ) ;
}

```

```

int IRrecive2() //赤外線受信部分 2 DATA_POINT4
{
    unsigned long t2 ;
    int i2 , j2 ;

```

```

int cnt2 , ans2 ;
char IRbit[64] ;

ans2 = 0 ;
t2 = 0 ;
if (digitalRead(IR_PIN) == LOW) {
    // リーダ部のチェックを行う
    t2 = micros() ; // 現在の時刻(us)を得る
    while (digitalRead(IR_PIN) == LOW) ; // HIGH(ON)になるまで待つ
    t2 = micros() - t2 ; // 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 (t2 >= 3400) {
    i2 = 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
        t2 = micros() ;
        cnt2 = 0 ;
        while (digitalRead(IR_PIN) == HIGH) { // LOW(OFF)になるまで待つ
            delayMicroseconds(10) ;
            cnt2++ ;
            if (cnt2 >= 1200) break ; // 12ms 以上 HIGH のままなら中断
        }
        t2 = micros() - t2 ;
        if (t2 >= 10000) break ; // ストップデータ
        if (t2 >= 1000) IRbit[i2] = (char)0x31 ; // ON 部分が長い
        else IRbit[i2] = (char)0x30 ; // ON 部分が短い
        i2++ ;
    }
    // データ有りなら指定位置のデータを取り出す
    if (i2 != 0) {
        i2 = (DATA_POINT4 - 1) * 8 ;
        for (j2 = 0 ; j2 < 8 ; j2++) {
            if (IRbit[i2 + j2] == 0x31) bitSet(ans2, j2) ;
        }
    }
}
return ( ans2 ) ;
}

```