```cpp
#include "Ubidots.h"
#include"Adafruit_I2CDevice.h"
#include<ESP8266WiFi.h>
#include "DHT.h"
#include "MQ135.h"
#include <Wire.h>
#include"Adafruit_Sensor.h"
#include <SPI.h>
#include<Adafruit_BMP280.h>

// ---------- CONEXIÓN WIFI
#define WIFISSID "LAB_MEL" // Put here your Wi-Fi SSID
#define PASSWORD "Mel_1981" // Put here your Wi-Fi password
//#define WIFISSID "Felipe" // Put here your Wi-Fi SSID
//#define PASSWORD "0804341584@" // Put here your Wi-Fi password

// ---------: DHT11 - sensor de temperatura y humedad :---------

//const byte  Obj_DHT11 = D1;
// Definimos el pin digital donde se conecta el sensor
#define DHTPIN 0
// Dependiendo del tipo de sensor
#define DHTTYPE DHT11

// ---------: BME280 :---------
//#define BMP280_I2C_ADDRESS  0x76 // initialize Adafruit BMP280 library

//Adafruit_BMP280 bmp; // I2C

// ---------: GP2Y1014 - sensor de humo :---------

int dustPin = A0; // dust sensor - Wemos A0 pin
int ledPin =14;

float voltsMeasured = 0;
float calcVoltage = 0;
float dustDensity = 0;
float pm05=0;

// Funciones de ayuda para imprimir un valor de datos en el monitor serie.
void printFValue(String text, float value, String units, bool isLast = false) {
 Serial.print(text);
 Serial.print("=");
 Serial.print(value);
 Serial.print(units);
  if (!isLast) {
    Serial.print(", ");
  }
}
```

```cpp
// Inicializamos el sensor DHT11
DHT dht(DHTPIN, DHTTYPE);


// ---------- UBIDOST

#define DEVICE  "esp8266_principal" // Put here your Ubidots device label

#define VARIABLE1  "C2-Indice-Calor"  // Put here your Ubidots variable label
#define VARIABLE2  "C2-Humedad"  // Put here your Ubidots variable label
#define VARIABLE3  "C2-Temperatura"  // Put here your Ubidots variable label

#define VARIABLE4  "C2-Altura"  // Put here your Ubidots variable label
#define VARIABLE5  "C2-Presion"  // Put here your Ubidots variable label

#define VARIABLE6  "C2-Densidad-Polvo"  // Put here your Ubidots variable label
#define VARIABLE7  "C2-PM0.5"  // Put here your Ubidots variable label

#define TOKEN  "BBFF-RBqYtahF0W7DKBXbTIwVHlDhxbXhh1"  // Put here your Ubidots
TOKEN

Ubidots ubidost(TOKEN);




///--------------------- inicio HW611 EP 280   -----------

#define BMP_SCK  (13)
#define BMP_MISO (12)
#define BMP_MOSI (11)
#define BMP_CS   (10)


// define device I2C address: 0x76 or 0x77 (0x77 is library default address)
#define BMP280_I2C_ADDRESS  0x76
// initialize Adafruit BMP280 library

Adafruit_BMP280 bmp; // I2C
//Adafruit_BMP280 bmp(BMP_CS); // hardware SPI
//Adafruit_BMP280 bmp(BMP_CS, BMP_MOSI, BMP_MISO,  BMP_SCK);


///--------------------- fin HW611 EP 280   -----------
```

```cpp
void setup() {

  // ---------: DHT11 - sensor de temperatura y humedad :---------

  // Comenzamos el sensor DHT
  dht.begin();

  // ---------: GP2Y1014 - sensor de humo :---------

  // Establecer el pin del LED para la salida.
  pinMode(ledPin,OUTPUT);

  // Inicie el puerto serial de hardware para el monitor serial.
  Serial.begin(115200);

  // Espera dos segundos para el inicio.
  delay(2000);
  Serial.println("Inicializando");
  Serial.println("================");

  ///-------------------- inicio HW611 EP 280  -----------

   while ( !Serial ) delay(100);   // wait for native usb
   Serial.println(F("BMP280 test"));
   unsigned status;
  //status = bmp.begin(BMP280_ADDRESS_ALT, BMP280_CHIPID);
  status = bmp.begin(0x76); //simply replaced status = bmp.begin(); by status =
bmp.begin(0x76);
  if (!status) {
    Serial.println(F("Could not find a valid BMP280 sensor, check wiring or "
                    "try a different address!"));
   Serial.print("SensorID was: 0x"); Serial.println(bmp.sensorID(),16);
    Serial.print("        ID of 0xFF probably means a bad address, a BMP 180 or
BMP 085\n");
    Serial.print("   ID of 0x56-0x58 represents a BMP 280,\n");
    Serial.print("        ID of 0x60 represents a BME 280.\n");
    Serial.print("        ID of 0x61 represents a BME 680.\n");
    while (1) delay(10);
  }

  /* Default settings from datasheet. */
  bmp.setSampling(Adafruit_BMP280::MODE_NORMAL,     /* Operating Mode. */
                  Adafruit_BMP280::SAMPLING_X2,     /* Temp. oversampling */
                  Adafruit_BMP280::SAMPLING_X16,    /* Pressure oversampling */
                  Adafruit_BMP280::FILTER_X16,      /* Filtering. */
                  Adafruit_BMP280::STANDBY_MS_500); /* Standby time. */
///----------------------fin HW611 EP 280   -----------
```

```arduino
  // ---------: BME280 :---------

 // if (!bme.begin(0x76)) {
 //   Serial.println("Could not find a valid BME280 sensor, check wiring!");
 //   while (1);
 // }

  // Ubidost
ubidost.wifiConnect(WIFISSID, PASSWORD);




}

void loop() {

  // ---------: MQ135 - sensor de gas : --------------
   MQ135 gasSensor = MQ135(A0);
   float air_quality = gasSensor.getPPM();
    Serial.print("Air Quality: ");
    Serial.print(air_quality);
    Serial.println("  PPM");
    Serial.println();

  // ---------: GP2Y1014 - sensor de humo :---------

 digitalWrite(ledPin,LOW); // power on the LED
 delayMicroseconds(280);

 voltsMeasured = analogRead(dustPin); // read the dust value

 delayMicroseconds(40);
 digitalWrite(ledPin,HIGH); // turn the LED off
 delayMicroseconds(9680);

 //measure your 5v and change below
 calcVoltage = voltsMeasured * (3.3 / 1024.0);
 dustDensity = 0.17 * calcVoltage - 0.1;
 //Ecuacion linear de PM 2.5
 pm05=(calcVoltage-0.0356)*120000;

  // ---------: DHT11 - sensor de temperatura y humedad :---------

  // Leemos la humedad relativa
  float h = dht.readHumidity();
  // Leemos la temperatura en grados centígrados (por defecto)
  float t = dht.readTemperature();
```

```arduino
  // Comprobamos si ha habido algún error en la lectura
  if (isnan(h) || isnan(t)) {
    Serial.println("Error obteniendo los datos del sensor DHT11");
    return;
  }

  // Calcular el índice de calor en grados centígrados
  float hic = dht.computeHeatIndex(t, h, false);

  // ---------: BME280 :---------
 //float presion = bme.readPressure() / 100.0F;
 //float altitud = bme.readAltitude(SEALEVELPRESSURE_HPA);

///----------------------inicio HW611 EP 280   -----------

  float presion = bmp.readPressure();
  float altitud = bmp.readAltitude(1013.25);

    Serial.print(F("Temperature = "));
    Serial.print(bmp.readTemperature());
    Serial.println(" *C");

    Serial.print(F("Pressure = "));
    Serial.print(bmp.readPressure());
    Serial.println(" Pa");

    Serial.print(F("Approx altitude = "));
    Serial.print(bmp.readAltitude(1013.25)); /* Adjusted to local forecast! */
    Serial.println(" m");

    Serial.println();
    delay(2000);
///----------------------fin HW611 EP 280   -----------


  //Imprimir los valores
  printFValue("Índice de calor", hic, "*C", true);
  printFValue("Humedad", h, "%");
  printFValue("Temperatura", t, "*C");
 printFValue("Gas",air_quality,"CO2");
  printFValue("Altitud", altitud, "m");
  printFValue("Presion", presion, "hPa");
  //printFValue("Altitud", altitud, "m");
  //printFValue("Presion", presion, "hPa");

  printFValue("Densidad de Polvo", dustDensity, "ug/m3");
 printFValue("PM0.5",pm05,"pie3");
```

```
  Serial.println("");

  ubidost.add(VARIABLE1, hic);
  ubidost.add(VARIABLE2, h);
  ubidost.add(VARIABLE3, t);
  ubidost.add(VARIABLE4, altitud);
  ubidost.add(VARIABLE5, presion);
  ubidost.add(VARIABLE6, dustDensity);
  ubidost.add(VARIABLE7, pm05);
  //ubidost.add(VARIABLE8, air_quality);
  bool bufferSent = false;
  bufferSent = ubidost.send(DEVICE);

  if (bufferSent) {
    Serial.println("Values device");
  }
  delay(5000);
}
```