

# Processing code

```

/*
  This program gives an idea of how an app, that supports the MeatMe, could look.
  In the App we display multiple things: the temperature, humidity and value measured
  by the Figaro TGS 822. The temperature and the humidity are displayed next to icons
  to make the app more clear for the user. The value measured by the sensor is
  projected using a meatbar. This meatBar will fill itself according to the
  "condition" of the meat; the higher the value measured by the sensor, the bigger
  part of the meatbar will be coloured. Next to that, the colour of the meatbar will
  also change according to the value measured by the sensor. If the meat is edible,
  the bar will be green, if the meat is close to spoil, the bar will be orange, and
  last, if the meat is spoiled, the bar will be red.
  Made by: Team Hummus
*/

import processing.serial.*;

Serial port;
App app;

int AMOUNT = 3; // The amount of values (sensors) the Arduino sends.
String buff = "";
char header[] = {'T', 'H', 'S'};
int value[] = new int[AMOUNT];
int diffValue[] = new int[AMOUNT];
int NEWLINE = 10;

void setup() {
  size(1500, 900);
  app = new App();
  for (int i = 0; i < Serial.list().length; i++) {
    print "[" + i + "] ";
    println(Serial.list()[i]);
  }
  port = new Serial(this, Serial.list()[0], 9600);
}

void draw() {
  while (port.available() > 0) {
    serialEvent(port.read()); // read data
  }
}

```

```

    app.play(); //display the app.
}

// This function collects the input from the Arduino.
void serialEvent(int serial)
{
    try {        // try-catch because of transmission errors
        if (serial != NEWLINE) {
            buff += char(serial);
        } else {
            // The first character tells us which axis this value is for
            char c = buff.charAt(0);
            // Remove it from the string
            buff = buff.substring(1);
            // Discard the carriage return at the end of the buffer
            buff = buff.substring(0, buff.length()-1);
            // Parse the String into an integer
            for (int z=0; z<AMOUNT; z++) {
                if (c == header[z]) {
                    value[z] = Integer.parseInt(buff);
                }
            }
            buff = "";          // Clear the value of "buff"
        }
    }
    catch(Exception e) {
        println("No valid data");
    }
}

```

```

// This Class handles the app, so makes sure the pictures and values are displayed.
class App {
    color red = color(255, 73, 69);
    color orange = color(246, 163, 65);
    color green = color(111, 187, 124);
    color black = color(0);
    color white = color (255);
    PImage background; //The background of the app.
    PImage thermometer; //Picture of a thermometer to indicate the temperature.
    PImage water; //Picture of an waterdrup to indicate the humidity.
    PImage meatLevel; //Picture of the text "(M)eat Level"
    PImage meatMe; //Picture of the text "MeatMe"
    PImage logo; //Picture of our logo.
}

```

```
int temperature = 0; //The temperature measured in our prototype.
int humidity = 0; //The humidity measured in our prototype.
int sensor = 0; //The value received from the sensor.
int heightBox = 76; //Height of the meatLevel bar.
int widthBox = 876; //Width of the meatLevel bar.
int topCornerX = 312; //X coordinate of the left top corner of the meatLevel bar.
int topCornerY = 287; //Y coordinate of the left top corner of the meatLevel bar.
```

```
App() {
    addImages();
    textSize(220);
    textAlign(CENTER);
    rectMode(CENTER);
    noStroke();
}
```

// This functions draws all the elements that are displayed in the app.

```
void play() {
    try {
        temperature = value[0];
        humidity = value[1];
        sensor = value[2];
    }
    catch(Exception e) {
        println("No valid data");
    }
    fill(0);
    image(background, 0, 0);
    image(thermometer, 0, 0);
    image(water, 0, 0);
    image(meatLevel, 0, 0);
    image(meatMe, 0, 0);
    image(logo, 0, 0);
    text(humidity, 1180, 730);
    meatLevel(); //Displays the meatLevel bar
    displayTemperature(); //Displays the temperature.
}
```

// This function loads all the images we use in our program. Next to that all the images are resized to the right width and height.

```
void addImages() {
    background = loadImage("background.jpg");
    background.resize(1500, 900);
    thermometer = loadImage("thermometer1.png");
    thermometer.resize(1500, 900);
    water = loadImage("water1.png");
    water.resize(1500, 900);
}
```

```

    meatLevel = loadImage("meatlevel.png");
    meatLevel.resize(1500, 900);
    meatMe = loadImage("meatMe.png");
    meatMe.resize(1500, 900);
    logo = loadImage("logo.png");
    logo.resize(1500, 900);
}

/* This function fills the meatLevel according to the value measured by the
sensor. If the value is lower than 475, the meatLevel will be colored green. If the
value is bigger the 550, the meatLevel will be colored red. If the value is in
between, then the meatLevel will be orange. */
void meatLevel() {
    rectMode(CENTER);
    fill(black);
    rect(750, 325, 900, 101); // Draws the black outline of the meatlevel.
    fill(white);
    rectMode(CORNER);
    rect(topCornerX, topCornerY, widthBox, heightBox); // Draws the white inside
of the meatlevel.
    float widthBar = map(sensor, 0, 1000, 0, widthBox); //Calculate width of the
meatLevel bar.
    if (sensor < 475) {
        fill(green); //The meatLevel will be green.
    } else {
        if (sensor > 550) {
            fill(red); //The meatLevel will be red.
        } else {
            fill(orange); //The meatLevel will be orange.
        }
    }
    rect(topCornerX, topCornerY, widthBar, heightBox); // Draws the indication
part of the meatLevel
}

/* This function displays the temperature. The temperature will be displayed
black if temperature is okay (below 7 degrees), otherwise the temperature
will be displayed in red. */
void displayTemperature() {
    fill(black);
    if (temperature > 7) { // Checks if the temperature is bigger than 7
degrees.
        fill(red); //The temperature will be red.
    }
    text(temperature, 380, 730); //Display the temperature
}
}

```

