```
// ArduinoIoTCloud - Version: Latest
#include <ArduinoIoTCloud.h>

#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include "DHT.h"
#include "WiFi.h"
#include "thingProperties.h"

#define READ_PINQ 22
#define DHTPIN 23
#define DHTTYPE DHT22

const char* ssid = "wifi name";
const char* password = "PASSWORD";

LiquidCrystal_I2C lcd(0x27, 16, 2);
int mesure;
DHT dht(DHTPIN, DHTTYPE);

float getTemp(String unit) {
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);
  float hif = dht.computeHeatIndex(f, dht.readHumidity());
  float hic = dht.computeHeatIndex(t, dht.readHumidity(), false);
  float k = t + 273.15;
  float h = dht.readHumidity();
  if (unit == "c") {
    return t;
```

```
  } else if (unit == "f") {
    return f;
  } else if (unit == "hic") {
    return hic;
  } else if (unit == "hif") {
    return hif;
  } else if (unit == "k") {
    return k;
  } else if (unit == "h") {
    return h;
  } else {
    return NAN;
  }
}


void setup() {
  Serial.begin(9600);
  initProperties();
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
  // Séparation DHT
  Serial.println("DHTxx Robojax test!");
  dht.begin();

  // Séparation qualité d'air
  Serial.println("Welcome to home AQI station!");
```

```
  // Séparation WiFi
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);
  Serial.println("Setup done");
}

void loop() {

  // Séparation WiFi
  Serial.println("scan start");
  int n = WiFi.scanNetworks();
  Serial.println("scan done");
  if (n == 0) {
    Serial.println("no networks found");
  } else {
    Serial.print(n);
    Serial.println(" networks found");
    for (int i = 0; i < n; ++i) {
      Serial.print(i + 1);
      Serial.print(": ");
      Serial.print(WiFi.SSID(i));
      Serial.print(" (");
      Serial.print(WiFi.RSSI(i));
      Serial.print(")");
      Serial.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN) ? " " : "*");
      delay(10);
    }
  }
```

```
Serial.println("");
delay(5000);

// Séparation DHT
Serial.print("Temperature: ");
Serial.print(getTemp("c"));
Serial.print(" *C ");
Serial.print(getTemp("f"));
Serial.println (" *F");
Serial.println("-----------------");
Serial.print("Heat index: ");
Serial.print(getTemp("hic"));
Serial.print(" *C ");
Serial.print(getTemp("hif"));
Serial.println(" *F");
Serial.print(getTemp("k"));
Serial.println(" *K");
Serial.println("-----------------");
Serial.print("Humidity: ");
Serial.print(getTemp("h"));
Serial.println(" % ");
Serial.println("============================");

// Séparation qualité d'air
int aqi = analogRead(READ_PINQ);
delay(5000);
Serial.print("Air Quality Index: ");
Serial.print(aqi, DEC);
Serial.print(" PPM");
```

```
  if (aqi < 500) {
    Serial.println(", Fresh Air");
  } else if (aqi > 500 && aqi <= 1000) {
    Serial.println(", Poor Air");
  } else if (aqi > 1000) {
    Serial.println(", Very Poor");
  }
}
```