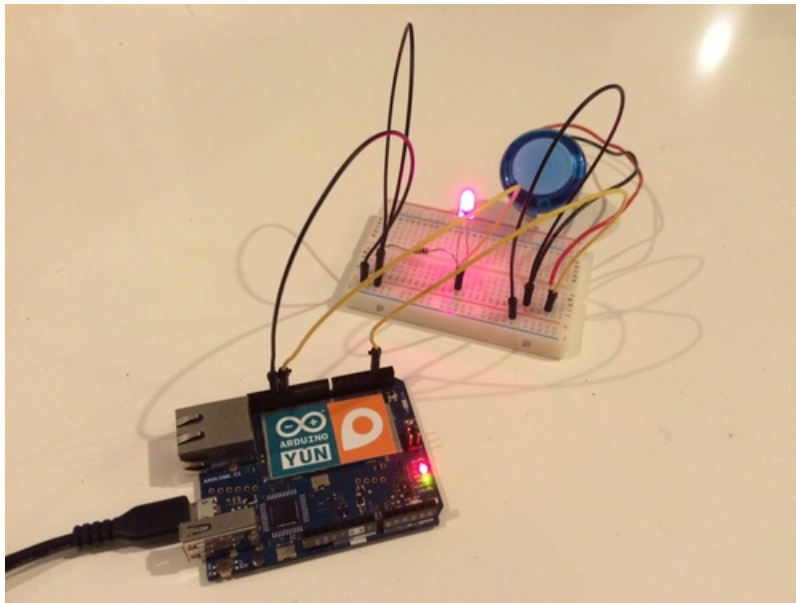


Arduino Yun Temboo Twitter Tracker

Created by Vaughn Shinall



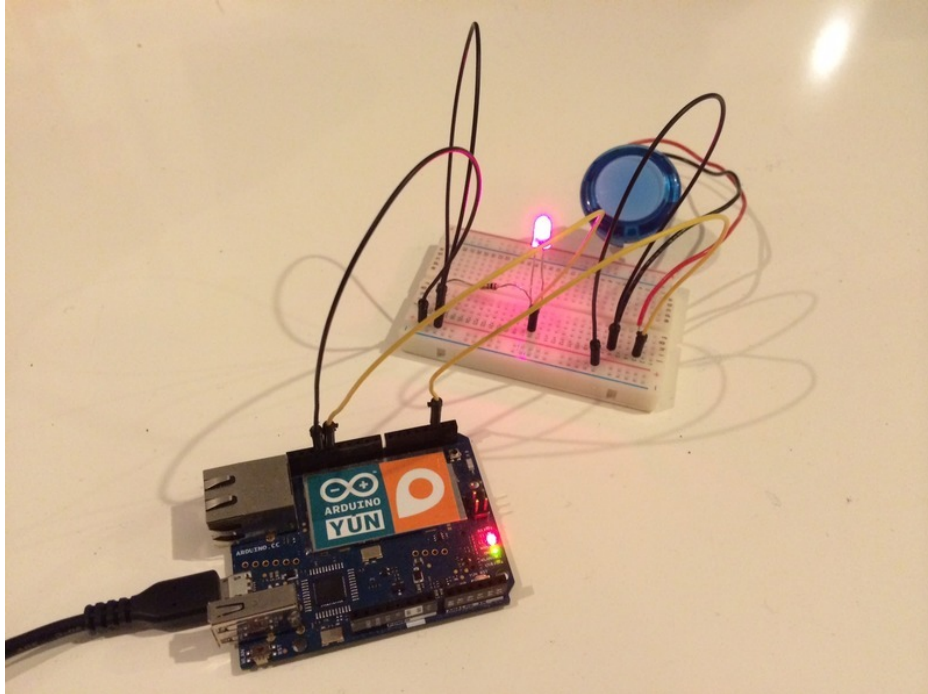
Last updated on 2019-05-10 06:11:10 PM UTC

Overview



The Temboo service has moved away from Arduino support. The service access in this older guide will no longer work.

This guide will introduce you to the basics of programming your Arduino Yún to interact with APIs using the Temboo platform. We'll build a simple device that checks Twitter for user mentions when a button is pressed, and then turns on an LED if there is a new mention. Of course, the same steps we take here can be adapted to work with all sorts of different sensors and any of the different processes in the Temboo Library.



Get Set Up

To build this example project, you'll be using an Arduino Yún, a push button, and an LED. You'll need:



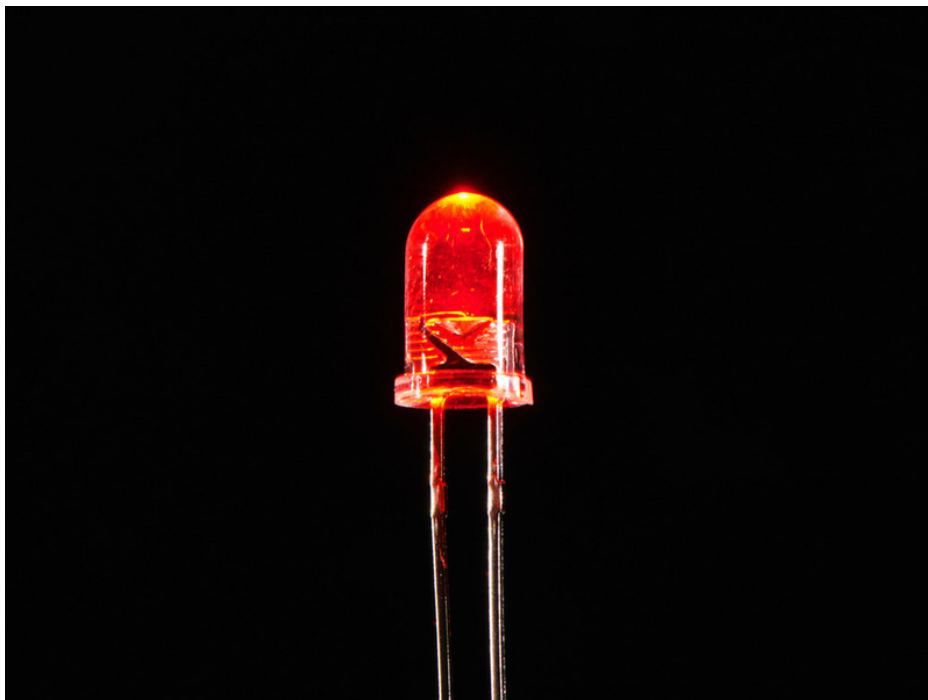
An Arduino Yún



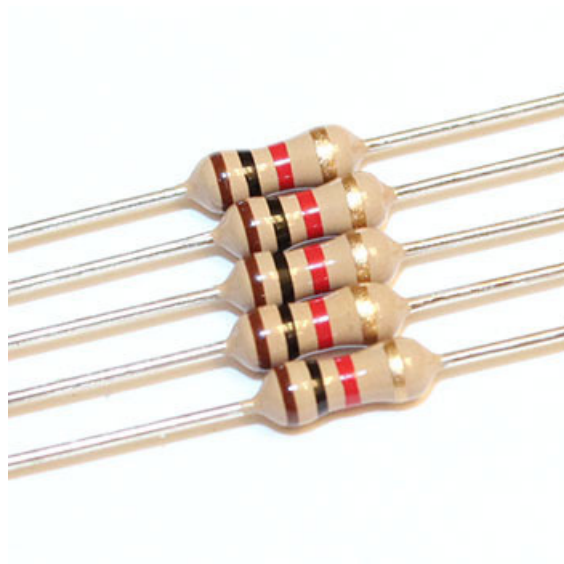
A USB A – Micro B cable



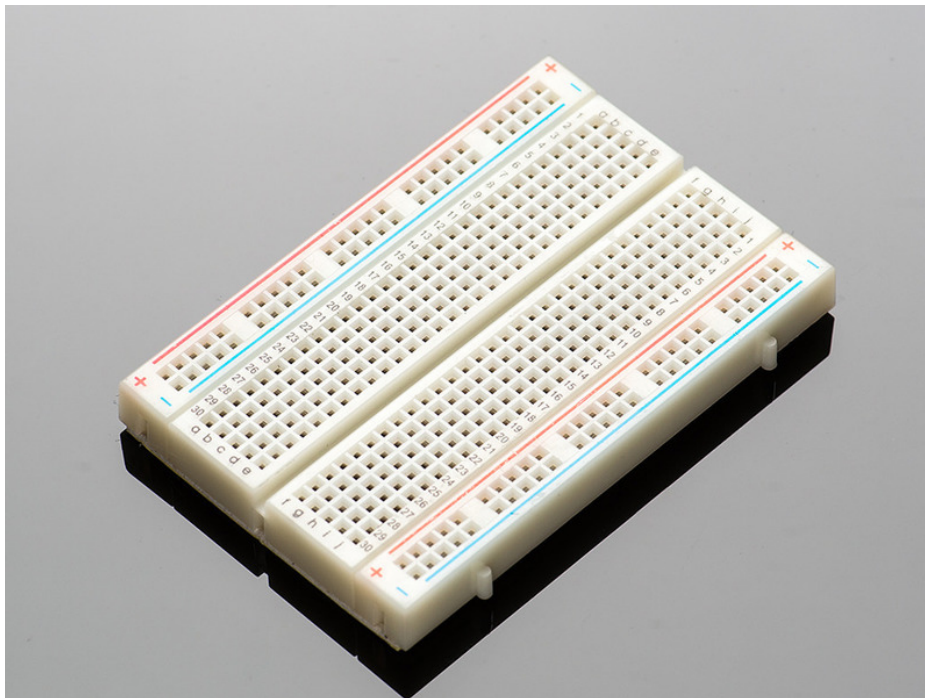
An arcade-style push button



An LED



A 1K Ohm resistor



A half-size breadboard



Some wires

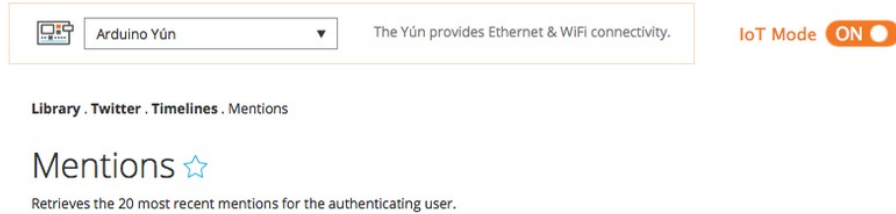
You'll also need a Temboo account and a Twitter account. To set up your Temboo account (if you don't already have one) you can create one for free [here \(https://adafruit.it/e53\)](https://adafruit.it/e53). You can create a Twitter account [here \(https://adafruit.it/cjA\)](https://adafruit.it/cjA), and register a Twitter application using the [Twitter dev console \(https://adafruit.it/cF0\)](https://adafruit.it/cF0). After creating the app, you'll find API keys for that application in the **API Keys** tab; you'll need these later.

Generate Your Sketch



The Temboo service has moved away from Arduino support. The service access in this older guide will no longer work.

Now that you're all set up, log in to Temboo and go to the [Twitter > Timelines > Mentions \(https://adafru.it/eoi\)](#) Choreo in the Temboo Library. Turn on IoT Mode and select **Arduino Yún** from the dropdown menu.



Fill out the **Input** fields using your Twitter keys, and click **Optional Inputs** at the bottom of the **Input** box to reveal a list of more input values that you can pass to the Choreo. Specify “1” for the **Count** input, since we only need to check the most recent tweet to determine whether any new mentions have come in since the last check. Test the Choreo from your browser by clicking **Run** at the bottom of the window. Now, if you scroll down and check the **Response** box under **Output**, you’ll see the JSON response returned by Twitter.

Mentions

Retrieves the 20 most recent mentions for the authenticating user.

INPUT Save Profile

AccessToken
The Access Token provided by Twitter or retrieved during the OAuth process.

AccessTokenSecret
The Access Token Secret provided by Twitter or retrieved during the OAuth process.

ConsumerKey
The API Key (or Consumer Key) provided by Twitter.

ConsumerSecret
The API Secret (or Consumer Secret) provided by Twitter.

OPTIONAL INPUT

ContributorDetails
Set to true to include the screen_name of the contributor. By default only the user_id of the contributor is included.

Count
Specifies the number of tweets to retrieve. Must be less than or equal to 200. Defaults to 20.

IncludeEntities
The "entities" node containing extra metadata will not be included when set to false.

MaxId
Returns results with an ID less than (older than) or equal to the specified ID.

SinceId
Returns results with an ID greater than (more recent than) the specified ID.

TrimUser
When set to true, each tweet returned in a timeline will include a user object including only the status authors numerical ID. Defaults to false, which returns the complete user object.

Run

Upload and Run



The Temboo service has moved away from Arduino support. The service access in this older guide will no longer work.

Now that you've tested the Choreo successfully, scroll down to find the **Code** section of the page. When you're in IoT Mode and you run a Choreo, Temboo automatically generates code that can be used to make the same API call from an Arduino sketch. Copy the code, and paste it into a new sketch in the Arduino IDE.

▼ CODE

Download

```
#include <Bridge.h>
#include <Temboo.h>
#include "TembooAccount.h" // contains Temboo account information, as described below

int numRuns = 1; // Execution count, so this doesn't run forever
int maxRuns = 10; // Maximum number of times the Choreo should be executed

void setup() {
  Serial.begin(9600);

  // For debugging, wait until the serial console is connected.
  delay(4000);
  while(!Serial);
  Bridge.begin();
}

void loop() {
  if (numRuns <= maxRuns) {
```

In order to run this sketch on your Arduino, it needs to be configured with an appropriate `TembooAccount.h` header file that contains your Temboo account information. To create the header file, make a new tab in the Arduino IDE, and name it `TembooAccount.h`.

On the Mentions Choreo page beneath the sketch code that you previously copied and pasted into the Arduino IDE, you'll find another block of generated code containing `#define` statements. This is your header file. Copy the contents of the header into the `TembooAccount.h` tab in your Arduino IDE.

▼ HEADER FILE

```
/*
IMPORTANT NOTE about TembooAccount.h

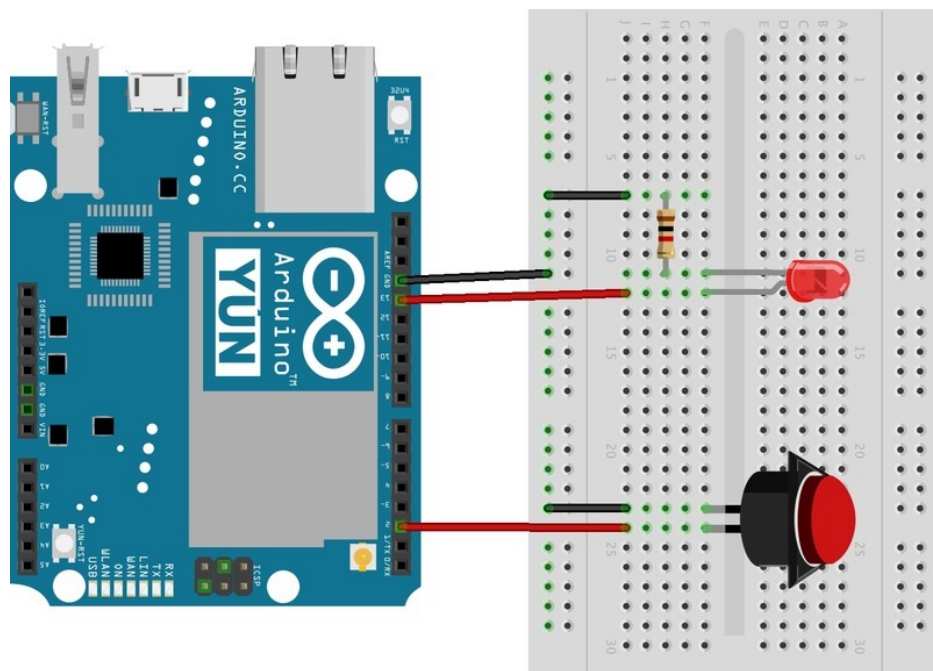
TembooAccount.h contains your Temboo account information and must be included
alongside your sketch. To do so, make a new tab in Energia, call it TembooAccount.h,
and copy this content into it.
*/
```

With both files in place, you are ready to searching Twitter from your Arduino. Make sure your Yún is connected to the Internet by following the directions [here \(https://adafru.it/d1Z\)](https://adafru.it/d1Z). Then, save and upload the sketch to the Yún and open the serial monitor; after a few seconds, the sketch will begin to run. By default we've set the Choreo to run ten times, executing once every thirty seconds.

Wiring Your Circuit

Now that your Yún is successfully interacting with Twitter, let's add the push button and LED. This will require two steps: first, you'll wire the extra components to your Arduino, and second, you'll modify your code slightly so that the Arduino knows when a new mention has come in. We'll begin with the circuit.

The push button has two wires to connect; one should go to ground, and the other should go to the pin you're reading from (we've chosen to use pin 2). The LED also has two connections; as with the wire, one will go to ground, and the other to the pin to which you're writing (we're using pin 13). The short leg of the LED should connect to ground, and the longer end should connect to the digital pin, with the resistor attached to either—it does not matter which.



fritzing

Adding Code



The Temboo service has moved away from Arduino support. The service access in this older guide will no longer work.

You've already generated all of the code you need to check Twitter using your Yún, so you only need to make a few changes to your sketch to have the Yún determine whether a Tweet is new. First, let's add an **Output Filter** so that we only have to worry about the Tweet ID among all the different information that the Twitter API will return. The Output Filter will filter the API response in the cloud so that only the Tweet ID of your most recent Twitter mention is returned to the board. Before

```
// Run the Choreo; when results are available, print them to serial
MentionsChoreo.run();
```

add

```
MentionsChoreo.addOutputFilter("TweetID", "[/id", "Response");
```

and in void loop(), replace

```
while(MentionsChoreo.available()) {
  char c = MentionsChoreo.read();
  Serial.print(c);
```

with

```
while(MentionsChoreo.available()) {
  String TweetIDName = MentionsChoreo.readStringUntil('\x1F');
  TweetIDName.trim();
  String TweetIDNumber = MentionsChoreo.readStringUntil('\x1E');
  TweetIDNumber.trim();
```

Then, we'll add some code that will let your Yún check the newly retrieved Tweet ID against the previously returned ID to determine whether or not there's been a new mention. To do it, first initialize a new variable at the top of your code:

```
int LastID = 0;
```

Then, add the following lines of code after TweetIDNumber.trim();

```

if(TweetIDName == "TweetID"){
  int NewID = TweetIDNumber.toInt();

  if(LastID != NewID){
    digitalWrite(13, HIGH);
    delay(5000);
    digitalWrite(13, LOW);
    LastID = NewID;
  }
}
}

```

Next, we'll introduce the push button that will trigger your sketch. First, initialize the pin that you will be reading (recall that we chose to use pin 2) by adding a few lines of code to what you already have in void setup(). While we're at it, let's also initialize the output pin to which the LED is attached (we've selected pin 13 for this):

```

// Initialize pin
pinMode(2, INPUT);
pinMode(13, OUTPUT);

```

Then, add the following two lines of code to void loop() to set the Mentions Choreo to run only when the button is pressed:

```

int sensorValue = digitalRead(13);
if (sensorValue == LOW) {

```

You should nest the code that you already have in void loop() within this new conditional.

Finally, delete the lines

```

Serial.println("Waiting...");
delay(30000); // wait 30 seconds between Mentions calls

```

Since you're calling the Choreo only when you press the button, that line is no longer necessary.

With that, your sketch is once again ready to go! It should look something like this (note that the **Input** values in the code below have been replaced by placeholders):

```

#include <Bridge.h>
#include <Temboo.h>
#include "TembooAccount.h" // contains Temboo account information, as described below

int numRuns = 1; // Execution count, so this doesn't run forever
int maxRuns = 10; // Maximum number of times the Choreo should be executed
int LastID = 0;

void setup() {
  Serial.begin(9600);

  // For debugging, wait until the serial console is connected.
  delay(4000);
  while(!Serial);
  Bridge.begin();
}

```

```

// Initialize pin
pinMode(2, INPUT);
pinMode(13, OUTPUT);
}

void loop() {
  int sensorValue = analogRead(2);
  if (sensorValue == LOW) {
    if (numRuns <= maxRuns) {
      Serial.println("Running Mentions - Run #" + String(numRuns++));

      TembooChoreo MentionsChoreo;

      // Invoke the Temboo client
      MentionsChoreo.begin();

      // Set Temboo account credentials
      MentionsChoreo.setAccountName(TEMBOO_ACCOUNT);
      MentionsChoreo.setAppKeyName(TEMBOO_APP_KEY_NAME);
      MentionsChoreo.setAppKey(TEMBOO_APP_KEY);

      // Set Choreo inputs
      MentionsChoreo.addInput("Count", "1");
      MentionsChoreo.addInput("AccessToken", "PLACEHOLDER");
      MentionsChoreo.addInput("AccessTokenSecret", "PLACEHOLDER");
      MentionsChoreo.addInput("ConsumerSecret", "PLACEHOLDER");
      MentionsChoreo.addInput("ConsumerKey", "PLACEHOLDER");

      // Identify the Choreo to run
      MentionsChoreo.setChoreo("/Library/Twitter/Timelines/Mentions");

      MentionsChoreo.addOutputFilter("TweetID", "[/id]", "Response");

      // Run the Choreo; when results are available, print them to serial
      MentionsChoreo.run();

      while(MentionsChoreo.available()) {
        String TweetIDName = MentionsChoreo.readStringUntil('\x1F');
        TweetIDName.trim();
        String TweetIDNumber = MentionsChoreo.readStringUntil('\x1E');
        TweetIDNumber.trim();

        if(TweetIDName == "TweetID"){
          int NewID = TweetIDNumber.toInt();

          if(LastID != NewID){
            digitalWrite(13, HIGH);
            delay(5000);
            digitalWrite(13, LOW);
            LastID = NewID;
          }
        }
      }
      MentionsChoreo.close();
    }
  }
}
}

```


You can continue to make changes if you'd like, but no further alterations are required. Save it, upload it to your Arduino, open the serial monitor, and press the button to begin checking your Tweets.