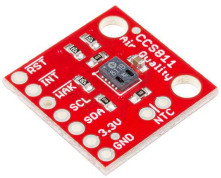
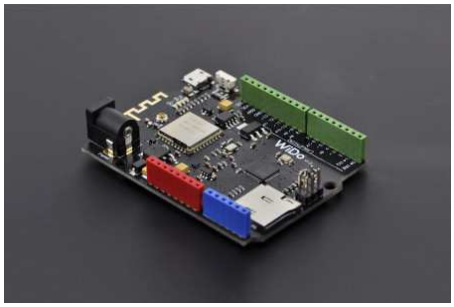
IoT Air Quality Monitor

Poluarea aerului este o problemă extrem de actuală mai ales în mediul urban. Măsurarea calității aerului ne poate indica la ce riscuri de sănătate ne expunem. Există o serie de măsurători oficiale dar ele sunt afectate în mare măsură de interesele sau neglijența celor care administrează rețelele de monitorizare. Se poate vedea o hartă în timp real a calității aerului în România pe site-ul aciqn.org (1), din păcate în majoritatea orașelor mari din România senzorii sunt nefuncționali…

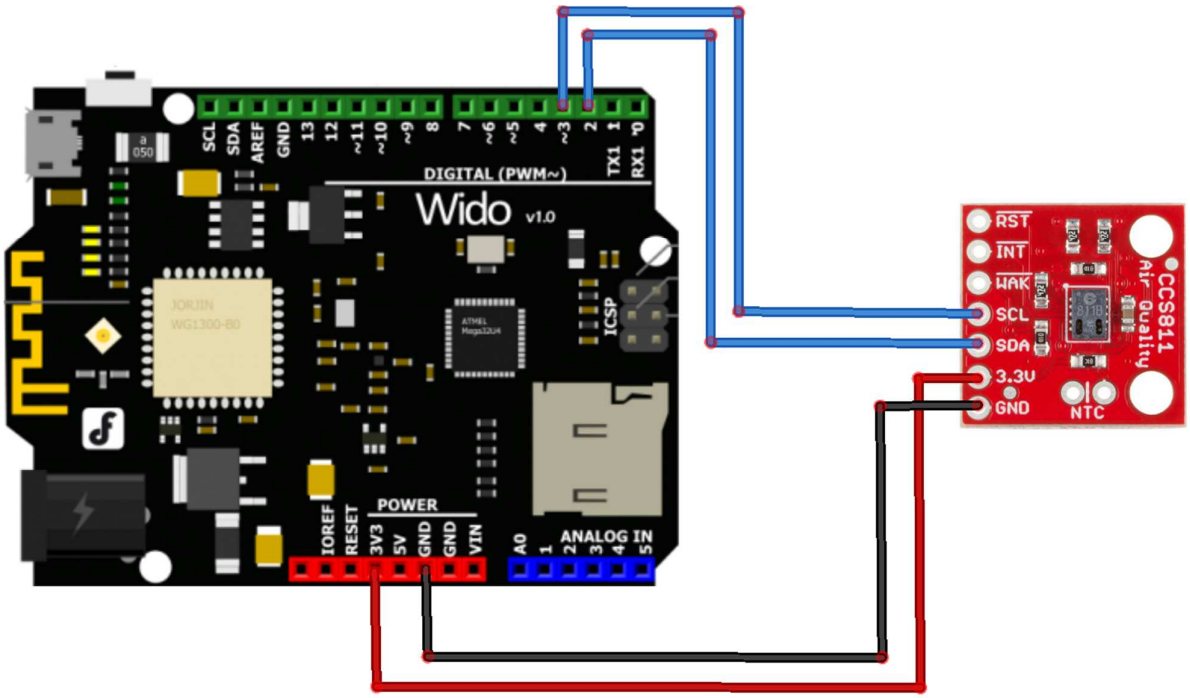


Există mai mulți indicatori ce influențează calitatea aerului. În cadrul lecției de față vom utiliza un senzor CCS811 (2) ce este capabil să măsoare concentrațiile de eCO2 (dioxid de carbon echivalent) și tVOC (total compuși volatili organici) din aer în spații închise (în interior). Pentru mai multe detalii despre semnificația celor doi indicatori se poate parcurge materialul „Air Quality Measurements with the CCS811” (3). Punerea în funcțiune a senzorului necesită o perioadă de 48 de ore de funcționare în gol (la prima utilizare) și ulterior un timp de 20 de minute până la stabilizarea măsurătorilor (la pornirile ulterioare ale sistemului). Pentru modul de funcționare a senzorului CCS811 se poate parcurge și materialul „CCS811 Air Quality Breakout Hookup Guide” (4).

Valorile furnizate de senzor vor fi înregistrate în cloud utilizând serviciul Robofun IoT (5). Acest serviciu este gratuit dar necesită înregistrare, mai multe detalii despre modul de utilizare a serviciului se pot găsi pe site-ul de documentație oficial (6). Pentru a trimite datele prin Internet vom utiliza o placă de dezvoltare WiDo-WIFI IoT (7) ce combină un microcontroler ATmega32U4 (la fel ca și placa de dezvoltare Arduino Leonardo) și un controler WiFi WG1300 (bazat pe circuitul integrat CC3000). Această combinație oferă avantajul programării foarte simple, la fel ca orice placă din familia Arduino, dar și posibilitatea de comunicație în rețea utilizând o conexiunea WiFi.



Schema de interconectare între senzor și placa de dezvoltare este următoarea (senzorul se va alimenta la 3.3V iar liniile de comunicație I2C sunt SCL – pin D3, SDA – pin D2):



Programul a fost dezvoltat și testat utilizând Arduino IDE 1.8.3 și bibliotecile SparkFun CCS811 1.0.0 și o versiune modificată a bibliotecii Adafruit CC3000 (8) – special pentru placa de dezvoltare WiDo-WIFI. Placa se va programa în Arduino IDE ca o placă Arduino Leonardo obișnuită.

```
#include <Adafruit_CC3000.h>
#include <ccspi.h>
#include <SPI.h>
#include <string.h>
#include "utility/debug.h"

#define WiDo_IRQ   7
#define WiDo_VBAT  5
#define WiDo_CS    10

Adafruit_CC3000 WiDo = Adafruit_CC3000(WiDo_CS, WiDo_IRQ,
```

```
                    WiDo_VBAT,SPI_CLOCK_DIVIDER);
```

În cadrul programului trebuie completate datele de conectare la rețeaua WiFi.

```
#define WLAN_SSID       "..."
#define WLAN_PASS       "..."
#define WLAN_SECURITY   WLAN_SEC_WPA2

#define TIMEOUT_MS  2000
#define WEBSITE  "iot.robofun.ro"

#include "SparkFunCCS811.h"
#define CCS811_ADDR 0x5B
CCS811 mySensor(CCS811_ADDR);
```

Comentarea directivei următoare va conduce la suprimarea mesajelor în consola serială.

```
#define debug
```

În cadrul secțiunii *setup()* se vor inițializa componenta WiFi, conexiunea WiFi și senzorul de calitate a aerului. Procedura *displayConnectionDetails()* permite afișarea în consola serială a datelor de configurare în rețea.

```
void setup(){
  #ifdef debug
    SerialUSB.begin(115200);
    while(!SerialUSB) { ; }
    SerialUSB.println(F("Robofun IoT Air Quality
                    Monitor\n"));
    SerialUSB.print(F("Free RAM: "));
    Serial.println(getFreeRam(), DEC);
    SerialUSB.println(F("\nInitialising the CC3000 ..."));
```

```
#endif

if (!WiDo.begin())
{
  #ifdef debug
    SerialUSB.println(F("Unable to initialise the
          CC3000! Check your wiring?"));
  #endif
  while(1);
}

CCS811Core::status returnCode = mySensor.begin();
if (returnCode != CCS811Core::SENSOR_SUCCESS)
{
  #ifdef debug
    SerialUSB.println(F("CCS811.begin() returned with an
            error."));
  #endif
  while (1);
}

if (!WiDo.connectToAP(WLAN_SSID,WLAN_PASS,
                  WLAN_SECURITY)) {
  #ifdef debug
    SerialUSB.println(F("Failed to connect to AP!"));
  #endif
  while(1);
}
#ifdef debug
  SerialUSB.println(F("Connected to AP!"));
  SerialUSB.println(F("Request DHCP"));
#endif
```

```
  while (!WiDo.checkDHCP())
  {
    delay(100);
  }
  #ifdef debug
    while (! displayConnectionDetails()) {
      delay(1000);
    }
  #endif
}

bool displayConnectionDetails(void)
{
  uint32_t ipAddress, netmask, gateway, dhcpserv, dnsserv;

  if(!WiDo.getIPAddress(&ipAddress,  &netmask,  &gateway,
&dhcpserv, &dnsserv))
  {
    SerialUSB.println(F("Unable  to  retrieve  the  IP
Address!\r\n"));
    return false;
  }
  else
  {
    SerialUSB.print(F("\nIP Addr: "));
    WiDo.printIPdotsRev(ipAddress);
    SerialUSB.print(F("\nNetmask: "));
    WiDo.printIPdotsRev(netmask);
    SerialUSB.print(F("\nGateway: "));
    WiDo.printIPdotsRev(gateway);
    SerialUSB.print(F("\nDHCPsrv: "));
    WiDo.printIPdotsRev(dhcpserv);
    SerialUSB.print(F("\nDNSserv: "));
```

```
    WiDo.printIPdotsRev(dnsserv);
    SerialUSB.println();
    return true;
  }
}
```

Măsurarea celor doi parametrii de calitate a aerului este afectată de temperatura și umiditatea ambientală. Biblioteca senzorului este capabilă să compenseze acest aspect prin preluarea din program a valorilor necesare. Pentru măsurători exacte este indicată utilizarea unui senzor de temperatură și umiditate în sistem care să furnizeze aceste valori în mod dinamic dar în cadrul sistemului nostru vom utiliza două valori constante.

```
#define temperatureVariable 28.0
#define humidityVariable 27.0
```

În cadrul secțiunii *loop()* se vor prelua datele de la senzor și se vor transmite către procedua *postIoT()* care se va ocupa cu înregistrarea în cloud. Este necesară particularizarea în program a valorilor *TOKEN1* și *TOKEN2*, valori obținute în urma definirii a celor doi senzori în serviciul Robofun IoT. Postarea se va realiza la un interval de 10 minute (600 secunde = 600000 milisecunde).

```
void loop(){
   if (mySensor.dataAvailable()) {
      mySensor.setEnvironmentalData(humidityVariable,
                temperatureVariable);
      delay(10);
      mySensor.readAlgorithmResults();
      int tempCO2 = mySensor.getCO2();
      int tempVOC = mySensor.getTVOC();
      #ifdef debug
        SerialUSB.print(F("CO2["));
        SerialUSB.print(tempCO2);
        SerialUSB.print(F("] tVOC["));
        SerialUSB.print(tempVOC);
```

```
        SerialUSB.println(F("]"));
      #endif
      String temp = "/api/v1/senzor/TOKEN1/input?value=" +
                    String(tempCO2);
      char clientString[50];
      temp.toCharArray(clientString,temp.length()+1);
      postIoT(clientString);
      delay(5000);
      temp = "/api/v1/senzor/TOKEN2/input?value=" +
                    String(tempVOC);
      temp.toCharArray(clientString,temp.length()+1);
      postIoT(clientString);
   }
   delay(600000);
}
```

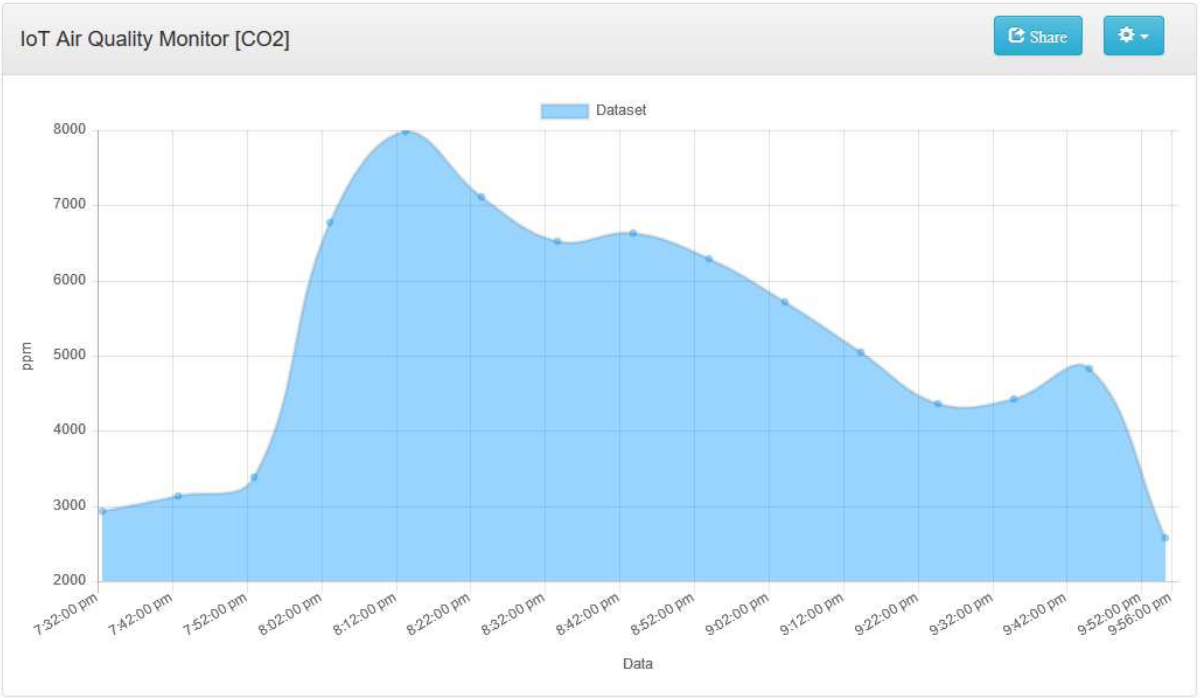
Procedura *postIoT()* preia datele ce urmează a fi trimise către serviciul cloud Robofun IoT și realizează comunicația HTTP (HTTP GET) aferentă.

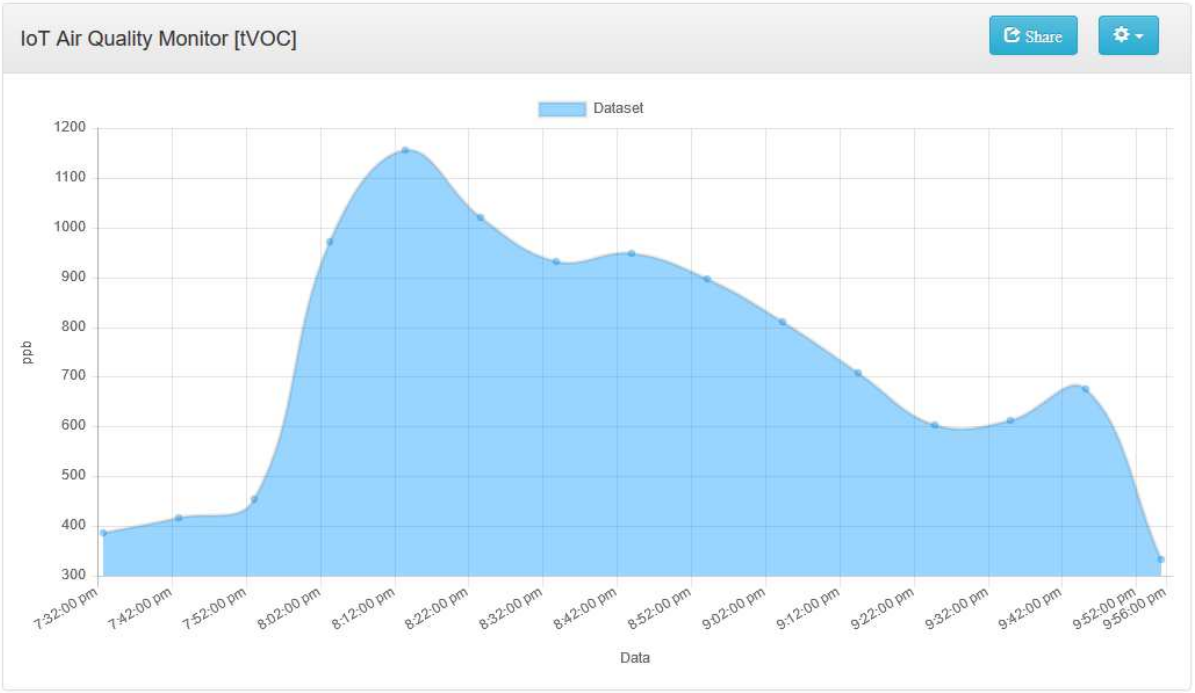
```
void postIoT(char* URLClient) {
    static Adafruit_CC3000_Client IoTclient;
    uint32_t ip = 0;
    while  (ip  ==  0)  {
      if  (!WiDo.getHostByName(WEBSITE, &ip))  {
        #ifdef debug
          SerialUSB.println(F("Couldn't resolve!"));
        #endif
        delay(500);
      }
      delay(500);
    }
    #ifdef debug
```

```
  SerialUSB.print(WEBSITE); Serial.print(F(" -> "));
  WiDo.printIPdotsRev(ip);
  SerialUSB.println();
#endif
IoTclient = WiDo.connectTCP(ip,80);
#ifdef debug
  SerialUSB.print(F("Connecting to IoT Server"));
  while(!IoTclient.connected()) SerialUSB.print(".");
  SerialUSB.println();
  SerialUSB.println(URLClient);
#else
  while(!IoTclient.connected()) delay(10);
#endif
IoTclient.fastrprint(F("GET "));
IoTclient.fastrprint(URLClient);
IoTclient.fastrprintln(F(" HTTP/1.1"));
IoTclient.fastrprintln(F("Host: iot.robofun.ro"));
IoTclient.fastrprintln(F("Connection: close"));
IoTclient.fastrprint(F("\r\n"));
IoTclient.println();
#ifdef debug
    SerialUSB.println(F("Upload data to the IoT
                Server"));
#endif
unsigned long lastRead = millis();
while (IoTclient.connected() && (millis() - lastRead <
                TIMEOUT_MS)) {
      while (IoTclient.available()) {
        char c = IoTclient.read();
        #ifdef debug
          SerialUSB.print(c);
        #endif
```

```
            lastRead = millis();
          }
    }
    #ifdef debug
      SerialUSB.println();
      SerialUSB.flush();
    #endif
    IoTclient.close();
}
```

După configurarea serviciului cloud și încărcarea programului putem deja observa online datele înregistrate (eCO2 – valori între 400 și 8,192 ppm – părți per milion; tVOC – valori într 0 și 1,187 ppb – părți per miliard):




IoT Air Quality Monitor [tVOC]
Share
Dataset
1200
1100
1000
900
800
700
600
500
400
300
ppb
7:32:00 pm
7:42:00 pm
7:52:00 pm
8:02:00 pm
8:12:00 pm
8:22:00 pm
8:32:00 pm
8:42:00 pm
8:52:00 pm
9:02:00 pm
9:12:00 pm
9:22:00 pm
9:32:00 pm
9:42:00 pm
9:52:00 pm
9:56:00 pm
Data

Referințe on-line

(1) Air Pollution in Romania: Real-time Air Quality Index Visual Map
http://aqicn.org/map/romania/

(2) Senzor pentru masurarea calitatii aerului - CCS811 (eCO2 TVOC)
https://www.robofun.ro/senzori/biometric/senzor-calitatea-aerului-eco2-tvoc-ccs811?utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(3) Hardware Hump Day: Air Quality Measurements with the CCS811
https://www.sparkfun.com/news/2369

(4) CCS811 Air Quality Breakout Hookup Guide
https://learn.sparkfun.com/tutorials/ccs811-air-quality-breakout-hookup-guide

(5) Robofun IoT
http://iot.robofun.ro/

(6) Robofun IoT API
http://iot.robofun.ro/doc

(7) WiDo - Arduino Iot
https://www.robofun.ro/wireless/wireless-wifi/wido-arduino-iot-internet-of-thing-board?utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(8) DFRobot_WiDo_Arduino_IOT_Board_DFR0321
https://github.com/Arduinolibrary/DFRobot_WiDo_Arduino_IOT_Board_DFR0321/