

```
/*
kyousitu_24_8
```

○音楽演奏曲目

- ・ジングルベル
  - ・赤鼻のトナカイ
  - ・大きな古時計
- LED を赤と緑交互に配置  
○LED フラッシュパターン変更  
○ランダム演奏はなし

ATMEGA 168

```
*/
#include "TimerOne.h"
#include "pitches4.h"

/* onkai 対応記号 = {
R, dDO, dDOs, dRE, dREs, dMI, dFA, dFAs, dSO, dSOs, dRA, dRAs, dSif, dSI,
DO, DOs, RE, REs, MI, FA, FAs, SO, SOs, RA, RAs, Sif, SI,
uDO, uDOs, uRE, uREs, uMI, uFA, uFAs, uSO, uSOs, uRA, uRAs, uSif, uSI,
};
*/
const int onkai[] = {
0, 131, 139, 147, 156, 165, 175, 185, 196, 208, 220, 233, 233, 247,
262, 277, 294, 311, 330, 349, 370, 392, 415, 440, 466, 466, 494,
523, 554, 587, 622, 658, 698, 740, 784, 831, 880, 932, 932, 988
};
const int kouka[] = { 523,587,659,698,784,880,988,1047 };
```

// 音楽データ

```
/*
teunit 16分音符の長さ(ms)
te 拍子(2/4 のときは 8, 3/4 のときは 12, 4/4 のときは 16)
sc[] 楽譜データ {音程、長さ}* 長さは16分音符を1とする
len 楽譜サイズ
*/
```

// ===== (1) ジングルベル

```
// 1 OCT UP..... u
// 1 OCT DOWN.... d
```

```
// DO RE MI FA SO RA SI R
```

```
// HALF UP ..... s
// HALF DOWN .... f
```

```
const int teunit1 = 120; // 16分音符の長さ(ms)
const int te1 = 16; // 拍子 2/4 のときは 8, 3/4 のときは 12, 4/4 のときは 16 ==> 16
const byte oto1[] = {
// (1)
DO,RA,SO,FA,DO,DO,DO,DO,RA,SO,FA,RE,RE,RE,Sif,RA,SO,MI,uDO,uDO,uDO,Sif,SO,RA,RE,
DO,RA,SO,FA,DO,DO,DO,RA,SO,FA,RE,RE,RE,Sif,RA,SO,uDO,uDO,uDO,uDO,uRE,uDO,Sif,SO,FA,R,uDO,
RA,RA,RA,RA,RA,RA,uDO,FA,SO,RA,R,Sif,Sif,Sif,Sif,Sif,RA,RA,RA,RA,RA,SO,SO,RA,SO,R,uDO,
RA,RA,RA,RA,RA,RA,RA,uDO,FA,SO,RA,R,Sif,Sif,Sif,Sif,Sif,RA,RA,RA,RA,uDO,uDO,Sif,SO,FA
};
const byte nag1[] = {
// (1)
2,2,2,2,6,1,1,2,2,2,2,6,2,2,2,2,6,2,2,2,2,6,2,
2,2,2,2,6,2,2,2,2,6,2,2,2,2,2,2,2,2,2,2,2,2,4,2,2,4,2,2,4,

```

```
2,2,3,1,6,2,2,2,3,1,2,2,2,1,1,2,2,2,2,2,4,  
2,2,4,2,2,4,2,2,3,1,6,2,2,2,3,1,2,2,2,1,1,2,2,2,2,6,2  
};
```

```
const int len1 = sizeof(oto1);  
// =====
```

```
// ===== (2) 赤鼻のトナカイ  
// 1 OCT UP..... u  
// 1 OCT DOWN.... d
```

```
// DO RE MI FA SO RA SI R
```

```
// HALF UP ..... s  
// HALF DOWN .... f
```

```
const int teunit2 = 110; // 16分音符の長さ(ms)  
const int te2 = 16; // 拍子 2/4 のときは 8, 3/4 のときは 12, 4/4 のときは 16 ==> 16  
const byte oto2[] = {
```

```
// (1)  
SO,RA,SO,MI,uDO,RA,SO,R,SO,RA,SO,RA,SO,uDO,SI,R,  
FA,SO,FA,RE,SI,RA,SO,R,SO,RA,SO,RA,SO,RA,MI,R,  
SO,RA,SO,MI,uDO,RA,SO,R,SO,RA,SO,RA,SO,uDO,SI,R,  
FA,SO,FA,RE,SI,RA,SO,R,SO,RA,SO,RA,SO,uRE,uDO,R,  
RA,RA,uDO,RA,SO,MI,SO,FA,RA,SO,FA,MI,R,  
RE,MI,SO,RA,SI,SI,SI,uDO,uDO,SI,RA,SO,FA,RE,RE,R,  
SO,RA,SO,MI,uDO,RA,SO,R,SO,RA,SO,RA,SO,uDO,SI,R,  
FA,SO,FA,RE,SI,RA,SO,R,SO,RA,SO,RA,SO,uRE,uDO,R  
};
```

```
const byte nag2[] = {  
// (1)
```

```
4,2,2,4,4,4,8,4,2,2,2,2,4,4,12,4,  
2,2,4,4,4,4,8,4,2,2,2,2,4,4,12,4,  
2,2,4,4,4,4,8,4,2,2,2,2,4,4,12,4,  
4,2,2,4,4,4,8,4,2,2,2,2,4,4,12,4,  
4,4,4,4,4,8,4,4,4,4,12,4,  
4,4,4,4,4,8,4,4,4,4,2,2,4,4,  
2,2,4,4,4,4,8,4,2,2,2,2,4,4,12,4,  
2,2,4,4,4,4,8,4,2,2,2,2,4,4,12,4  
};
```

```
const int len2 = sizeof(oto2);  
// =====
```

```
// ===== (3) 大きな古時計
```

```
const int teunit3 = 130; // 16分音符の長さ(ms)  
const int te3 = 16; // 拍子 2/4 のときは 8, 3/4 のときは 12, 4/4 のときは 16 ==> 16  
const byte oto3[] = {
```

```
// (1)  
RE,SO,FAs,SO,RA,SO,RA,SI,SI,uDO,SI,MI,RA,RA,SO,SO,SO,FAs,MI,FAs,SO,R,RE,RE,  
SO,FAs,SO,RA,SO,RA,SI,uDO,SI,MI,RA,RA,SO,SO,SO,FAs,MI,FAs,SO,R,SO,SI,  
uRE,SI,RA,SO,FAs,SO,RA,SO,FAs,MI,RE,SO,SI,uRE,SI,RA,SO,FAs,SO,RA,R,RE,  
SO,SO,R,RA,R,SI,SI,uDO,SI,MI,RA,RA,SO,FAs,SO,R,RE,RE,  
SO,RE,RE,MI,RE,RE,dSI,RE,dSI,RE,RE,SO,RE,RE,MI,RE,RE,dSI,RE,dSI,RE,RE,  
SO,SO,R,RA,R,SI,SI,uDO,SI,MI,RA,RA,SO,FAs,SO,R
```

```
};
```

```

const byte nag3[] = {
// (1)
  4,4,2,2,4,2,2,2,2,2,2,4,2,2,4,2,2,4,2,2,8,4,2,2,
  4,2,2,4,2,2,4,2,2,4,2,2,4,2,2,4,2,2,8,4,2,2,
  4,2,2,4,2,2,2,2,2,2,4,2,2,4,2,2,4,2,2,12,2,2,
  2,2,4,4,4,2,2,2,2,4,2,2,8,8,8,4,2,2,
  4,2,2,2,2,4,4,4,2,2,4,2,2,4,2,2,4,4,4,2,2,
  2,2,4,4,4,2,2,2,2,4,2,2,8,8,12,4
};

const int len3 = sizeof(oto3);
// =====

const int out_p = 8; // スピーカーピン
const int led_1[] = {5, 6, 7, 9, 10, 11, 12, 13}; // LEDピン
const int ir_in =2; // 赤外線入力 PIN
const int test =4; // TEST PIN "1"...動作確認モード、"0"...通常モード

int le = 0; // 現在の LEDピン
int ole = 0; // 直前の LEDピン
int no = 1; // 現在の演奏曲ナンバー

int md_time; // リモコン受信時間
int val,co,co2,ijj2,pt,e_flg;
int cur,note,dur;
unsigned long t1,t2;
float pt2;

void setup() { // ***** SETUP

  for (int led = 0; led < 8; led++) {
    pinMode(led_1[led], OUTPUT);
    digitalWrite(led_1[led], LOW); // set the LED off
  }
  pinMode(ir_in, INPUT);
  pinMode(test, INPUT);

  randomSeed(analogRead(0));

  Timer1.initialize(150000); // タイマ1を初期化し、0.15 S (= 150000 uS)を設定
  Timer1.attachInterrupt(ir_chk); // タイマ・オーバフロー割込みとして ir_chk()をアタッチする
}

void ir_chk() { //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! TIMER 1 INT ir_chk

  if ( digitalRead(ir_in)==0 ) {
    co++;
    if ( co==6 ) { co=0; }
    if ( co2==0 ) { t2=millis(); }
    co2++;
    if ( co2==8 ) {
      if ( millis()-t2<=2000 ) { e_flg=1; }
      else co2=0;
    }
  }
}

void ir_md() //***** SUB IR_MODE INPUT
{

```

```

digitalWrite(led_1[0], HIGH);
t1 = millis();

do {
  if ( co==0 ) { digitalWrite(led_1[7], LOW); digitalWrite(led_1[6], LOW);
    digitalWrite(led_1[5], LOW); digitalWrite(led_1[4], LOW); digitalWrite(led_1[3], LOW);
    digitalWrite(led_1[2], LOW);}
  if ( co==1 ) { digitalWrite(led_1[7], HIGH); }
  if ( co==2 ) { digitalWrite(led_1[6], HIGH); }
  if ( co==3 ) { digitalWrite(led_1[5], HIGH); }
  if ( co==4 ) { digitalWrite(led_1[4], HIGH); }
  if ( co==5 ) { digitalWrite(led_1[3], HIGH); }
  if ( e_flg==1 ) { break; }
} while ( millis()-t1 < md_time );
}

```

```

void ir_pt1() { //***** SUB IR_PITCH 1 INPUT

  if ( co==4 ) { co=1; }

  if ( co==1 ) { pt=40; }
  if ( co==2 ) { pt=90; }
  if ( co==3 ) { pt=300; }
  delay(pt);
}

```

```

void ir_pt2() { //***** SUB IR_PITCH 2 INPUT

  if ( co==4 ) { co=1; }

  if ( co==1 ) { pt2=0.7; }
  if ( co==2 ) { pt2=1.1; }
  if ( co==3 ) { pt2=1.8; }
  delay(dur*pt2);
}

```

```

void loop() { // ***** MAIN

  if ( digitalRead(test)==1 ) { // ***** 動作確認モード (4PIN) ="1"
    for(;;) {
      for (int led = 0; led < 8; led++) { digitalWrite(led_1[led], HIGH); }
      if ( digitalRead(ir_in)==1 ) { //キー入力なし
        tone(out_p, onkai[DO], 200); }
      if ( digitalRead(ir_in)==0 ) { //キー入力あり
        tone(out_p, onkai[DO6], 200); }
      delay(1000);

      if ( digitalRead(ir_in)==1 ) { //キー入力なし
        for (int led = 0; led < 8; led++) { digitalWrite(led_1[led], LOW); }
        tone(out_p, onkai[DO], 200); }
      if ( digitalRead(ir_in)==0 ) { //キー入力あり
        tone(out_p, onkai[DO6], 200); }
      delay(1000);
    }
  }
}

```

```

//***** 通常モード

```

```

for (int led = 0; led < 8; led++) {

```

```

    digitalWrite(led_1[led], HIGH);
    delay(200);
}
st1:// * st1 *
co=0; //***** ホットスタート
co2=0;
e_flg=0;

for (int led = 0; led < 8; led++) {
    digitalWrite(led_1[led], LOW);
}

tone(out_p, onkai[DO], 200); //キー入力受付
delay(100);
tone(out_p, onkai[DO6], 200);

md_time = 5000;
ir_md(); //=====
if ( e_flg==1 ) { goto st1; }

digitalWrite(led_1[0], LOW);

for ( j=0; j<3; j++) { // 3回点滅、“ヒ°ヒ°”
    for ( i=1; i<=co; i++) {
        digitalWrite(led_1[8-i], LOW); }
    delay(200);
    for ( i=1; i<=co; i++) {
        digitalWrite(led_1[8-i], HIGH); }
    tone(out_p, onkai[DO6], 200);
    delay(200);
}
for ( i=1; i<=co; i++) {
    digitalWrite(led_1[8-i], LOW); }

switch (co) {

case 0://***** MODE ( 0 ).... ろうそくの炎
    c0:// * c0 *
    for (i=0;i<8;i++) {
        j=random(0, 13); analogWrite(led_1[i], 20*j); // analogWrite()に与える値は0から255
        delay(20);
    }
    digitalWrite(led_1[5], HIGH);
    if ( e_flg==1 ) { break; }
    goto c0;

case 1:// ***** MODE ( 1 )...LED 点滅 1
    co=2;
    c1:// * c1 *
    for ( j=0; j<3; j++) {
        for (i=0; i<8; i++) { digitalWrite(led_1[i], HIGH); ir_pt1(); digitalWrite(led_1[i], LOW); }
    }
    if ( e_flg==1 ) break;
    for ( j=0; j<3; j++) {
        for (i=0; i<4; i++) { digitalWrite(led_1[i*2], HIGH); ir_pt1(); digitalWrite(led_1[i*2], LOW); }
    }
    if ( e_flg==1 ) break;
    for ( j=0; j<3; j++) {
        for (i=0; i<4; i++) { digitalWrite(led_1[i*2+1], HIGH); ir_pt1(); digitalWrite(led_1[i*2+1], LOW); }
    }
}

```

```

if ( e_flg==1 ) break;

////
for ( j=0; j<3; j++ ) {
  for ( i=0; i<8; i++ ) { digitalWrite(led_1[7-i], HIGH); ir_pt1(); digitalWrite(led_1[7-i], LOW); }
}
if ( e_flg==1 ) break;
for ( j=0; j<3; j++ ) {
  for ( i=0; i<4; i++ ) { digitalWrite(led_1[6-i*2], HIGH); ir_pt1(); digitalWrite(led_1[6-i*2], LOW); }
}
if ( e_flg==1 ) break;
for ( j=0; j<3; j++ ) {
  for ( i=0; i<4; i++ ) { digitalWrite(led_1[7-i*2], HIGH); ir_pt1(); digitalWrite(led_1[7-i*2], LOW); }
}
if ( e_flg==1 ) break;

goto c1;
case 2:// ***** MODE ( 2 )...LED 点滅 2
co=2;
c2:// * c2 *
// (1-1)
for (i=0; i<8; i++) { digitalWrite(led_1[i], HIGH); ir_pt1(); }
if ( e_flg==1 ) break;
for (i=0; i<8; i++) { digitalWrite(led_1[7-i], LOW); ir_pt1(); }
if ( e_flg==1 ) break;

// (2-1)
for ( j=0; j<3; j++ ) {
  for (i=0; i<4; i++) { digitalWrite(led_1[i*2], HIGH); ir_pt1(); }
  for (i=0; i<4; i++) { digitalWrite(led_1[6-i*2], LOW); ir_pt1(); }
}
if ( e_flg==1 ) break;
// (2-2)
for ( j=0; j<3; j++ ) {
  for (i=0; i<4; i++) { digitalWrite(led_1[i*2+1], HIGH); ir_pt1(); }
  for (i=0; i<4; i++) { digitalWrite(led_1[7-i*2], LOW); ir_pt1(); }
}
if ( e_flg==1 ) break;
// (3-1)
for (i=0; i<3; i++) { digitalWrite(led_1[i*3], HIGH); ir_pt1(); }
if ( e_flg==1 ) break;
for (i=0; i<3; i++) { digitalWrite(led_1[6-i*3], LOW); ir_pt1(); }
if ( e_flg==1 ) break;
// (3-2)
for (i=0; i<3; i++) { digitalWrite(led_1[i*3+1], HIGH); ir_pt1(); }
if ( e_flg==1 ) break;
for (i=0; i<3; i++) { digitalWrite(led_1[7-i*3], LOW); ir_pt1(); }
if ( e_flg==1 ) break;
////
for (i=0; i<8; i++) { digitalWrite(led_1[7-i], HIGH); ir_pt1(); }
if ( e_flg==1 ) break;
for (i=0; i<8; i++) { digitalWrite(led_1[i], LOW); ir_pt1(); }
if ( e_flg==1 ) break;

for ( j=0; j<3; j++ ) {
  for (i=0; i<4; i++) { digitalWrite(led_1[6-i*2], HIGH); ir_pt1(); }
  for (i=0; i<4; i++) { digitalWrite(led_1[i*2], LOW); ir_pt1(); }
}
if ( e_flg==1 ) break;
for ( j=0; j<3; j++ ) {

```

```

    for (i=0; i<4; i++) { digitalWrite(led_1[7-i*2], HIGH); ir_pt1(); }
    for (i=0; i<4; i++) { digitalWrite(led_1[i*2+1], LOW); ir_pt1(); }
}
if ( e_flg==1 ) break;

for (i=0; i<3; i++) { digitalWrite(led_1[6-i*3], HIGH); ir_pt1(); }
if ( e_flg==1 ) break;
for (i=0; i<3; i++) { digitalWrite(led_1[i*3], LOW); ir_pt1(); }
if ( e_flg==1 ) break;
for (i=0; i<3; i++) { digitalWrite(led_1[7-i*3], HIGH); ir_pt1(); }
if ( e_flg==1 ) break;
for (i=0; i<3; i++) { digitalWrite(led_1[i*3+1], LOW); ir_pt1(); }
if ( e_flg==1 ) break;
goto c2;

```

case 3:// \*\*\*\*\* MODE ( 3 )...音楽演奏 1

```

co=2;
c3:// * c3 *
// teunit 16分音符の長さ(ms)
// te 拍子(2/4 のときは 8, 3/4 のときは 12, 4/4 のときは 16)
// len 楽譜サイズ
// oto[] nag[] 楽譜データ アドレス
// ex. oto[note] が DO なら ==> oto[note] = 14 ==> onkai[ oto[note] ] = 262 (Hz)
cur = 0;
for (note = 0; note < len1; note++) { dur = teunit1 * nag1[note]; cur += nag1[note];
if (cur >= te1) { digitalWrite(led_1[le], HIGH); cur = 0; le++; if (le >= 8) le = 0; }
else { digitalWrite(led_1[ole], LOW); ole = le; }
tone(out_p, onkai[oto1[note]], dur); ir_pt2();
if ( e_flg==1 ) break; }

digitalWrite(led_1[le], LOW); // set the LED off
digitalWrite(led_1[ole], LOW); // set the LED off
delay(1000);
le = 0; ole = 0;
if ( e_flg==1 ) break;
goto c3;

```

case 4:// \*\*\*\*\* MODE ( 4 )...音楽演奏 2

```

co=2;
c4:// * c4 *
cur = 0;
for (note = 0; note < len2; note++) { dur = teunit2 * nag2[note]; cur += nag2[note];
if (cur >= te2) { digitalWrite(led_1[le], HIGH); cur = 0; le++; if (le >= 8) le = 0; }
else { digitalWrite(led_1[ole], LOW); ole = le; }
tone(out_p, onkai[oto2[note]], dur); ir_pt2();
if ( e_flg==1 ) break; }

digitalWrite(led_1[le], LOW); // set the LED off
digitalWrite(led_1[ole], LOW); // set the LED off
delay(1000);
le = 0; ole = 0;
if ( e_flg==1 ) break;
goto c4;

```

case 5:// \*\*\*\*\* MODE ( 5 )...音楽演奏 3

```

co=2;
c5:// * c5 *
cur = 0;
for (note = 0; note < len3; note++) { dur = teunit3 * nag3[note]; cur += nag3[note];
if (cur >= te3) { digitalWrite(led_1[le], HIGH); cur = 0; le++; if (le >= 8) le = 0; }
else { digitalWrite(led_1[ole], LOW); ole = le; }
tone(out_p, onkai[oto3[note]], dur); ir_pt2();

```

```

if ( e_flg==1 ) break; }

digitalWrite(led_1[le], LOW); // set the LED off
digitalWrite(led_1[ole], LOW); // set the LED off
delay(1000);
le = 0; ole = 0;
if ( e_flg==1 ) break;
goto c5;
}
goto st1;
}

```

//////// pitches4.h の内容 //////////////////////////////////////

/\*\*\*\*\*

\* Public Constants

1 OCT UP..... u

1 OCT DOWN.... d

DO RE MI FA SO RA SI R

HALF UP ..... s

HALF DOWN .... f

\*\*\*\*\*/

#define DO2 65

#define DOs2 69

#define RE2 73

#define REs2 78

#define MI2 82

#define FA2 87

#define FAs2 93

#define SO2 98

#define SOs2 104

#define RA2 110

#define RAs2 117

#define SI2 123

#define R 0

#define dDO 1 // 131 // == DO3

#define dDOs 2 // 139

#define dRE 3 // 147

#define dREs 4 // 156

#define dMI 5 // 165

#define dFA 6 // 175

#define dFAs 7 // 185

#define dSO 8 // 196

#define dSOs 9 // 208

#define dRA 10 // 220

#define dRAs 11 // 233

#define dSI 12 // 233

#define dSI 13 // 247

#define DO 14 // 262 // == DO4

#define DOs 15 // 277

#define RE 16 // 294

#define REs 17 // 311



```
#define MI 18 // 330
#define FA 19 // 349
#define FAs 20 // 370
#define SO 21 // 392
#define SOs 22 // 415
#define RA 23 // 440
#define RAs 24 // 466
#define SIf 25 // 466
#define SI 26 // 494
#define uDO 27 // 523 // == DO5
#define uDOs 28 // 554
#define uRE 29 //587
#define uREs 30 // 622
#define uMI 31 // 659
#define uFA 32 // 698
#define uFAs 33 //740
#define uSO 34 // 784
#define uSOs 35 // 831
#define uRA 36 // 880
#define uRAs 37 // 932
#define uSIf 38 // 932
#define uSI 39 // 988
```

```
#define DO6 1047
#define DOs6 1109
#define RE6 1175
#define REs6 1245
#define MI6 1319
#define FA6 1397
#define FAs6 1480
#define SO6 1568
#define SOs6 1661
#define RA6 1760
#define RAs6 1865
#define SI6 1976
#define DO7 2093
#define DOs7 2217
#define RE7 2349
#define REs7 2489
#define MI7 2637
#define FA7 2794
#define FAs7 2960
#define SO7 3136
#define SOs7 3322
#define RA7 3520
#define RAs7 3729
#define SI7 3951
#define DO8 4186
#define DOs8 4435
#define RE8 4699
#define REs8 4978
```