Smart Home

```
// Smart Home Code for the ESP

/* Includes Section*/
#include <Wire.h>
#include <SoftwareSerial.h>
#include <ThingSpeak.h>
#include <ESP8266WiFi.h>

SoftwareSerial s(3,1);

// Set up the CLIENT connection
WiFiClient  client;
unsigned long counterChannelNumber = 2536970;                // Channel ID
const char * myCounterReadAPIKey = "0AN0ZZ63TWGFSD8F";       // Read API Key
const int FieldNumber1 = 1;                                  // The field you wish to read
const int FieldNumber2 = 2;                                  // The field you wish to read

// Timer
#define INTERVAL 1000
unsigned long time_for_action;

// Set up the pin variables
#define FLAME_SENSOR      5
#define EXHAUST_FAN       4
#define BATH_TUB          12
#define FIRE_EXTINGUISHER 14

bool FLAME_SENSOR_STATUS = 0;

bool ALARM_OFF_Status = 0;

void setup() {
  // Connecting to Wi-Fi
  s.begin(115200);
  s.println();

  WiFi.begin("Galal", "600070000");                 // write wifi name & password

  s.print("Connecting");
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    s.print(".");
  }
  s.println();
  s.print("Connected, IP address: ");
  s.println(WiFi.localIP());
  ThingSpeak.begin(client);

  pinMode(FLAME_SENSOR, INPUT);
  pinMode(EXHAUST_FAN, OUTPUT);
  pinMode(BATH_TUB, OUTPUT);
  pinMode(FIRE_EXTINGUISHER, OUTPUT);

  digitalWrite(EXHAUST_FAN, HIGH);
  digitalWrite(BATH_TUB, HIGH);
  digitalWrite(FIRE_EXTINGUISHER, HIGH);
}

void loop() {
  FLAME_SENSOR_STATUS = digitalRead(FLAME_SENSOR);

  int A;
  if(millis() > time_for_action)
  {
    time_for_action = millis() + (unsigned long) INTERVAL;
    A = ThingSpeak.readLongField(counterChannelNumber, FieldNumber1, myCounterReadAPIKey);
    s.println(A);
  }

  if(A == 5)
  {
    digitalWrite(BATH_TUB, LOW);
  }
  else if(A == 6)
  {
    digitalWrite(BATH_TUB, HIGH);
  }

  if(FLAME_SENSOR_STATUS == LOW)
  {
    while(!FLAME_SENSOR_STATUS)
    {
      FLAME_SENSOR_STATUS = digitalRead(FLAME_SENSOR);
      s.println(2); // FIRE ALERT
      digitalWrite(EXHAUST_FAN, LOW);
      digitalWrite(FIRE_EXTINGUISHER, LOW);
      delay(2000);
    }
  }
  else
  {
    digitalWrite(EXHAUST_FAN, HIGH);
    digitalWrite(FIRE_EXTINGUISHER, HIGH);
  }

}
```

PDF document made with CodePrint.org