

```

import processing.serial.*;
Serial myPort;
PrintWriter output;
/*===== 変数宣言 =====*/
int x; //図形の X 座標の変数を用意
int NUM = 6; //センサーの数
int[] receivedData = new int[NUM]; //センサーの値を格納する配列
int secondIcntM = 0; // OUT コース:分
int secondIcntS = 0; // OUT コース:秒
int secondITIME = 0; // OUT コース:センチ秒
int inTime_m = 0; // IN コース:分
int inTime_s = 0; // IN コース:秒
int inTime_c = 0; // IN コース:センチ秒

int BsecondIcntM = 0; // OUT コース:分
int BsecondIcntS = 0; // OUT コース:秒
int BsecondITIME = 0; // OUT コース:センチ秒
int BinTime_m = 0; // IN コース:分
int BinTime_s = 0; // IN コース:秒
int BinTime_c = 0; // IN コース:センチ秒

int couseoutIN = 0;
int couseoutOUT = 0;
int cntIN = 0;
int cntOUT = 0;

void setup(){
    // 画面サイズ
    size(1080, 780); // 1024 x 768 だと下が切れる 1080*780 size(1080,
780);
    textFont(createFont("x", 15)); // 日本語を表示用。バージョンアップで不要?
    myPort = new Serial(this,"COM8",250000);
}

void draw(){
    // 受信バッファから描画用変数に格納
    secondIcntM = receivedData[2];
    secondIcntS = receivedData[3];
    secondITIME = receivedData[0];

```

```

inTime_m = receivedData[4];
inTime_s = receivedData[5];
inTime_c = receivedData[1];

// 分,秒,センチ秒のバラバラのデータを時計形式に結合
String outTime = nf(secondIcntM, 2)+":"+nf(secondIcntS, 2)+":"+nf(secondITIME, 2);
String inTime = nf(inTime_m, 2)+":"+nf(inTime_s, 2)+":"+nf(inTime_c, 2);
println(outTime);          // デバッグ用に出力
println(inTime);          // デバッグ用に出力

background(255);          // 背景を指定色:白で塗りつぶす

//===OUT コースタイム表示===
if(couseoutOUT == 0){
  textSize(90);           // 文字サイズ 90
  fill(0, 0, 255);        // 文字色:BLUE
  text("OUT", 30, height*15/100); // 表示するテキスト, x座標, y座標 30,15/100
}
if(couseoutOUT == 1){
  textSize(90);           // 文字サイズ 90
  fill(0, 0, 255);        // 文字色:BLUE
  text("OUT コースアウト", 30, height*15/100); // 表示するテキスト, x座標, y座標
30,15/100
}
textSize(220);           // 文字サイズ 220
fill(0, 0, 255);        // 文字色:BLUE
text(outTime, 110, height*40/100); //80,40/100

//===IN コースタイム表示===
if(couseoutIN == 0){
  textSize(90);           // 文字サイズ 90
  fill(255, 0, 0);        // 文字色:RED
  text("IN", 30, height*65/100); // 表示するテキスト, x座標, y座標 30,65/100
}
if(couseoutIN == 1){
  textSize(90);           // 文字サイズ 90
  fill(255, 0, 0);        // 文字色:RED

```

```

    text("IN コースアウト", 30, height*65/100); // 表示するテキスト, x 座標, y 座標
30,65/100
}
textSize(220);          // 文字サイズ 220
fill(255, 0, 0);       // 文字色:RED
text(inTime, 110, height*90/100);//80,90/100

cntIN++;
cntOUT++;

if(secondIcntM != BsecondIcntM || secondIcntS != BsecondIcntS || secondITIME !=
BsecondITIME || inTime_m != BinTime_m || inTime_s != BinTime_s || inTime_c !=
BinTime_c){
    output.println("OUT"+ outTime + "    ;    " + "IN" +inTime);
    output.flush();
    BsecondITIME = secondITIME;
    BinTime_c = inTime_c;
    BsecondIcntM = secondIcntM;
    BsecondIcntS = secondIcntS;
    BinTime_m = inTime_m;
    BinTime_s = inTime_s;
}
}

void serialEvent(Serial p){
    if ( myPort.available() >= 7 ) { // ヘッダ + 時間データ で合計 7 バイト
        if ( myPort.read() == 'H' ) { // ヘッダ文字を見つけたところから読み取る
            receivedData[0] = myPort.read(); // OUT:分読み込み
            receivedData[1] = myPort.read(); // OUT:秒読み込み
            receivedData[2] = myPort.read(); // OUT:センチ秒読み込み
            receivedData[3] = myPort.read(); // IN:分読み込み
            receivedData[4] = myPort.read(); // IN:秒読み込み
            receivedData[5] = myPort.read(); // IN:センチ秒読み込み
        }
    }
}

void keyPressed(){
    if(key == 'q'){
        String filename = nf(year(),4) + nf(month(),2) + nf(day(),2) + nf(hour(),2) +
nf(minute(),2) ;

```

```
// 新しいファイルを生成
output = createWriter( filename + ".csv");
}
if(key == 'w'){
    output.close();
}
if(key == 'i'){
    if(couseoutIN == 0 && cntIN > 10){
        cntIN = 0;
        couseoutIN = 1;
    }
    if(couseoutIN == 1 && cntIN > 10){
        cntIN = 0;
        couseoutIN = 0;
    }
}
if(key == 'o'){
    if(couseoutOUT == 0 && cntOUT > 10){
        cntOUT = 0;
        couseoutOUT = 1;
    }
    if(couseoutOUT == 1 && cntOUT > 10){
        cntOUT = 0;
        couseoutOUT = 0;
    }
}
}
```