Twitter til Arduino LED - Python

Med utgangspunktet i http://nickbester.com/send-commands-to-arduino-using-python-from-the-twitter/

Du trenger
Installert på maskinen:

- Python 2.7.13 - https://www.python.org/downloads/
- Arduino IDE - https://www.arduino.cc/en/Main/Software
- Twitter API installert (i terminalvindu MAC - **pip install TwitterAPI**)
  https://github.com/geduldig/TwitterAPI

Utstyr:

- Arduino
- LED på port A1

I tillegg må du ha en twitterkonto og en Twitter APP

- Slik gjør du det https://iktsenteret.app.box.com/notes/138219350712

Endringer i Python konden er merket med **fet** skrift.

- *arduinoPort* endres til den porten som din maskin bruker. På MAC finner du den ved å bruke terminalvindu og skrive ls /dev/cu.* (i mitt tilfellet er det /dev/cu.usbmodem40111). Du kan også finne det nederst i Arduino IDE programmet.

- *arduinoWaitTime* er satt til 0.001 for rask oppdatering, men er opprinnelig 1
- *stringToTrack* endres til ditt søk på twitter som får LED til å blinke - i koden #arduino men @potus eller @potus44 er stilige

Kodene finner dere også i boks -
https://iktsenteret.box.com/s/7n5swv9qsbdc7er45gyd1avypmx75qut

Python (2.7.13) koden

```
# Author: Nicholas Bester
# Title: Twitter Stream connection
# Description: Tracks a string using the Twitter API and sends a Serial command to an Arduino once a Tweet matching the tracked string is found
# Version: 1.1

# imports
import time
from time import sleep
from TwitterAPI import TwitterAPI
import struct
import os
from serial import Serial
import httplib
from httplib import IncompleteRead

# Pretty console colours
class bcolors:
    HEADER = '\033[95m'
    OKBLUE = '\033[94m'
    OKGREEN = '\033[92m'
    WARNING = '\033[93m'
    FAIL = '\033[91m'
    ENDC = '\033[0m'
    BOLD = '\033[1m'
    UNDERLINE = '\033[4m'

# Variables
availableArduino = True # Debugging without an Arduino
testSerial = False # Debugging without Twitter connection
arduinoPort = '/dev/cu.usbmodem40111' # USB port address for the Arduino
arduinoBaud = '9600' # Baud for serial communication
arduinoWaitTime = 0.001 # The length of time Python wait before attemping to issue commands to the Arduino - default 1
stringToTrack = "#arduino" # Change this to the search term you wish to track from Twitter

# Access requirements for Twitter API connection
access_token = 'xxxxx din egen kode xxxxxxxx'
```

```
access_token_secret = 'xxxxx din egen kode xxxxxxxx'
consumer_key = 'xxxxx din egen kode xxxxxxxx'
consumer_secret = 'xxxxx din egen kode xxxxxxxx'

# Clearing the screen for aesthetic display of console statements
os.system('cls' if os.name == 'nt' else 'clear')

print bcolors.WARNING + "Initialising Twitter Stream Application" + bcolors.ENDC

print bcolors.OKGREEN + "Initialisation OK!" + bcolors.ENDC

print bcolors.WARNING + 'Initialising Arduino Board through Serial' + bcolors.ENDC

# Arduino serial communication
if availableArduino:
    ser = Serial(arduinoPort, arduinoBaud, timeout=3)

# Gives the Arduino board time to initialise
sleep(arduinoWaitTime)

# Testing serial send to Arduino (ensure available Arduino is True)
if testSerial:
    print "On"
    ser.write(bytes(1))
    sleep(arduinoWaitTime)
    print "Off"
    ser.write(bytes(0))
    sleep(arduinoWaitTime)
else:
    print bcolors.OKGREEN + "Initialisation OK!" + bcolors.ENDC
    print bcolors.WARNING + 'Initialising Twitter Stream API Authorisation' + bcolors.ENDC
    try:
        # Setting up a connection to Twitter using the Twitter API
        api = TwitterAPI(consumer_key, consumer_secret, access_token, access_token_secret)

        print bcolors.OKGREEN + "Initialisation OK!" + bcolors.ENDC
        print bcolors.OKBLUE + 'Twitter Stream Request running' + bcolors.ENDC
```

```
        # A request object which opens a stream to Twitter tracking the hashtag in question
        r = api.request('statuses/filter', {'track':stringToTrack})

        # Checks if text within the stream item is populated and issues a command to the Arduino
        for item in r.get_iterator():
            if 'text' in item:
                print item['user']['screen_name'].encode('utf-8') + ' tweeted: ' + item['text'].encode('utf-8')# Print screen name and the tweet text

                # It is possible to check the tweets for further commands using regular expressions to send multiple commands to the Arduino

                if availableArduino:
                    print "Arduino turning on the LED"
                    ser.write(bytes(1)) # The command is a simple byte intepretation of the integer 1
                    sleep(arduinoWaitTime) # Wait before sending next command
                    ser.write(bytes(0))
                    sleep(arduinoWaitTime) # Wait before sending next command
    except IncompleteRead:
        # Oh well, reconnect and keep trucking
        print "IncompleteRead occurred"
    except KeyboardInterrupt:
        # Or however you want to exit this loop
        api.disconnect()
        exit()
```

Arduino kode med LED på A1

```
/*
 * Author: Nicholas Bester
 * Title: Twitter Mention Test
 * Version: 0.1
*/
```

```
// Debuging variables
int const DEBUG = 0; // Test LED without Serial feedback

// LED control
int ledPin = A1;

// Value sent from Python
int signalState;

void setup() {
  // Transistor pin connection on board
  pinMode(ledPin, OUTPUT);

  // Enabling communication
  Serial.begin(9600);

  // Test breadboard setup to LED
  if (DEBUG) {
    tweetReceived();
  }
}

void loop() {
  if (!DEBUG) {
    if (Serial.available()) {
      byte receivedValue = Serial.read() - '0';
      signalState = receivedValue;

      if (signalState == 1) {
        tweetReceived();
      }

      else if (signalState == 0) {
        ledToggle(false);
      }

      Serial.flush();
    }
```

```
  }
}

// Flash the light when tweet is received
void tweetReceived() {
  for (int i = 0; i < 10; i++) {
    ledToggle(true);
    delay(100);
    ledToggle(false);
    delay(100);
  }
}

// turn LED on and off
void ledToggle(boolean value) {
  if (value) {
    analogWrite(ledPin, 1023);
  } else {
    analogWrite(ledPin, 0);
  }
}
```