```
//includes
#include <SPI.h> // needed in Arduino 0019 or later
#include <Ethernet.h>
#include <Twitter.h>
#include <time.h>

// Ethernet Shield Settings
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };


// If you don't specify the IP address, DHCP is used(only in Arduino 1.0 or later).
byte ip[] = { 192, 168, 137, 34 };

// Your Token to Tweet (get it from http://arduino-tweet.appspot.com/)
Twitter twitter("INSERT TOKEN HERE");

int tweetcount = 0;



/*
 * PIR sensor tester
 */

int ledPin = 13;                // choose the pin for the LED
int inputPin = 12;               // choose the input pin (for PIR sensor)
int pirState = LOW;             // we start, assuming no motion detected
int val = 0;                    // variable for reading the pin status---
//LED Pin Variables
int ledPins[] = {2,3,4,5,6,7,8,9};

void setup() {
  pinMode(ledPin, OUTPUT);      // declare LED as output
  pinMode(inputPin, INPUT);     // declare sensor as input

  //Set each pin connected to an LED to output mode (pulling high (on) or low (off)
  for(int i = 0; i < 8; i++){         //this is a loop and will repeat eight times
      pinMode(ledPins[i],OUTPUT); //we use this to set each LED pin to output

  Serial.begin(9600);
}
}

void loop(){
  int trashcount = 0;

  val = digitalRead(inputPin);  // read input value
  if (val == HIGH) {            // check if the input is HIGH
    oneAfterAnotherLoop();
    trashcount++;
    tweetcount++;

    Serial.println(tweetcount);
    //if(tweetcount%2) {
      if(tweetcount) {
      TweetThis(tweetcount);
    }

    if (pirState == LOW) {
      // we have just turned on
      Serial.println("Motion detected!");
      // We only want to print on the output change, not state
      pirState = HIGH;
    }
  } else {
    digitalWrite(ledPin, LOW); // turn LED OFF
    if (pirState == HIGH){
      // we have just turned of
      Serial.println("Motion ended!");
      // We only want to print on the output change, not state
      pirState = LOW;
    }
  }
}


void TweetThis(int tweetcount) {
 delay(1000);

 // Message to post
char msg[] = "One less piece of trash! # of items kept off our streets: ";
char integer_string[32];



  sprintf(integer_string, "%d", tweetcount);

  strcat(msg, integer_string);

  Ethernet.begin(mac, ip);
  // or you can use DHCP for autoomatic IP address configuration.
  // Ethernet.begin(mac);
  Serial.begin(9600);

  Serial.println("connecting ...");
  if (twitter.post(msg)) {
    // Specify &Serial to output received response to Serial.
    // If no output is required, you can just omit the argument, e.g.
    // int status = twitter.wait();
    int status = twitter.wait(&Serial);
    if (status == 200) {
      Serial.println("OK.");
    } else {
      Serial.print("failed : code ");
      Serial.println(status);
    }
  } else {
    Serial.println("connection failed.");
  }
}

void oneAfterAnotherLoop(){
  int delayTime = 100; //the time (in milliseconds) to pause between LEDs
                       //make smaller for quicker switching and larger for slower

//Turn Each LED on one after another
  for(int i = 0; i <= 7; i++){
    digitalWrite(ledPins[i], HIGH);  //Turns on LED #i each time this runs i
    delay(delayTime);                //gets one added to it so this will repeat
  }                                  //8 times the first time i will = 0 the final
                                     //time i will equal 7;

//Turn Each LED off one after another
  for(int i = 7; i >= 0; i--){  //same as above but rather than starting at 0 and counting up
                                //we start at seven and count down
    digitalWrite(ledPins[i], LOW);  //Turns off LED #i each time this runs i
    delay(delayTime);                //gets one subtracted from it so this will repeat
  }                                  //8 times the first time i will = 7 the final
                                     //time it will equal 0


}
```