

LEDを光らせよう

第11回 赤外線リモコンを使ってみよう

<http://cms.db.tokushima-u.ac.jp/DAV/person/S10723/LEDを光らせよう/>

川上 博

2015/11/07

今日のテーマ

1 前回の補足

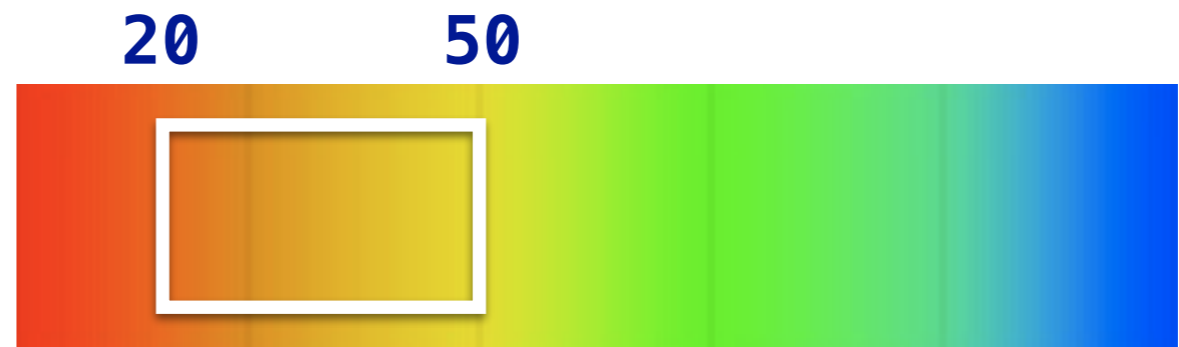
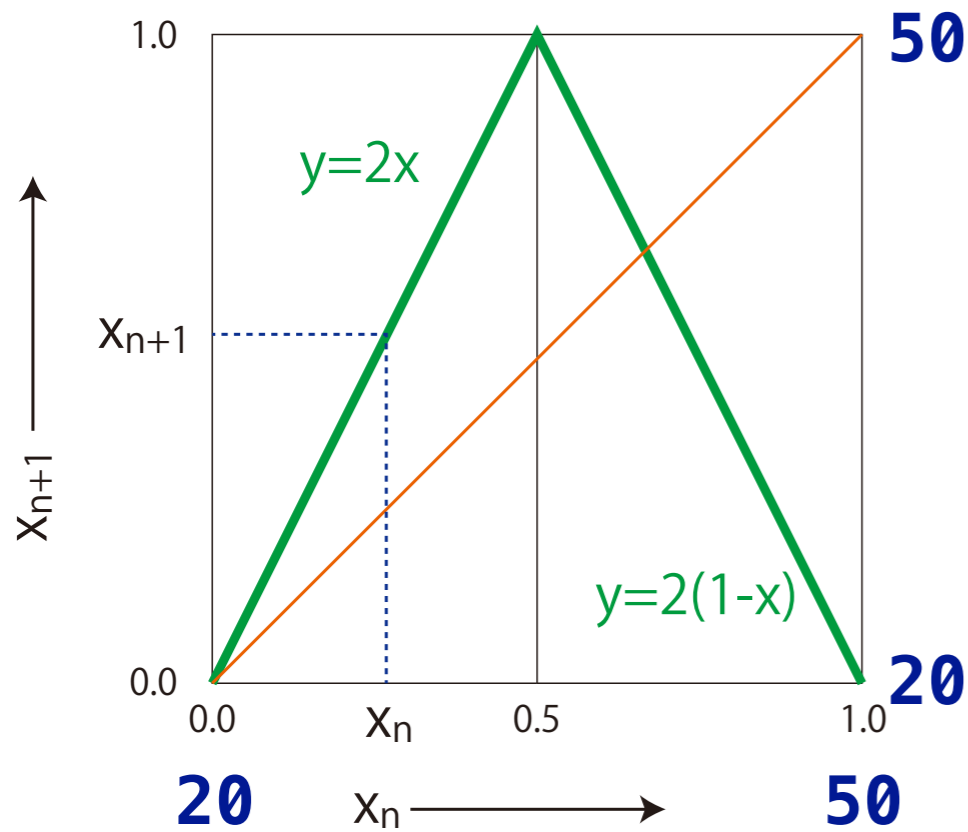
- a) カラーLEDをローソクのように点灯する
- b) 距離センサー入力の「ちらつき」を除去する
- c) シリアル・モニターで変数を表示する

2 赤外線(IR: infra-red)リモコンを使ってみる

- a) リモコンのボタン・コードを知る
- b) ボタンごとにLEDの色相を変えるスケッチを書く

ローソクの光を真似る : candleLight

簡単な乱数っぽい数列を使って見る : 色相, 輝度



```
sensor=map(sensor, 255, 0, 20, 50);
```

```
int flicker()
```

```
..., 27, 35, 20, 44, 38, 27, 35, 20, 45, 40, 31, ..., Xn, Xn+1, ...
```

カラーLEDの「ちらつき」を減らす : flicker

適当な回数センサーの値を平均する

```
int NN=10;

sensor=0;
for(int i=0; i<NN; i++){
    sensor += analogRead(SENSOR_PIN);
}
sensor = sensor/NN;
```

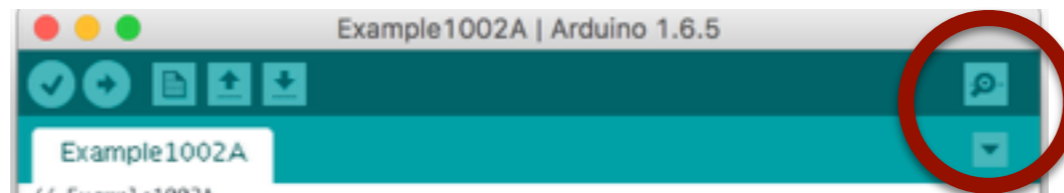
シリアル・モニターで数値を見る : SerialMonitor

```
void setup() {
  Serial.begin(9600);
}
```

```
float x, y;
```

```
void loop() {
  delay(1000);
  for(int i=1; i<101; i++){
    Serial.println(i,DEC); // decimal format
    Serial.println(i,HEX); // hexadecimal format
    Serial.println(i,BIN); // binary format
    x=(float)i;
    Serial.print(x);
    Serial.print(" ");
    y=log10(x);
    Serial.println(y,4); // print 4 decimal places
    delay(2000);
  }
}
```

シリアル・モニター



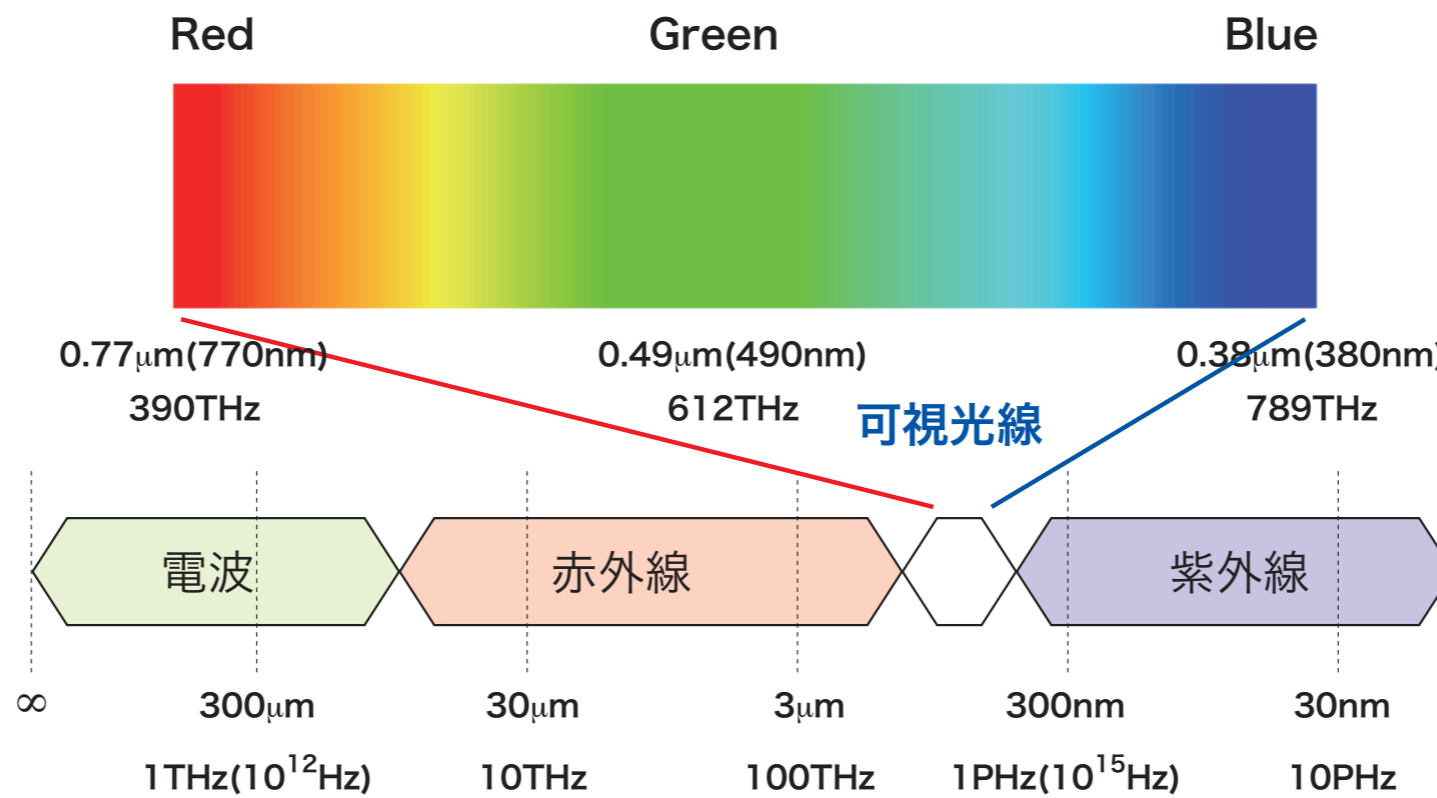
モニター表示例

```
24
18
11000
24.00    1.3802
```

赤外線リモコン(IR remote)を使って見る

可視光線：波長と周波数

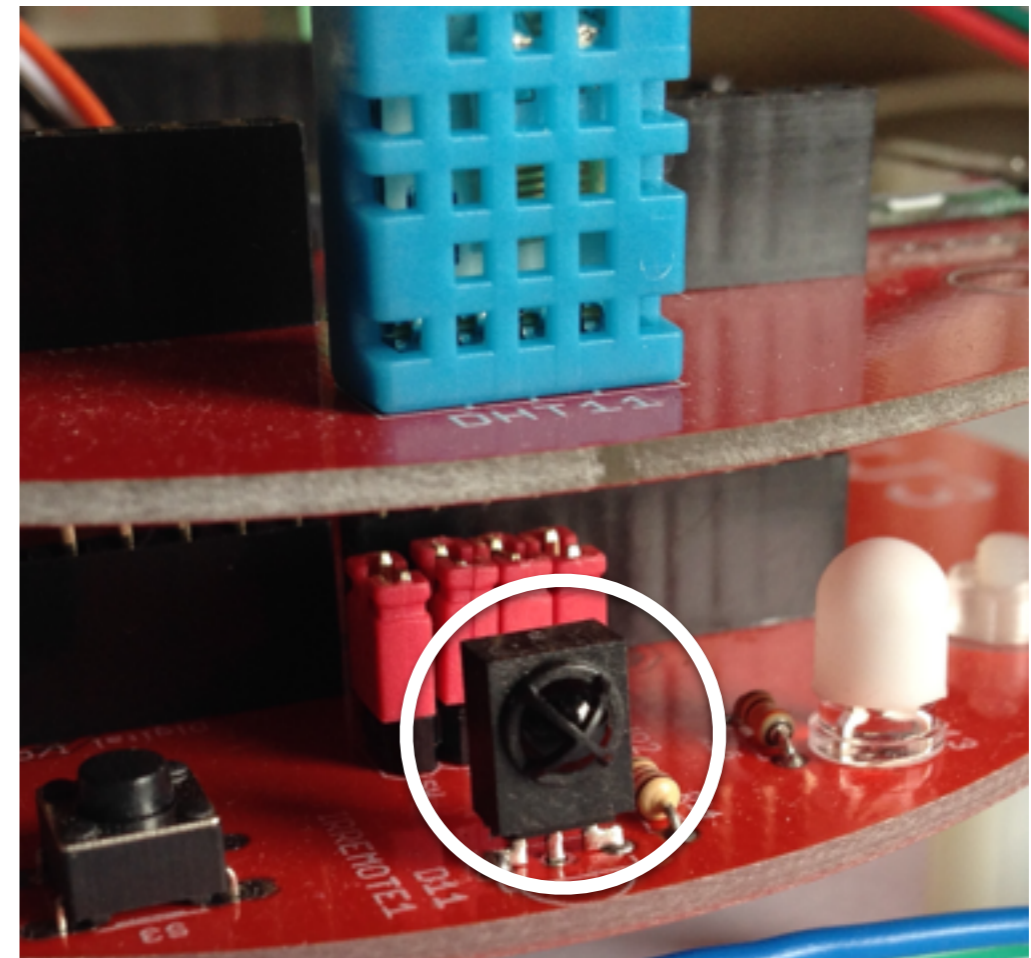
940nm



赤外線リモコン : IR remote

IR 送信器

IR 受信器



家電製品協会フォーマット
NECフォーマット
sonyフォーマット

IR レシーバーピン : D11

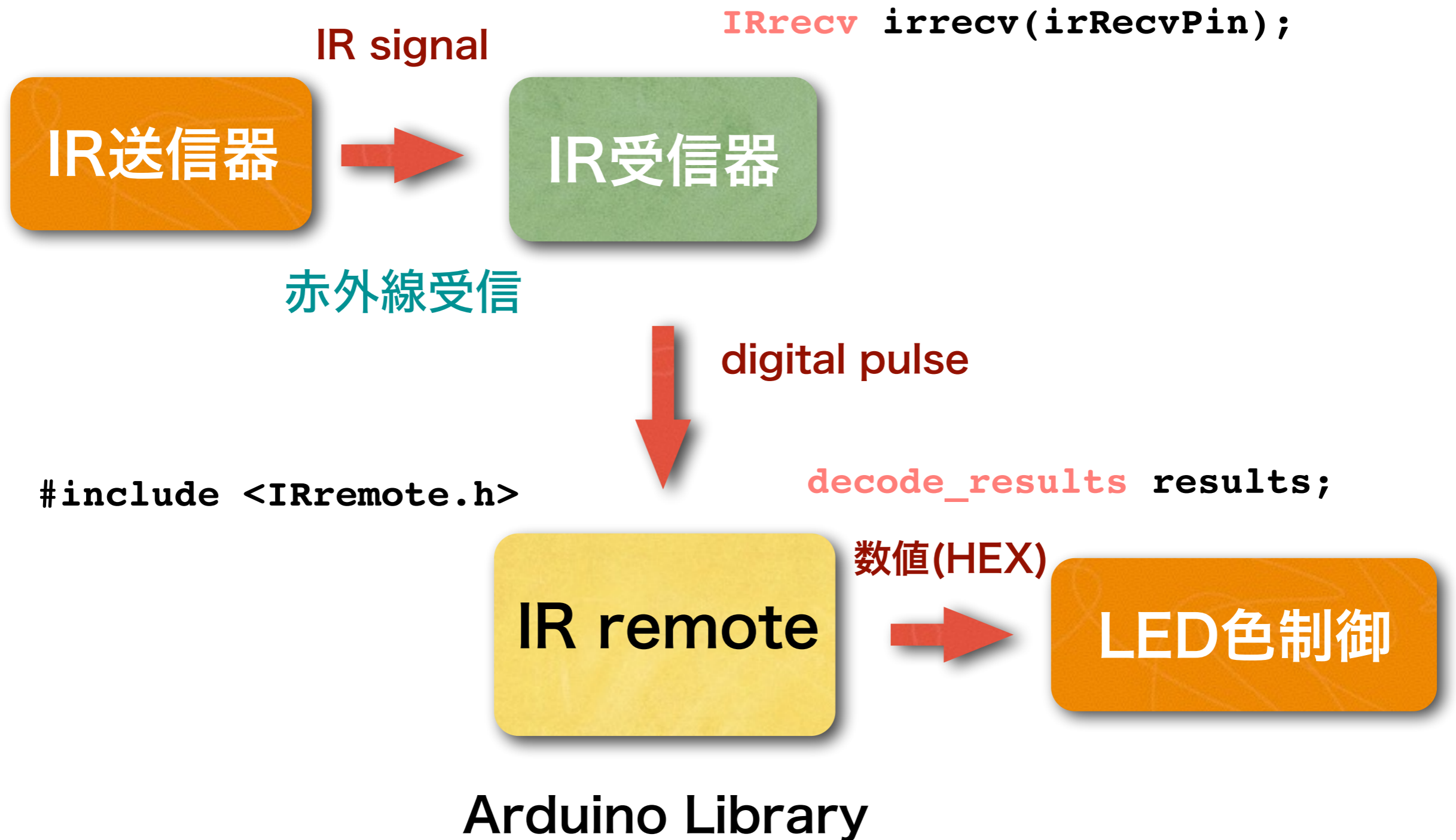
辻先生配布のDVD/資料

JJ3 ポート一覧

ポート	接続先	型番	ライブラリ
D0 (INT2)	人感センサ*1	SB-612A	digitalRead, INT2
D1	—	—	—
D2 (SDA)	液晶ディスプレイ 8x2	AQM0802	ST7032
D3 (SCL)	高精度温湿度センサ*1	AE-HDC1000	
D4	フルカラーLED (WS2811 互換)	WS2811LED	FastLED
D5, D6, D7	—		—
D8	タクトスイッチ	SW1	digitalRead INPUT_PULLUP
D9	温湿度センサ	DHT11	DHT
D10	圧電スピーカー	PKLCS1212	Tone
D11	赤外線受信センサ	GP1UXC4XQS	IRremote
D12	—		—
D13	青色 LED		digitalWrite
A0	照度センサ	NJL7502L	analogRead
A1, A2	—		—
A3	赤外線距離センサ	GP2Y0A21YK	analogRead
A4, A5	—		—

*1 オプション

赤外線通信 (受信の場合)



IRリモコンのボタンを押す毎に、 LEDが点灯・消灯を繰り返す

```
// Example1100:
// IRremote ON/OFF detector sketch
// H. Kawakami

#include <FastLED.h>
#include <IRremote.h>

const int dataPin = 4;
const int irRecvPin = 11;
const int ledPin = 13;

CRGB leds[1];
IRrecv irrecv(irRecvPin); // IR receive object
decode_results results; // IR decode object

void setup(){
  pinMode(ledPin, OUTPUT);
  FastLED.addLeds<WS2811, dataPin, RGB>(leds, 1);
  irrecv.enableIRIn();
}
```

```
int lightState = 0;

void loop(){
  if (irrecv.decode(&results)==true){
    delay(200);
    lightState = 1-lightState;
    digitalWrite(ledPin, lightState);
    irrecv.resume();

    if(lightState==0){
      leds[0].setRGB(0, 0, 0);
    }else{
      leds[0].setRGB(255, 68, 221);
    }
    FastLED.show();
  }
}
```

ボタンとコードの対応表

ボタン	コード	ボタン	コード
Power	0xFF48B7	Ok	0xFFA05F
Source	0xFF7887	Menu	0xFF50AF
Up	0xFF40BF	Vol_Down	0xFF30CF
Left	0xFF20DF	Vol_Off	0xFFF807
Down	0xFF10EF	Vol_Up	0xFF708F
Right	0xFF609F	Repeat	0xFFFFFFFF
		Unknown	0

シリアルモニタで ボタンのコードを読む

```
// Example1100A:

#include <FastLED.h>
#include <IRremote.h>

const int dataPin = 4;
const int irRecvPin = 11;
const int ledPin = 13;

CRGB leds[1];
IRrecv irrecv(irRecvPin); // IR receive object
decode_results results; // IR decode object

void setup(){
  pinMode(ledPin, OUTPUT);
  FastLED.addLeds<WS2811, dataPin, RGB>(leds, 1);
  irrecv.enableIRIn();
  Serial.begin(9600);
}
```

```
int lightState = 0;
void loop(){
  if (irrecv.decode(&results)==true){
    delay(200);
    lightState = 1-lightState;
    digitalWrite(ledPin, lightState);
    Serial.println(results.value, HEX);
    irrecv.resume();

    if(lightState==0){
      leds[0].setRGB(0, 0, 0);
    }else{
      leds[0].setRGB(255, 68, 221);
    }
    FastLED.show();
  }
}
```

ボタンのコード毎に LEDを着色する

```
// Example1101

#include <FastLED.h>
#include <IRremote.h>

const int dataPin = 4;
const int irRecvPin = 11;
const int ledPin = 13;

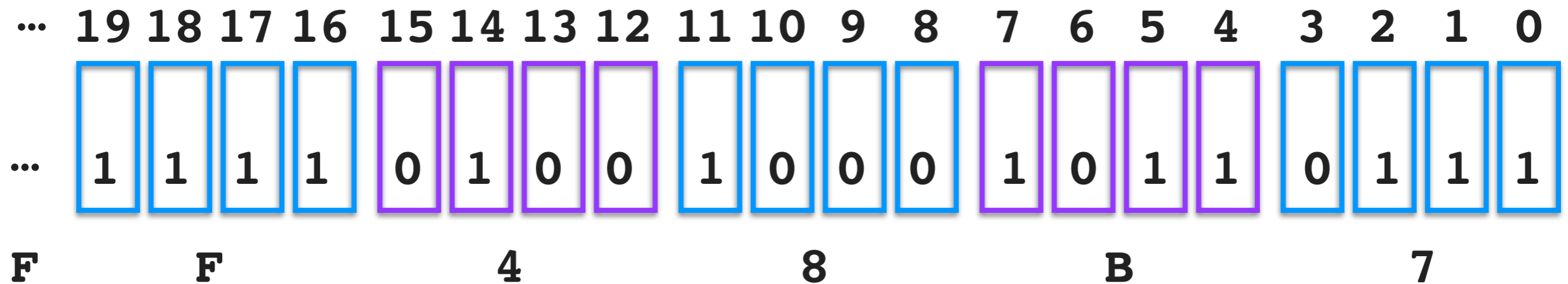
CRGB leds[1];
IRrecv irrecv(irRecvPin); // IR receive object
decode_results results; // IR decode object

void setup(){
  pinMode(ledPin, OUTPUT);
  FastLED.addLeds<WS2811, dataPin, RGB>(leds, 1);
  irrecv.enableIRIn();
}
```

```
int lightState = 0;

void loop(){
  if (irrecv.decode(&results)){
    unsigned long rv = results.value;
    delay(200);
    lightState = 1-lightState;
    digitalWrite(ledPin, lightState);
    if(rv == 0xFF48B7){ // Power button
      leds[0].setRGB(255, 68, 221); // hot pink
    }
    if(rv == 0xFF7887){ // Source button
      leds[0].setRGB(0, 0, 255); // blue
    }
    FastLED.show();
    irrecv.resume();
  }
}
```

POWER : 0xFF48B7



0xFF48B7

HEX	binary	HEX	binary
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

```

if (irrecv.decode(&results)) {
  unsigned long rv = results.value;
  delay(250);
  if (rv==0xFF48B7) {
    POWERが押された時にする仕事
  }
  irrecv.resume();
}

```

今日の演習：LEDの色指定

1 Example1101A の各ボタンに色を定義してください

```
void process_result() {
  int code = ir_key();
  switch (code) {
    case KEY_POWER: {
      Serial.println("Power ON");
      leds[0].setRGB(0, 0, 0);
      FastLED.show();
    }
    break;
    case KEY_SOURCE: {
      Serial.println("Source");
      leds[0].setRGB(0, 0, 0);
      FastLED.show();
    }
    break;
    case KEY_UP: {
      Serial.println("UP");
      leds[0].setRGB(255, 0, 0); // red
      FastLED.show();
    }
    break;
  }
}
```

今日の演習：音階を指定

2 toneExampleを参考にしてIR送信機のボタンを押すと音階が出るスケッチを作ってください

```
// toneExample
// tone(pin, frequency)
// pin: the pin on which to generate the tone
// frequency: the frequency of the tone in hertz(unsigned int)
//
// (note:frequency)
// C4(Do):262, D4(Re):294, E4(Mi):330, F4(Fa):349,
// G4(So):392, A4(Ra):440, B4(Si):494, C5(Do):523
//
// void myTone(int frequency, int duration)
```

```
void myTone(int freq, int dur){
    tone(10, freq);    delay(dur);    noTone(10);    delay(dur*4);
}
```

```
void setup() {
    myTone(440, 100); myTone(440, 100); myTone(440, 100); myTone(880, 1000);
}
```

```
void loop() {}
```


今日の演習：Example1100T

```
// Example1100T

#include <IRremote.h>

const int tonePin = 10;
const int irRecvPin = 11;

IRrecv irrecv(irRecvPin); // IR receive object
decode_results results; // IR decode object

void myTone(int freq, int dur){
    tone(tonePin, freq);
    delay(dur);
    noTone(tonePin);
    delay(dur*4);
}

void setup(){
    delay(2000);
    irrecv.enableIRIn();
}

void loop(){
    if (irrecv.decode(&results)==true){
        delay(200);
        if(results.value== 0xFF48B7){ // Power button
            myTone(440, 100);
        }
        if(results.value== 0xFF7887){ // Source button
            myTone(880, 500);
        }
        irrecv.resume();
    }
}
```