

Textul și imaginile din acest document sunt licențiate

Attribution-NonCommercial-NoDerivs

CC BY-NC-ND



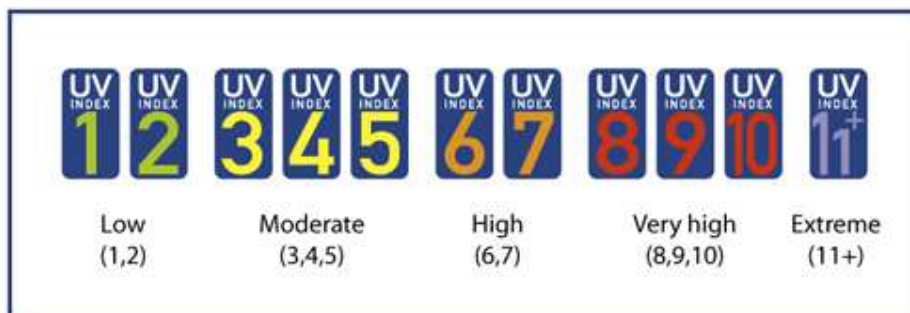
Codul sursă din acest document este licențiat

Public-Domain

Ești liber să distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, tipărire, sau orice alt mijloc), atât timp cât nu aduci nici un fel de modificări acestuia. Codul sursă din acest document poate fi utilizat în orice fel de scop, de natură comercială sau nu, fără nici un fel de limitări dar autorii nu își asumă nici o răspundere pentru pagubele pricinuite de implementările realizate de utilizatori. Schemele și codul sursă au un rol educativ și nu sunt gândite pentru a fi utilizate în mediu de producție (industrial, casnic sau comercial).

# IoT UV Monitor

Razele ultraviolete din lumina solară reprezintă un factor de risc pentru om. Monitorizarea nivelului de radiații UV (UV index) se face pe o scară de la 1 la 10 (11 fiind deja un nivel extrem). Realizarea unui sistem electronic ce măsoară cu precizie indexul UV poate oferi informații utile pentru modul în care putem să ne protejăm (îmbrăcăminte, ochelari de protecție, creme de protecție etc.) în activitățile de zi cu zi.



În cadrul lecției de față vom realiza un sistem care să măsoare indexul UV și să înregistreze datele măsurate pe serviciul cloud Robofun IoT ([1](https://www.robofun.ro/)). Pentru utilizarea acestui serviciu este necesară înregistrarea gratuită.

The screenshot shows a web browser window with the URL [iot.robofun.ro/signup](https://iot.robofun.ro/signup). The page has a navigation bar with links for Home, Documentatie, and Contact, along with Login and Cont nou buttons. The main content area is titled 'Please Sign Up' and contains two input fields: 'Email' (with placeholder 'Email address') and 'Password' (with placeholder 'Password'). A 'Sign up' button is located below the password field. At the bottom, there is a link for 'Already have an account? Sign In'.

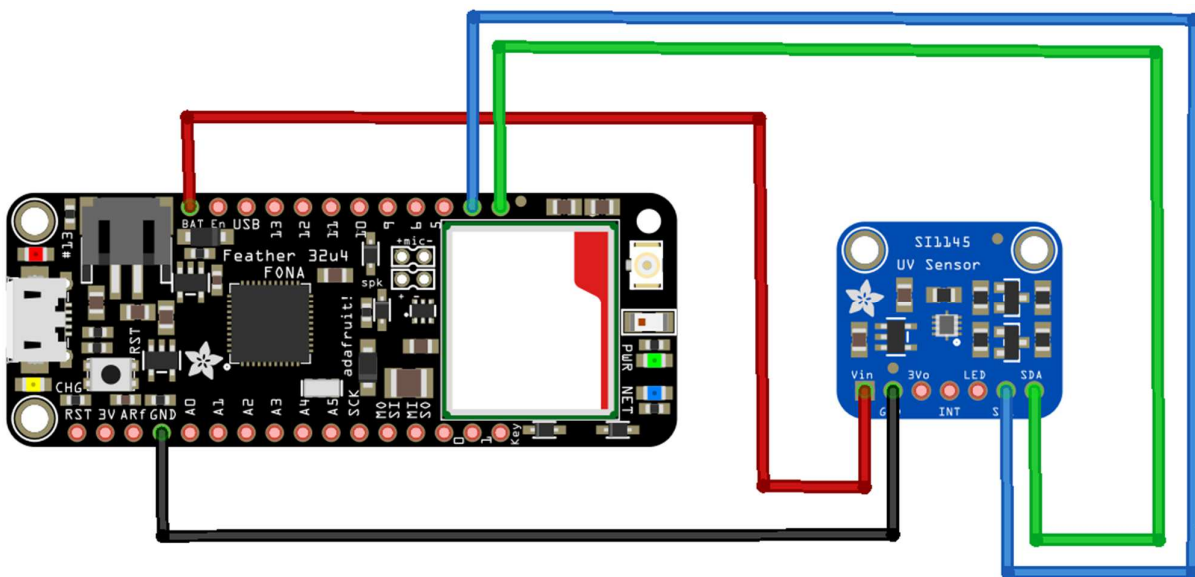
După înregistrare și conectare este necesară definirea unui senzor (*Adauga senzor*) pentru a putea înregistra valorile măsurate (UV index).



După definirea senzorului este necesar să copiem cheia de autentificare (*Token*) pentru a o utiliza în program.



Pentru implementarea hardware vom utiliza o placă de dezvoltare Adafruit Feather 32U4 FONA (2) și senzorul digital I2C Si1145 (3). Schema de interconectare între cele două componente este:



Senzorul se va alimenta de la pinul BAT al plăcii de dezvoltare. Pinul SDA al senzorului se va conecta la pinul 2 al plăcii de dezvoltare iar pinul SCK la pinul 3. Atenție!!! Placa Adafruit Feather 32U4 FONA necesită conectarea unui acumulator LiPo de 3.7V chiar dacă este alimentată prin intermediul cablului USB de programare. Componenta GSM FONA se alimentează direct din acumulator nu din alimentarea USB. Pentru mai multe detalii legate de funcționarea plăcii Adafruit Feather 32U4 FONA se poate parcurge materialul „Adafruit Feather 32u4 FONA - Take your Feather anywhere in the world” (4).



Funcționarea sistemului necesită o cartelă GSM 2G cu capabilități de transfer de date. Pentru acest lucru vă propunem utilizarea unui SIM Simfony Mobile M2M (5) – cartelă GSM ce oferă exclusiv servicii mobile de date. Cartela este disponibilă gratuit prin comandă pe site-ul companiei Simfony Mobile SRL (6) sau împreună cu un produs din gama GSM pe site-ul Robofun (7). Cartela necesită înregistrarea și introducerea codului promoțional pentru activare și oferă gratuit 10MB de date mobile valabile 3 luni. Ulterior costurile de funcționare sunt de 0.25EURO, 0.5EURO, 1EURO pentru 1MB, 5MB respectiv 10MB trafic de date. Chiar dacă traficul inclus are valori modice pentru un sistem de raportare IoT este suficient iar costurile sunt rezonabile. O caracteristică importantă a cartelei SIM Simfony este independența de un operator de telefonie anume, dispozitivul GSM ce utilizează cartela Simfony poate utiliza orice operator de telefonie mobilă în funcție de zona în care se află chiar și afara României.

Programul a fost dezvoltat și testat utilizând Arduino IDE 1.8.3 cu extensia Adafruit AVR Boards 1.4.9 instalată și bibliotecile Adafruit FONA Library 1.3.2 (pentru partea de comunicație cu componenta GSM FONA) și Adafruit SI1145 1.0.0 (pentru comunicația cu senzorul SI1145). În program este necesară completarea cheii de autentificare (*Token*) oferite de serviciul cloud (variabila *temp*).

```
#include "Adafruit_FONA.h"
#define FONA_RX 9
#define FONA_TX 8
#define FONA_RST 4
#define FONA_RI 7
#define FONA_DTR 5
#define apn "internet.simfony.net"
#define apnusername ""
```

```

#define apnpassword ""
char replybuffer[255];
#include <SoftwareSerial.h>
SoftwareSerial fonaSS = SoftwareSerial(FONA_TX, FONA_RX);
SoftwareSerial *fonaSerial = &fonaSS;
Adafruit_FONA fona = Adafruit_FONA(FONA_RST);
uint8_t type;
#include <Wire.h>
#include "Adafruit_SI1145.h"
Adafruit_SI1145 uv = Adafruit_SI1145();

```

**Decomentarea directivei #define debug va permite urmărirea funcționării sistemului în consola serială.**

```

//#define debug

void setup() {
  #ifdef debug
    while (!Serial);
    Serial.begin(115200);
    Serial.println("Initializing...");
  #endif
  if (! uv.begin()) {
    #ifdef debug
      Serial.println("Didn't find Si1145");
    #endif
    while (1);
  }
  digitalWrite(FONA_DTR, LOW);
  fonaSerial->begin(4800);
  if (! fona.begin(*fonaSerial)) {
    #ifdef debug

```

```

    Serial.println(F("Couldn't find FONA"));
#endif
    delay(1);
    while (1);
}
#ifdef debug
    type = fona.type();
    Serial.println(F("FONA is OK"));
    Serial.print(F("Found "));
    switch (type) {
        case FONA800L:
            Serial.println(F("FONA 800L")); break;
        case FONA800H:
            Serial.println(F("FONA 800H")); break;
        case FONA808_V1:
            Serial.println(F("FONA 808 (v1)")); break;
        case FONA808_V2:
            Serial.println(F("FONA 808 (v2)")); break;
        case FONA3G_A:
            Serial.println(F("FONA 3G (American)")); break;
        case FONA3G_E:
            Serial.println(F("FONA 3G (European)")); break;
        default:
            Serial.println(F("???")); break;
    }
#endif
#ifdef debug
    char imei[15] = {0};
    uint8_t imeiLen = fona.getIMEI(imei);
    if (imeiLen > 0) {
        Serial.print("Module IMEI: "); Serial.println(imei);
    }
}

```

```

#endif
fona.getSIMCCID(replybuffer);
#ifdef debug
    Serial.print(F("SIM CCID = "));
    Serial.println(replybuffer);
#endif
if (!fona.enableNetworkTimeSync(true)) {
    #ifdef debug
        Serial.println(F("Failed to enable NTS"));
    #else
        ;
    #endif
}
delay(5000);
fona.setGPRSNetworkSettings(F(apn), F(apnusername),
                             F(apnpassword));

uint8_t n=0;
#ifdef debug
    Serial.print("Connecting to network.");
#endif
while (n!=5) {
    n = fona.getNetworkStatus();
    #ifdef debug
        Serial.print(".");
    #endif
    delay(1000);
}
#ifdef debug
    Serial.println("OK");
#endif
#ifdef debug
    n = fona.getRSSI();

```

```

    int8_t r;
    if (n == 0) r = -115;
    if (n == 1) r = -111;
    if (n == 31) r = -52;
    if ((n >= 2) && (n <= 30)) { r = map(n, 2, 30, -110, -
                                     54); }

    Serial.print(r); Serial.println(F("dBm"));
#endif
    delay(5000);
}

void loop() {
    while (!fona.enableGPRS(true)) {
        #ifdef debug
            Serial.println(F("Failed to turn on GPRS"));
        #endif
        delay(5000);
    }
#ifdef debug
    uint16_t vbat;
    uint16_t pbat;
    if (fona.getBattVoltage(&vbat)) {
        Serial.print(F("VBat = "));
        Serial.print(vbat); Serial.println(F(" mV")); }
    if (fona.getBattPercent(&pbat)) {
        Serial.print(F("VPct = "));
        Serial.print(pbat); Serial.println(F("%")); }
    Serial.println("-----");
#endif
    float UVindex = uv.readUV();
    UVindex /= 100.0;
#ifdef debug
    int light = uv.readVisible();

```



```

        Serial.print("Vis: "); Serial.println(light);
        Serial.print("UV: "); Serial.println(UVindex);
    #endif
    String temp = "http://iot.robofun.ro/api/v1/input/.../" +
                  String(UVindex,2);

    uint16_t statuscode;
    int16_t length;
    char url[100];
    temp.toCharArray(url,temp.length()+1);
    #ifdef debug
        Serial.println(url);
        if (!fona.HTTP_GET_start(url, &statuscode, (uint16_t
                                                    *)&length))
            Serial.println("Failed read HTTP!");
    #else
        fona.HTTP_GET_start(url, &statuscode, (uint16_t
                                                    *)&length);
    #endif
    while (length > 0) {
        while (fona.available()) {
            char c = fona.read();
            #ifdef debug
                Serial.write(c);
            #endif
            length--;
            if (! length) break;
        }
        #ifdef debug
            Serial.println();
        #endif
        break;
    }
}

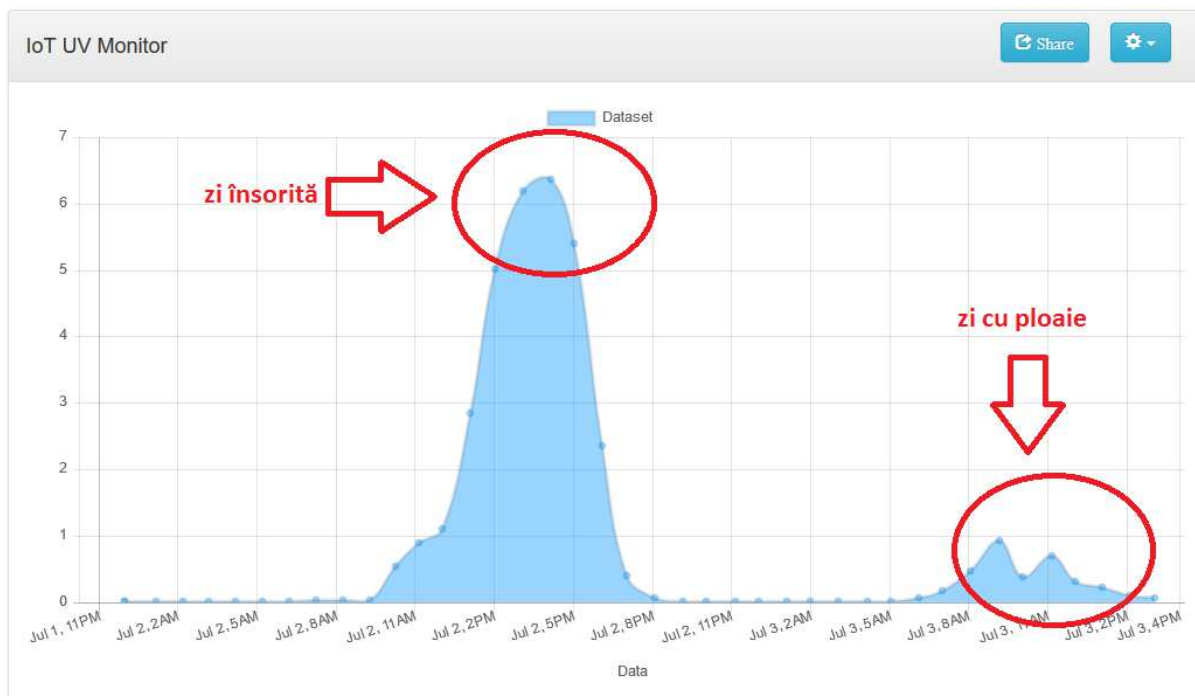
```

```

fona.HTTP_GET_end();
delay(100);
#ifdef debug
    Serial.println("-----");
    if (!fona.enableGPRS(false))
        Serial.println(F("Failed to turn off GPRS"));
#else
    fona.enableGPRS(false);
#endif
delay(100);
digitalWrite(FONA_DTR,HIGH);
delay(600000);
digitalWrite(FONA_DTR,LOW);
}

```

Programul va citi valorile indexului UV și va înregistra datele on-line la un interval de 10 minute (60 secunde = 60000 milisecunde). Intervalul de măsurare poate fi modificat în funcție de dinamica monitorizării dorite. Graficul de supraveghere va arăta în felul următor (se poate observa că, într-o zi însorită, indexul UV poate ajunge la valori periculoase în intervalul 15:00-17:00 iar într-o zi noroasă / cu ploaie rămâne la valori subunitare):



# Referințe on-line

(1) Robofun IoT

<http://iot.robofun.ro/>

(2) Feather 32u4 FONA

[https://www.robofun.ro/wireless/wireless-gsm/feather-32u4-fona?utm\\_medium=email&utm\\_content=productLink&utm\\_campaign=CURS\\_EMAIL](https://www.robofun.ro/wireless/wireless-gsm/feather-32u4-fona?utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL)

(3) Senzor de lumina/IR/UV Si1145

[https://www.robofun.ro/senzori/lumina/si1145-digital-uv-index-ir-sensor?utm\\_source=newsletter&utm\\_medium=email&utm\\_content=productLink&utm\\_campaign=CURS\\_EMAIL](https://www.robofun.ro/senzori/lumina/si1145-digital-uv-index-ir-sensor?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL)

(4) Adafruit Feather 32u4 FONA - Take your Feather anywhere in the world

<https://learn.adafruit.com/adafruit-feather-32u4-fona>

(5) SIM Simfony Mobile M2M

[https://www.robofun.ro/wireless/wireless-gsm/sim-simfony-mobile?utm\\_source=newsletter&utm\\_medium=email&utm\\_content=productLink&utm\\_campaign=CURS\\_EMAIL](https://www.robofun.ro/wireless/wireless-gsm/sim-simfony-mobile?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL)

(6) M2M Global Connectivity, Internet of Things Platform | Simfony Mobile

<http://simfonymobile.com/>

(7) Produe GSM

[https://www.robofun.ro/wireless/wireless-gsm?utm\\_source=newsletter&utm\\_medium=email&utm\\_content=productLink&utm\\_campaign=CURS\\_EMAIL](https://www.robofun.ro/wireless/wireless-gsm?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL)