



```

char val; // data received from the serial port

#include <RCSwitch.h> // library for the RF Transmitter
#include <Stepper.h> // library for the step motor
#include <Wire.h> // library for the pulse sensor

RCSwitch mySwitch = RCSwitch();

// ***** Stepper ***** //

int stepsPerRevolution = 83; // onerevolution (2048) = 24,2 cm ; 83 = 1cm

Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11); //pins for the step motor

float dis; // location in cm that the ball will move to
float old_dis = 0; // previous location, here starting at 0 cm
float way; // distance between the dis and old_dis
float rate = 1; // value to scale the metric values to suit the size of the aquarium

// ***** Flowsensor ***** //

volatile int flow_frequency; // measures flow sensor pulsesunsigned
int l_minute;
unsigned char flowsensor = 2; // sensor Input
unsigned long currentTime;
unsigned long cloopTime;

// Interrupt function
void flow () {
flow_frequency++;
}

// ***** Modes ***** //

int co; // cardiac output
int hr; // value of the heart rate sensor in bpm
float wf; // value of the water sensor in l/min
int sv; // stroke volume in l

// ***** Setup ***** //

void setup() {

mySwitch.enableTransmit(13); // transmitter is connected to Arduino Pin #10
mySwitch.setPulseLength(395); // optional set pulse length

myStepper.setSpeed(60); // set the speed at 60 rpm
//old_dis = 0; // set initial old_dis = 0

Serial.begin(9600); // communication rate
Wire.begin(); // initialize heartrate sensor

mySwitch.send(16776961, 24);
mySwitch.send(16776970, 24);

// ***** Flowsensor ***** //

pinMode(flowsensor, INPUT);
digitalWrite(flowsensor, HIGH); // optional Internal Pull-Up
attachInterrupt(0, flow, RISING); // setup Interrupt
sei(); // enable interrupts
currentTime = millis();
cloopTime = currentTime;

```

```

// ***** Handshake processing ***** //
establishContact(); // send a byte to establish contact until receiver responds
}

// ***** Loop ***** //

void loop(){

// ***** Heartrate ***** //

Wire.requestFrom(0xA0 >> 1, 1); // request 1 bytes from slave device
Wire.available();
unsigned char c = Wire.read(); // receive heart rate value (a byte)
delay(50);
int(hr) = c;

// ***** Mode: heartrate ***** //

if(val == '1') {
// the light blinks by switching it on and off
mySwitch.send(16776963, 24); // send the code for switching the light off
mySwitch.send(16776970, 24); // send the code for switching the light red
mySwitch.send(16776961, 24); // send the code for switching the light on
delay(hr);

dis = 0; // returns the ball to the starting position once the program has run once

Serial.write(hr); // sends the heart rate to the processing screen which displays it
}

if (Serial.available() > 0) { // if data is available to read,
val = Serial.read(); // read it and store it in val

// ***** Flowsensor ***** //

currentTime = millis();

if(currentTime >= (cloopTime + 1000)){ // every second, calculate and print litres/hour

cloopTime = currentTime; // updates cloopTime
l_minute = (flow_frequency * 1 / 7.5); // pulse frequency (Hz) = 7.5Q, Q is flow rate in
l/min.
flow_frequency = 0; // reset Counter
}

// ***** Mode: cardiac ***** //

if (val == '2'){

mySwitch.send(16776961, 24); // send the code for switching the light on
mySwitch.send(16776970, 24); // send the code for switching the light red

sv = 2;
co = hr * sv;
dis = co * 0.02;

Serial.write(hr); // sends the heart rate to the processing screen which displays it
}
}

```

```

else if (val == '3'){

    mySwitch.send(16776961, 24); // send the code for switching the light on
    mySwitch.send(16776970, 24); // send the code for switching the light red

    sv = 6;
    co = hr * sv;
    dis = co * 0.02;

    Serial.write(hr); // sends the heart rate to the processing screen which displays it
}

else if (val == '4'){

    mySwitch.send(16776961, 24); // send the code for switching the light on
    mySwitch.send(16776970, 24); // send the code for switching the light red

    sv = 8.5;
    co = hr * sv;
    dis = co * 0.02;

    Serial.write(hr); // sends the heart rate to the processing screen which displays it
}

else if (val == '5'){

    mySwitch.send(16776961, 24); // send the code for switching the light on
    mySwitch.send(16776970, 24); // send the code for switching the light red

    sv = 6.5;
    co = hr * sv;
    dis = co * 0.02;

    Serial.write(hr); // sends the heart rate to the processing screen which displays it
}

// ***** Mode: waterflow ***** //

else if (val == '6'){

    mySwitch.send(16776961, 24); // send the code for switching the light on
    mySwitch.send(16776972, 24); // send the code for switching the light cobalt blue

    wf = 1_minute;
    dis = wf * 2;

    hr = wf;
    Serial.write(hr); // sends the flow rate to the processing screen which displays it,
its called hr as it acts like the data from the heart rate
}

// ***** Mode: waterloss ***** //

else if (val == '7'){

    mySwitch.send(16776961, 24); // send the code for switching the light on
    mySwitch.send(16776977, 24); // send the code for switching the light cyan blue
    dis = 3.3;
}

```

```

// ***** Stepper ***** //

// the stepper motor can only achieve 600 steps at once which is 7.2 cm with
// our string roll so we divide the distance in brackets of 7.2 cm

dis = dis * rate;
way = dis - old_dis;

if (dis < 7.2){ // max rotation = 7,2 cm
  myStepper.step(stepsPerRevolution*way);
}

else if (dis > 7.2 && dis < 14.4){
  myStepper.step(stepsPerRevolution*way/2);
  myStepper.step(stepsPerRevolution*way/2);
}

else if (dis > 14.4 && dis < 21.6){
  myStepper.step(stepsPerRevolution*way/3);
  myStepper.step(stepsPerRevolution*way/3);
  myStepper.step(stepsPerRevolution*way/3);
}

else if (dis > 21.6 && dis < 28.8){
  myStepper.step(stepsPerRevolution*way/4);
  myStepper.step(stepsPerRevolution*way/4);
  myStepper.step(stepsPerRevolution*way/4);
  myStepper.step(stepsPerRevolution*way/4);
}

else if (dis > 28.8 && dis < 36.0){
  myStepper.step(stepsPerRevolution*way/5);
  myStepper.step(stepsPerRevolution*way/5);
  myStepper.step(stepsPerRevolution*way/5);
  myStepper.step(stepsPerRevolution*way/5);
  myStepper.step(stepsPerRevolution*way/5);
}

old_dis = dis; // sets the location to the old location for the next run
}
}

// ***** Establish ***** //

void establishContact() {
  while (Serial.available() <= 0) {
    Serial.println("A");
  }
}
}

```

```
import processing.serial.*;
Serial myPort;

boolean firstContact = false;
String val;
float myVal;

PImage base;

PImage heart;
PImage child;
PImage teen;
PImage adult;
PImage elderly;
PImage spring;
PImage groundwater;

PImage up;
PImage left;
PImage down;
PImage right;

boolean event_up = false;
boolean event_right = false;
boolean event_down = false;
boolean event_left = false;

boolean event_right_child = false;
boolean event_right_teen = false;
boolean event_right_adult = false;
boolean event_right_elderly = false;

// ***** SETUP ***** //

void setup() {

  String portName = Serial.list()[1];
  myPort = new Serial(this, Serial.list()[1], 9600);
  myPort.bufferUntil('\n');

  size(850, 850);
  background(255);
  cursor(CROSS);

  base = loadImage("interf_base.png");

  heart = loadImage("interf_heart.png");
  child = loadImage("interf_child.png");
  teen = loadImage("interf_teen.png");
  adult = loadImage("interf_adult.png");
  elderly = loadImage("interf_elderly.png");
  spring = loadImage("interf_spring.png");
  groundwater = loadImage("interf_groundwater.png");

  up = loadImage("interf_up.png");
  right = loadImage("interf_right.png");
  down = loadImage("interf_down.png");
  left = loadImage("interf_left.png");

}
```

```

// ***** DRAW ***** //

void draw() {
  imageMode(CENTER);
  image(base, width/2, height/2, 750, 750);

  // ***** button 'heart' ***** //

  if (dist(mouseX, mouseY, 428, 153) < 40 ) {
    image(heart, width/2, height/2, 750, 750);
  }

  if (event_up) {
    image(up, width/2, height/2, 750, 750);

    if (myPort.available() > 0) {
      myVal = myPort.read();
    }

    textSize(20);
    fill(177,68,32);
    text(int(myVal) + " bpm", width/2-35, height/2-47);
  }

  // ***** button 'age' ***** //

  if (dist(mouseX, mouseY, 676, 397) < 25 ) {
    image(child, width/2, height/2, 750, 750);
  }

  if (dist(mouseX, mouseY, 731, 397) < 25 ) {
    image(teen, width/2, height/2, 750, 750);
  }

  if (dist(mouseX, mouseY, 676, 452) < 25 ) {
    image(adult, width/2, height/2, 750, 750);
  }

  if (dist(mouseX, mouseY, 731, 452) < 25 ) {
    image(elderly, width/2, height/2, 750, 750);
  }

  // . . . . .

  if (event_right) {
    image(right, width/2, height/2, 750, 750);
  }

  if (event_right_child){

    image(child, width/2, height/2, 750, 750);

    if (myPort.available() > 0) {
      myVal = myPort.read();
      myVal = myVal * 0.02;
    }

    textSize(20);
    fill(177,68,32);
    text(myVal + " l/m", width/2-40, height/2-47);
  }
}

```

```

if (event_right_teen){

    image(teen, width/2, height/2, 750, 750);

    if (myPort.available() > 0) {
        myVal = myPort.read();
        myVal = myVal * 0.06;
    }

    textSize(20);
    fill(177,68,32);
    text(myVal + " l/m", width/2-40, height/2-47);
}

if (event_right_adult){

    image(adult, width/2, height/2, 750, 750);

    if (myPort.available() > 0) {
        myVal = myPort.read();
        myVal = myVal * 0.085;
    }

    textSize(20);
    fill(177,68,32);
    text(myVal + " l/m", width/2-40, height/2-47);
}

if (event_right_elderly){

    image(elderly, width/2, height/2, 750, 750);

    if (myPort.available() > 0) {
        myVal = myPort.read();
        myVal = myVal * 0.065;
    }

    textSize(20);
    fill(177,68,32);
    text(myVal + " l/m", width/2-40, height/2-47);
}

// ***** button 'spring' ***** //

if (dist(mouseX, mouseY, 427, 688) < 40 ) {
    image(spring, width/2, height/2, 750, 750);
}

if (event_down) {

    image(down, width/2, height/2, 750, 750);

    if (myPort.available() > 0) {
        myVal = myPort.read();
    }

    textSize(20);
    fill(77,145,167);
    text(myVal + " l/m", width/2-40, height/2-47);
}

```



```

// ***** button 'groundwater' ***** //

if (dist(mouseX, mouseY, 144, 425) < 40 ) {
  image(groundwater, width/2, height/2, 750, 750);
}

if (event_left) {
  image(left, width/2, height/2, 750, 750);

  textSize(20);
  fill(77,145,167);
  text("5.34 l/m", width/2-40, height/2-47);
}

}

// ***** MOUSE ***** //

void mousePressed(){

// ***** HR & '1' ***** //

if (dist(mouseX, mouseY, 428, 153) < 40 ) {

  event_up = true;

  myPort.write('1');
  println('1');

  event_right=false;
  event_down=false;
  event_left=false;
  event_right_child = false;
  event_right_teen = false;
  event_right_adult = false;
  event_right_elderly = false;
}

// ***** SV & '2,3,4,5' ***** //

if (
  || dist(mouseX, mouseY, 676, 397) < 25
  || dist(mouseX, mouseY, 731, 397) < 25
  || dist(mouseX, mouseY, 676, 452) < 25
  || dist(mouseX, mouseY, 731, 452) < 25 ) {

  event_right = true;

  event_up=false;
  event_down=false;
  event_left=false;
}

if (dist(mouseX, mouseY, 676, 397) < 25){

  event_right_child = true;

  myPort.write('2');
  println('2');

  event_right_teen = false;
  event_right_adult = false;
  event_right_elderly = false;
}
}

```

```
if (dist(mouseX, mouseY, 731, 397) < 25){

    event_right_teen = true;

    myPort.write('3');
    println('3');

    event_right_child = false;
    event_right_adult = false;
    event_right_elderly = false;
}

if (dist(mouseX, mouseY, 676, 452) < 25){

    event_right_adult = true;

    myPort.write('4');
    println('4');

    event_right_child = false;
    event_right_teen = false;
    event_right_elderly = false;
}

if (dist(mouseX, mouseY, 731, 452) < 25){

    event_right_elderly = true;

    myPort.write('5');
    println('5');

    event_right_child = false;
    event_right_teen = false;
    event_right_adult = false;
}

// ***** WF & '6' ***** //

if(dist(mouseX, mouseY, 427, 688) < 40 ) {

    event_down = true;

    myPort.write('6');
    println('6');

    event_up = false;
    event_right = false;
    event_left = false;
    event_right_child = false;
    event_right_teen = false;
    event_right_adult = false;
    event_right_elderly = false;
}
```

```
// ***** GL & '7' ***** //

if (dist(mouseX, mouseY, 144, 425) < 40 ) {

    event_left = true;

    myPort.write('7');
    println('7');

    event_up = false;
    event_right = false;

    event_down = false;
    event_right_child = false;
    event_right_teen = false;
    event_right_adult = false;
    event_right_elderly = false;
}

}

// ***** CONTACT ***** //

void serialEvent(Serial myPort) {

val = myPort.readStringUntil('\n');

if (val != null) {

    val = trim(val);
    println(val);

    if (firstContact == false) {

        if (val.equals("A")) {
            myPort.clear();
            firstContact = true;
            myPort.write("A");
            println("contact");
        }

    } else {
        println(val);
        myPort.write("A");
    }

}

}
```