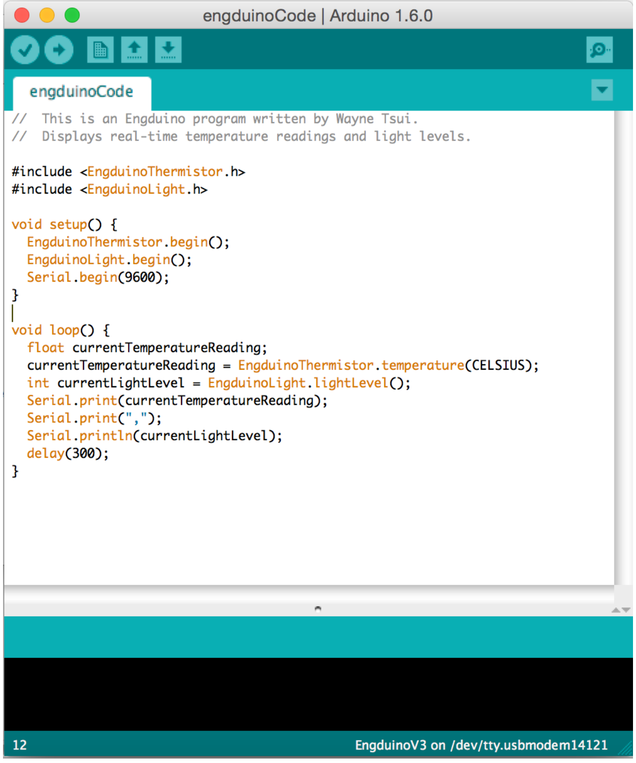
Engduino Application

This Engduino application utilises the temperature and light sensors of the Engduino to send data to the computer. Processing IDE is used to collect the data and represent it in a graphical and informative format. The current temperature readings and light levels are updated in real-time, and the highest and lowest values are recorded as the programme runs.

To set up and run the application:

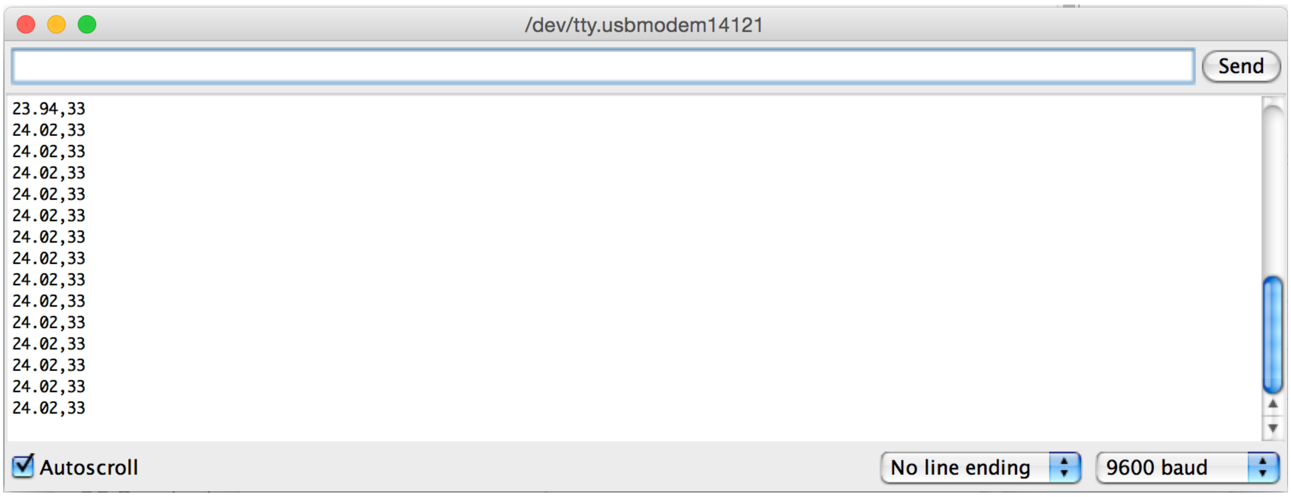
1. Open up the “engduinoCode.ino” file.
2. Choose the correct USB port that Engduino is connected to.
3. Upload the Engduino code.
4. Check that Engduino code is functioning properly by opening the serial monitor, string values should appear.
5. Open up the “displayCode.pde”.
6. Run the Processing sketch.
7. Select the correct serial port by looking at the Processing console and update the correct serial list value in “setup”.
8. Run the sketch again.
9. A display window will appear to show the graphical data of temperature readings and light level.

Screenshots:



```
//  This is an Engduino program written by Wayne Tsui.
//  Displays real-time temperature readings and light levels.

#include <EngduinoThermistor.h>
#include <EngduinoLight.h>

void setup() {
  EngduinoThermistor.begin();
  EngduinoLight.begin();
  Serial.begin(9600);
}

void loop() {
  float currentTemperatureReading;
  currentTemperatureReading = EngduinoThermistor.temperature(CELSIUS);
  int currentLightLevel = EngduinoLight.lightLevel();
  Serial.print(currentTemperatureReading);
  Serial.print(",");
  Serial.println(currentLightLevel);
  delay(300);
}
```

Screenshot 1. Engduino source code.



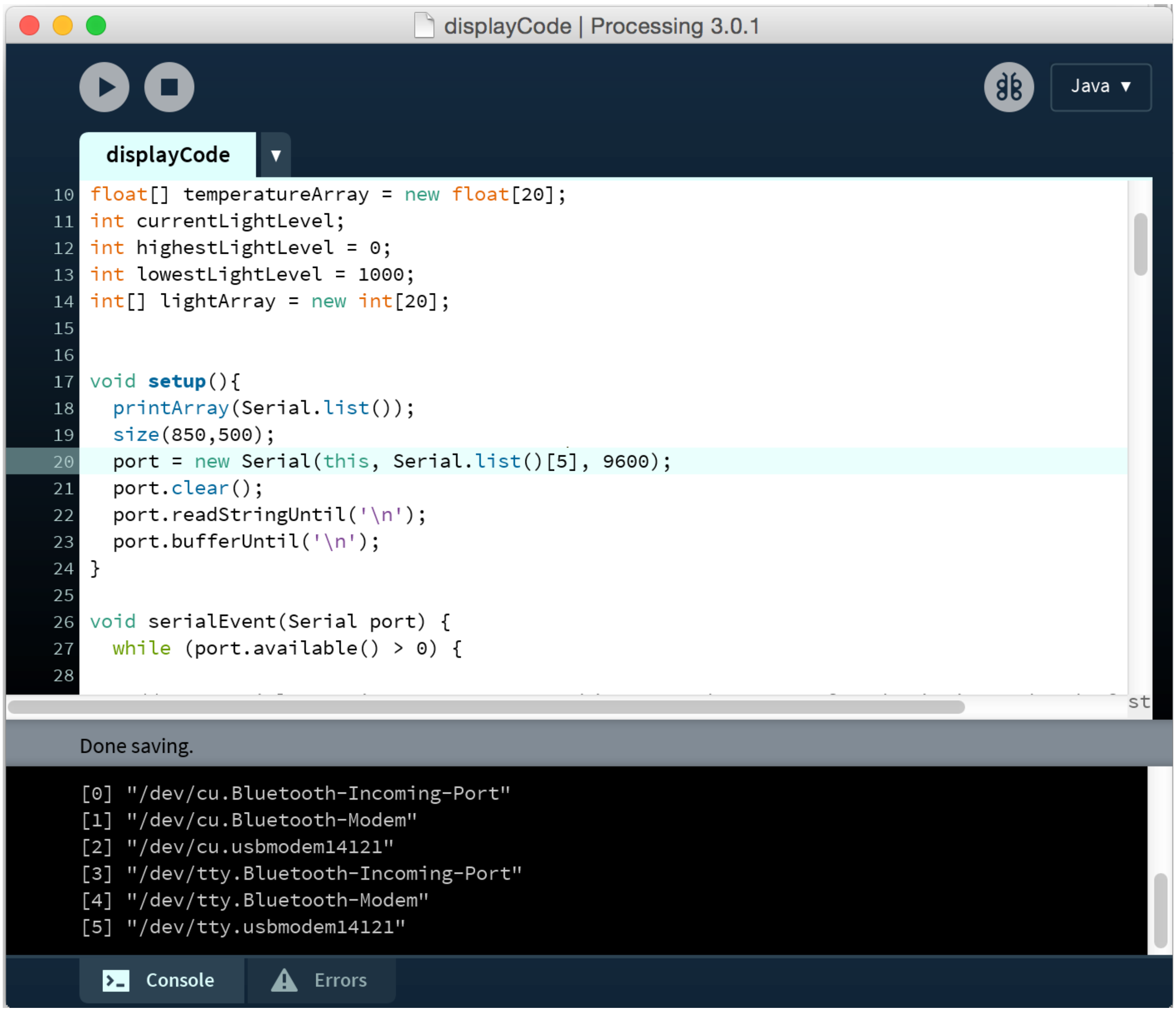
```
23.94,33
24.02,33
24.02,33
24.02,33
24.02,33
24.02,33
24.02,33
24.02,33
24.02,33
24.02,33
24.02,33
24.02,33
24.02,33
24.02,33
24.02,33
```

☑ Autoscroll

No line ending

9600 baud

Screenshot 2. Engduino Serial Monitor.

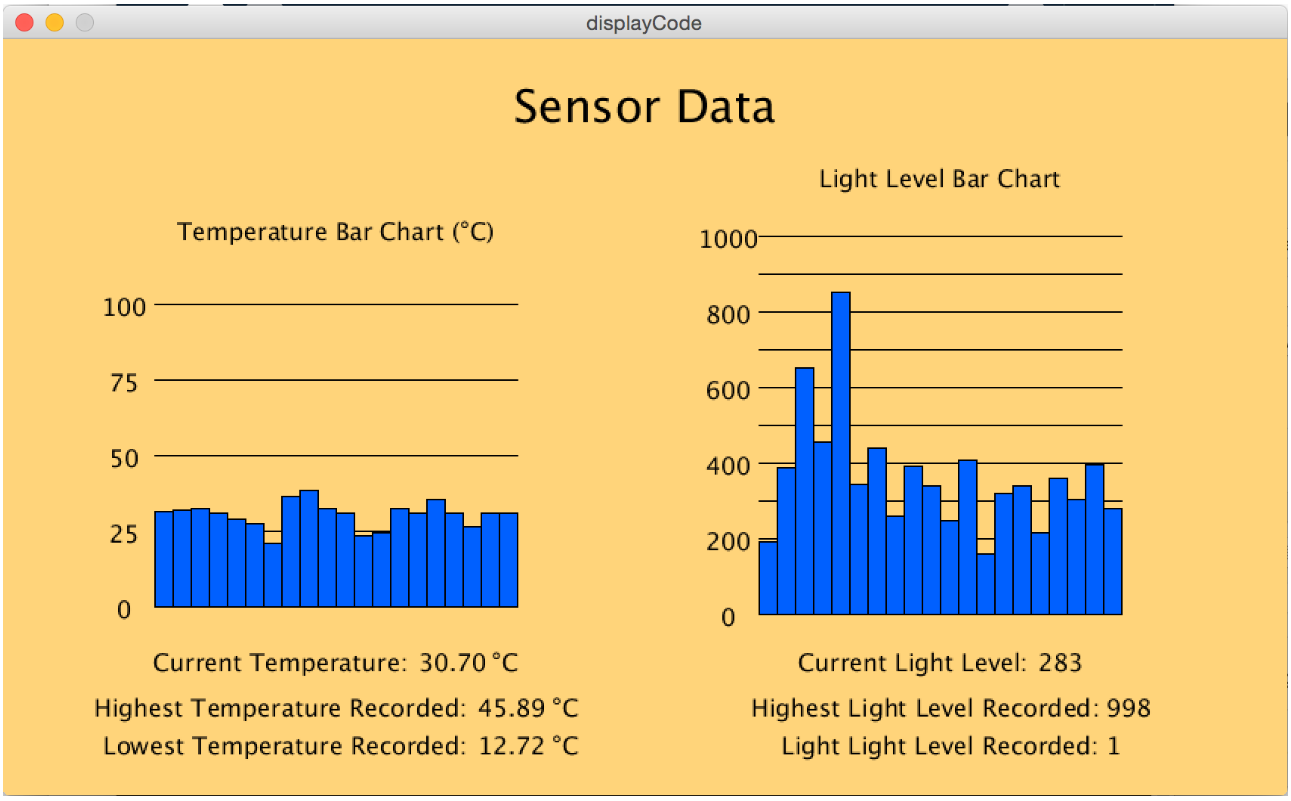


```
float[] temperatureArray = new float[20];
int currentLightLevel;
int highestLightLevel = 0;
int lowestLightLevel = 1000;
int[] lightArray = new int[20];


void setup(){
  printArray(Serial.list());
  size(850,500);
  port = new Serial(this, Serial.list()[5], 9600);
  port.clear();
  port.readStringUntil('\n');
  port.bufferUntil('\n');
}

void serialEvent(Serial port) {
  while (port.available() > 0) {

```

Done saving.

```
[0] "/dev/cu.Bluetooth-Incoming-Port"
[1] "/dev/cu.Bluetooth-Modem"
[2] "/dev/cu.usbmodem14121"
[3] "/dev/tty.Bluetooth-Incoming-Port"
[4] "/dev/tty.Bluetooth-Modem"
[5] "/dev/tty.usbmodem14121"
```

Screenshot 3. Processing IDE source code.



Sensor Data

Current Temperature: 30.70 °C

Highest Temperature Recorded: 45.89 °C

Lowest Temperature Recorded: 12.72 °C

Current Light Level: 283

Highest Light Level Recorded: 998

Light Light Level Recorded: 1

Screenshot 4. Processing IDE Processing Window.