```
/*
 * Rapid Systems Engineering
 * rapidsystemsengineering.com
 * Sketch uses Twilio and Thingspeak APIs to get be notified by text message when
motion
 * is detected.
 */

#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>

#define USE_SERIAL Serial

ESP8266WiFiMulti WiFiMulti;
int led = 13;                // the pin that the LED is atteched to
int sensor = 2;              // the pin that the sensor is atteched to
int state = LOW;             // by default, no motion detected
int val = 0;                 // variable to store the sensor status (value)

void setup() {
  pinMode(led, OUTPUT);      // initalize LED as an output
  pinMode(sensor, INPUT);    // initialize sensor as an input
  USE_SERIAL.begin(115200);

  USE_SERIAL.println();
  USE_SERIAL.println();
  USE_SERIAL.println();

  for (uint8_t t = 4; t > 0; t--) {
    USE_SERIAL.printf("[SETUP] WAIT %d...\n", t);
    USE_SERIAL.flush();
    delay(1000);
  }

  WiFiMulti.addAP("WIFI_SSID", "WIFI_PASSWORD");
}

void loop() {
  val = digitalRead(sensor);   // read sensor value
  if (val == HIGH) {           // check if the sensor is HIGH
    digitalWrite(led, HIGH);   // turn LED ON
    delay(100);                // delay 100 milliseconds

    if (state == LOW) {
      Serial.println("Motion detected!");
      HTTPClient http;

      USE_SERIAL.print("[HTTP] begin...\n");
      // configure traged server and url
```

```
http.begin("https://api.thingspeak.com/apps/thinghttp/send_request?api_key=[apikey]");

      USE_SERIAL.print("[HTTP] GET...\n");
      // start connection and send HTTP header
      int httpCode = http.GET();

      // httpCode will be negative on error
      if (httpCode > 0) {
        // HTTP header has been send and Server response header has been handled
        USE_SERIAL.printf("[HTTP] GET... code: %d\n", httpCode);

        // file found at server
        if (httpCode == HTTP_CODE_OK) {
          String payload = http.getString();
          USE_SERIAL.println(payload);
        }
      }
      state = HIGH;       // update variable state to HIGH
    }
  }
  else {
    digitalWrite(led, LOW); // turn LED OFF
    delay(200);             // delay 200 milliseconds

    if (state == HIGH) {
      Serial.println("Motion stopped!");
      state = LOW;       // update variable state to LOW
    }
  }
}
```