

loverino.c

```
#include <Adafruit_NeoPixel.h>
#include <MsTimer2.h>
#include <MsTimer2.h>

#ifdef __AVR__
#include <avr/power.h>
#include <MsTimer2.h>
#endif

#define MAXPOWER 100
#define LOW 0
#define HIGH 1
#define BASE 60 //基準心拍数
#define HEIKIN_VALUE 5//平均回数
#define CHECK_NUM 3 //センサーの確認回数

// Which pin on the Arduino is connected to the NeoPixels?
// On a Trinket or Gemma we suggest changing this to 1
#define PIN 5

// How many NeoPixels are attached to the Arduino?
#define NUMPIXELS 3
int SenserPin = 7;

#define SENSER_VALUE 40// 心拍センサーの閾値

Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

unsigned int r, g, b; // 色の強さ
long hart; //心拍数カウント用

void timerHandler(void) {
    hart++;/* 割り込みのたびにカウントアップ*/
}

void setup() {
    // ビンモード初期化
    pinMode(SenserPin, INPUT); // sets the digital pin as output
    pinMode(PIN, OUTPUT); // sets the digital pin as output
    //タイマー初期化
    MsTimer2::set(10, timerHandler);// 10msでタイマー起動
    MsTimer2::start();// タイマー開始

    // put your setup code here, to run once:
    pixels.begin(); // This initializes the NeoPixel library.
    g = 0;
    r = 0;
    b = MAXPOWER;
    pixels.setPixelColor(0, pixels.Color(r, g, b));
    pixels.setPixelColor(1, pixels.Color(r, g, b));
    pixels.setPixelColor(2, pixels.Color(r, g, b));

    pixels.show(); // This sends the updated pixel color to the hardware.
}

int check_senser(void)
{
    int i;
    for (i = 0; i < CHECK_NUM; i++) {
        if (analogRead(SenserPin) < SENSER_VALUE) {
            return -1;
        }
    }
    return 0;
}

void loop() {

    // put your main code here, to run repeatedly:
    static long state;
    int i;

    if (check_senser() == 0) { // 心拍検出時にL
        if (state == LOW) { //心拍検出時に計算
```

loverino.c

```
// 感覚で色を決定
if (hart <= 100) {
    r = (100 - hart);
    b = (MAXPOWER - r);
}
else {
    r = 0;
    b = MAXPOWER;
}
state = HIGH;
hart = 0;
for (i = 0; i < NUMPIXELS; i++) {
    pixels.setPixelColor(i, pixels.Color(r, g, b));
}
}
}
else {
    for (i = 0; i < NUMPIXELS; i++) {
        pixels.setPixelColor(i, pixels.Color(0, 0, 0));
    }
    state = LOW;
}
pixels.show(); // This sends the updated pixel color to the hardware.
delay(50);
}
```