Sarah Wood

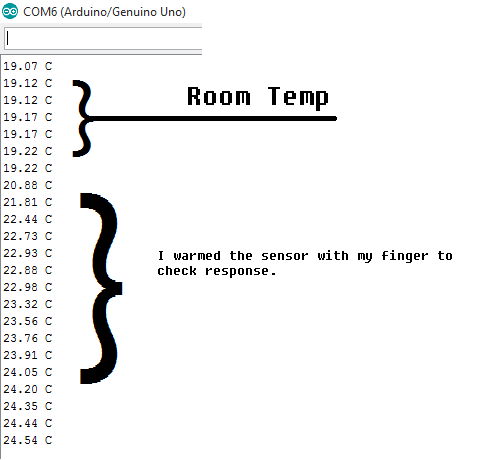
Embedded Programming

Chapter 10

Assignment 1

“Twitter Temperature”

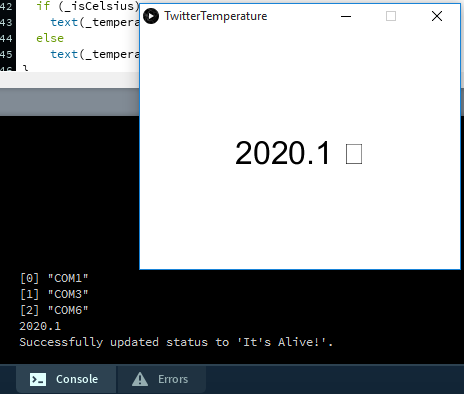
Screen Caps/Pictures



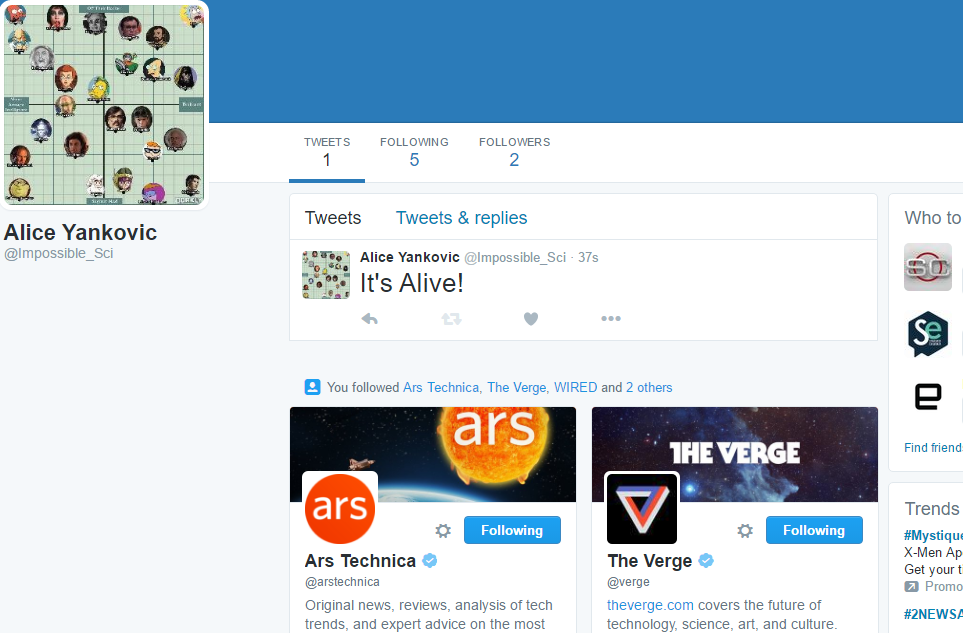
Ok, basic hardware functionality is OK. Getting Twitter4j to work was a bit of a pain. “Drag and drop” libraries didn’t quite work here the way Processing said they should.

<http://codasign.com/tutorials/processing-and-twitter/processing-and-twitter-getting-started/>

was invaluable in walking me through actually making it work.

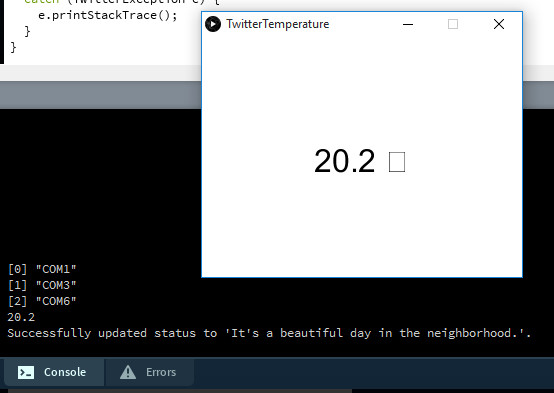


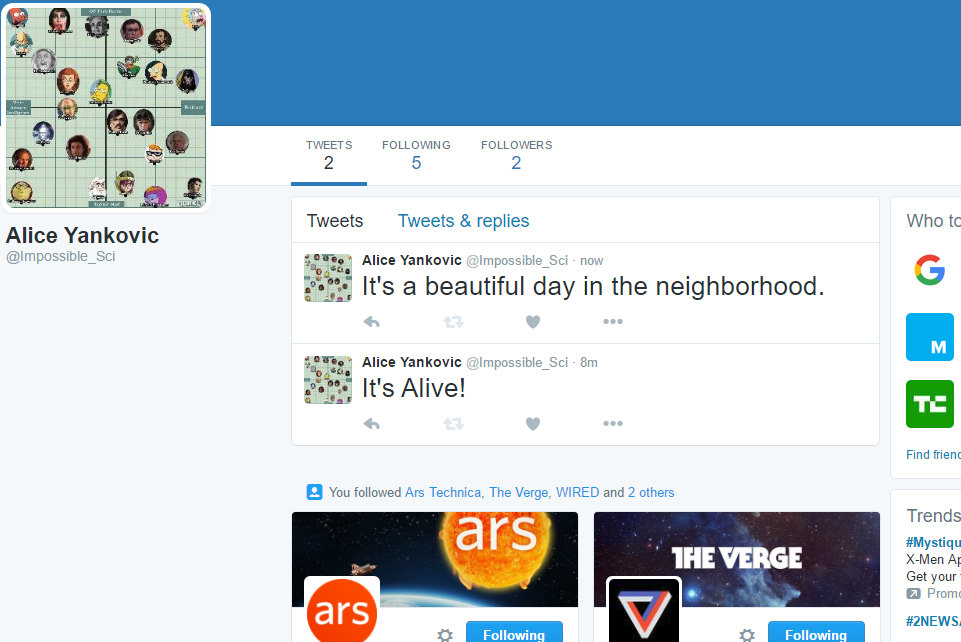
Hmm, 2020.1 degrees. Must be a bug somewhere.



Actual tweet generated. (I’m a Weird Al fan, obviously.)

I tried “turning it off and on again.” (Translation: I couldn’t find anything glaringly wrong, so I recompiled and restarted the sketch.) I changed the MAX\_TEMP parameter and the output phrase. It works.





Source Code

Arduino Code

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

#define CELSIUS

const unsigned int TEMP\_SENSOR\_PIN = 0;

const unsigned int BAUD\_RATE = 9600;

const float SUPPLY\_VOLTAGE = 5.0;

void setup() {

Serial.begin(BAUD\_RATE);

}

void loop() {

const int sensor\_voltage = analogRead(TEMP\_SENSOR\_PIN);

const float voltage = sensor\_voltage \* SUPPLY\_VOLTAGE / 1024;

const float celsius = (voltage \* 1000 - 500) / 100;

#ifdef CELSIUS

Serial.print(celsius);

Serial.println(" C");

#else

Serial.print(9.0 / 5.0 \* celsius + 32.0);

Serial.println(" F");

#endif

delay(5000);

}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Processing Code

(Don’t forget to check the codasign tutorial for Twitter4j if everything doesn’t play nice.)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

import processing.serial.\*;

import twitter4j.conf.\*;

import twitter4j.\*;

import twitter4j.auth.\*;

import twitter4j.api.\*;

import java.util.\*;

final float MAX\_WORKING\_TEMP = 20.0; //I played with this, it was originally 32.0

final int LINE\_FEED = 10;

final int BAUD\_RATE = 9600;

final int FONT\_SIZE = 32;

final int WIDTH = 320;

final int HEIGHT = 240;

final String API\_KEY = "**REDACTED**"; //You’ll have your own access tokens

final String API\_SECRET = "**REDACTED**";

final String ACCESS\_TOKEN = "**REDACTED**";

final String ACCESS\_TOKEN\_SECRET = "**REDACTED**";

Serial \_arduinoPort;

float \_temperature;

boolean \_isCelsius;

PFont \_font;

void setup() {

size(320, 240);

\_font = createFont("Arial", FONT\_SIZE, true);

printArray(Serial.list());

\_arduinoPort = new Serial(this, Serial.list()[2], BAUD\_RATE);

\_arduinoPort.clear();

\_arduinoPort.bufferUntil(LINE\_FEED);

\_arduinoPort.readStringUntil(LINE\_FEED);

}

void draw() {

background(255);

fill(0);

textFont(\_font, FONT\_SIZE);

textAlign(CENTER, CENTER);

if (\_isCelsius)

text(\_temperature + " \u2103", WIDTH / 2, HEIGHT / 2);

else

text(\_temperature + " \u2109", WIDTH / 2, HEIGHT / 2);

}

void serialEvent(Serial port) {

final String arduinoData = port.readStringUntil(LINE\_FEED);

if (arduinoData != null) {

final String[] data = split(trim(arduinoData), ' ');

if (data.length == 2 &&

(data[1].equals("C") || data[1].equals("F")))

{

\_isCelsius = data[1].equals("C");

\_temperature = float(data[0]);

if (Float.isNaN(\_temperature))

return;

println(\_temperature);

int sleepTime = 5 \* 60 \* 1000;

if (\_temperature > MAX\_WORKING\_TEMP) {

tweetAlarm();

sleepTime = 120 \* 60 \* 1000;

}

try {

Thread.sleep(sleepTime);

}

catch(InterruptedException ignoreMe) {}

}

}

}

void tweetAlarm() {

ConfigurationBuilder cb = new ConfigurationBuilder();

cb.setDebugEnabled(true)

.setOAuthConsumerKey(API\_KEY)

.setOAuthConsumerSecret(API\_SECRET)

.setOAuthAccessToken(ACCESS\_TOKEN)

.setOAuthAccessTokenSecret(ACCESS\_TOKEN\_SECRET);

TwitterFactory tf = new TwitterFactory(cb.build());

Twitter twitter = tf.getInstance();

try {

Status status = twitter.updateStatus(

"It's a beautiful day in the neighborhood." //I played with this as well.

);

println(

"Successfully updated status to '" + status.getText() + "'."

);

}

catch (TwitterException e) {

e.printStackTrace();

}

}