#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

#include <RF24\_config.h>

#include <Bridge.h>

#include <Temboo.h>

#include "TembooAccount.h" // contains Temboo account information

/\*\*\* SUBSTITUTE YOUR VALUES BELOW: \*\*\*/

// Note that for additional security and reusability, you could

// use #define statements to specify these values in a .h file.

const String TWITTER\_ACCESS\_TOKEN = "Insert Your Token";

const String TWITTER\_ACCESS\_TOKEN\_SECRET = "Insert Your Token";

const String TWITTER\_API\_KEY = "Insert Your Token";

const String TWITTER\_API\_SECRET = "Insert Your Token";

int numRuns = 1; // execution count, so this doesn't run forever

int maxRuns = 10; // the max number of times the Twitter HomeTimeline Choreo should run

int msg[1];

RF24 radio(9,10);

String theMessage = " ";

char incomingByte = 0;

const uint64\_t pipe = 0xE8E8F0F0E1LL;

void setup() {

Serial.begin(9600);

// For debugging, wait until a serial console is connected.

delay(4000);

while(!Serial);

Bridge.begin();

radio.begin();

radio.openWritingPipe(pipe);

}

void loop()

{

// while we haven't reached the max number of runs...

if (numRuns <= maxRuns) {

Serial.println("Running ReadATweet - Run #" + String(numRuns++));

TembooChoreo HomeTimelineChoreo;

// invoke the Temboo client.

// NOTE that the client must be reinvoked, and repopulated with

// appropriate arguments, each time its run() method is called.

HomeTimelineChoreo.begin();

// set Temboo account credentials

HomeTimelineChoreo.setAccountName(TEMBOO\_ACCOUNT);

HomeTimelineChoreo.setAppKeyName(TEMBOO\_APP\_KEY\_NAME);

HomeTimelineChoreo.setAppKey(TEMBOO\_APP\_KEY);

// tell the Temboo client which Choreo to run (Twitter > Timelines > HomeTimeline)

HomeTimelineChoreo.setChoreo("/Library/Twitter/Timelines/HomeTimeline");

// set the required choreo inputs

// see https://www.temboo.com/library/Library/Twitter/Timelines/HomeTimeline/

// for complete details about the inputs for this Choreo

HomeTimelineChoreo.addInput("Count", "1"); // the max number of Tweets to return from each request

HomeTimelineChoreo.addInput("AccessToken", TWITTER\_ACCESS\_TOKEN);

HomeTimelineChoreo.addInput("AccessTokenSecret", TWITTER\_ACCESS\_TOKEN\_SECRET);

HomeTimelineChoreo.addInput("ConsumerKey", TWITTER\_API\_KEY);

HomeTimelineChoreo.addInput("ConsumerSecret", TWITTER\_API\_SECRET);

// next, we'll define two output filters that let us specify the

// elements of the response from Twitter that we want to receive.

// see the examples at http://www.temboo.com/arduino

// for more on using output filters

// we want the text of the tweet

HomeTimelineChoreo.addOutputFilter("tweet", "/[1]/text", "Response");

// and the name of the author

HomeTimelineChoreo.addOutputFilter("author", "/[1]/user/screen\_name", "Response");

// tell the Process to run and wait for the results. The

// return code will tell us whether the Temboo client

// was able to send our request to the Temboo servers

unsigned int returnCode = HomeTimelineChoreo.run();

// a response code of 0 means success; print the API response

if(returnCode == 0) {

String author; // a String to hold the tweet author's name

String tweet; // a String to hold the text of the tweet

// choreo outputs are returned as key/value pairs, delimited with

// newlines and record/field terminator characters, for example:

// Name1\n\x1F

// Value1\n\x1E

// Name2\n\x1F

// Value2\n\x1E

// see the examples at http://www.temboo.com/arduino for more details

// we can read this format into separate variables, as follows:

while(HomeTimelineChoreo.available()) {

// read the name of the output item

String name = HomeTimelineChoreo.readStringUntil('\x1F');

name.trim();

// read the value of the output item

String data = HomeTimelineChoreo.readStringUntil('\x1E');

data.trim();

// assign the value to the appropriate String

if (name == "tweet") {

tweet = data;

} else if (name == "author") {

author = data;

}

}

Serial.println("@" + author + " - " + tweet);

String steve = ("@" + author + " - " + tweet);

int messageSize = steve.length();

for (int i = 0; i < messageSize; i++){

int charToSend[1];

charToSend[0] = steve.charAt(i);

radio.write(charToSend,1);

}

msg[0] = 2;

radio.write(msg,1);

} else {

// there was an error

// print the raw output from the choreo

while(HomeTimelineChoreo.available()) {

char c = HomeTimelineChoreo.read();

Serial.print(c);

}

}

HomeTimelineChoreo.close();

}

Serial.println("Waiting...");

delay(90000); // wait 90 seconds between HomeTimeline calls

}