

## Proiect Environmental Station

---

### Componente necesare realizării proiectului

- Modul interior
  - Placă de dezvoltare Reduino RF
  - Senzor de temperatură și umiditate HTU21D
  - LCD grafic monocrom Nokia 5110
  - Regulator step-down 3.3V D24V6F3
  - Fire de interconectare
- Modul exterior
  - Placă de dezvoltare Reduino RF
  - Senzor de presiune BMP0180
  - Senzor UV Si1145
  - 1 LED + rezistor 220ohm
  - Regulator step-up U1V10F3 3.3V
  - Baterie AA 1.5V
  - Fire de interconectare
- Conector FTDI 3.3V + cablu usb

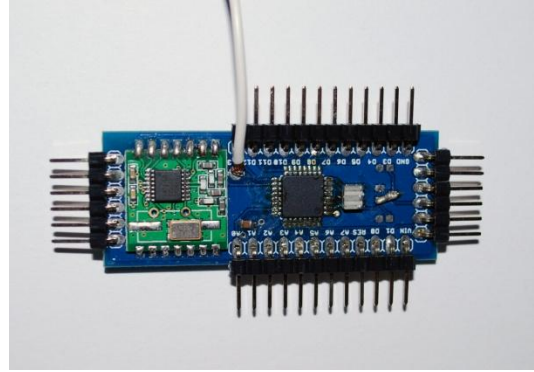
### Descriere proiect

Dispozitivele de tip stație meteo de uz domestic reprezintă deja o categorie de dispozitive larg răspândite acoperind o gamă destul de mare de performanțe: de la simple dispozitive care măsoară temperatura din cameră și până la dispozitive semiprofesionale care furnizează date complexe precum temperatura interioară / exterioară, umiditatea aerului, presiunea atmosferică, predicția evoluției vremii ș.a.m.d. Proiectul Environmental Station își propune să implementeze un sistem care să măsoare și să furnizeze utilizatorului o gamă de parametri de mediu utili în viața de zi cu zi: temperatura interioară și exterioară, umiditatea interioară, evoluția presiunii atmosferice și factorul de radiații ultraviolete. Sistemul este format din două module: modulul interior va măsura temperatura și umiditatea ambientală și se va ocupa cu colectarea și afișarea informațiilor de la modulul exterior; modulul exterior va colecta și va transmite radio către modulul interior măsurători pentru parametri de temperatură, presiune atmosferică și nivelul radiațiilor ultraviolete. Comunicația radio între modulul exterior și cel interior se va realiza în bandă ISM de 433MHz FSK prin intermediul modului radio HopeRF RFM12B [1]. Placa de dezvoltare utilizată pentru implementarea atât a modului extern cât și a celui intern este o placă de tip Reduino RF [2] ce include, pe lângă microcontrolerul ATmega 328P, un modul radio RFM12B – placa este o combinație între o placă Arduino Pro Mini și un modul radio RFM12B. Utilizarea acestei plăci este la fel de facilă ca utilizarea unei plăci Arduino Pro Mini dar nu

---

---

necesită conectarea unui modul radio extern – placa de dezvoltare Reduino RF se poate înlocui în cadrul proiectului cu o combinație Arduino Pro Mini + modul RFM12B sau Arduino Uno + modul RFM12B dar costul, complexitatea și dimensiunea sistemului vor crește. Placa Reduino RF funcționează la o tensiune de 3.3V iar microcontrolerul ATmega 328, aflat pe placa de dezvoltare, la o frecvență de 8MHz. Pentru programarea acestei plăci este necesară utilizarea unui conector FTDI 3.3V [3], în mediul Arduino IDE la încărcare programului se va selecta la Board: *Arduino Pro or Pro Mini* și la procesor: *ATmega328 (3.3V, 8MHz)* – placa Reduino FM este preprogramată cu un bootloader identic cu al plăcii Arduino Pro Mini.



## Biblioteci software utilizate în cadrul proiectului

Principala componentă software utilizată în cadrul proiectului este biblioteca JeeLib [4] ce permite lucrul cu modulul radio HopeRF RFM12B [1]. Această bibliotecă este o colecție de componente software și exemple destinate mediului Arduino IDE și componentelor integrate plăci de dezvoltare Arduino și componente hardware diverse (senzori, afișaje, componente I2C). Elementul principal al bibliotecii îl reprezintă *RF12 wireless driver* implementat în fișierele sursă ale bibliotecii *RF12.h* și *RF12.cpp*, element ce implementează atât primitive de acces la componenta hardware RFM12B (inițializare, transmisie, recepție, verificare stare) cât și un protocol de comunicație radio complex bazat pe adrese ale nodurilor, rețele în care se pot separa nodurile și metode de retransmisie ale mesajelor pentru a crește aria de acoperire. Din cadrul acestei biblioteci vom utiliza în cadrul proiectului atât componenta *RF12 wireless driver* cât și componenta *Sleepy* – clasă ce implementează partea de management a modurilor de consum redus ale microcontrolerului, astfel vom putea implementa în cadrul modulului exterior ce va funcționa pe baterie o modalitate de economisire a puterii consumate.

În cadrul modulului interior utilizarea afișajului LCD grafic monocrom de Nokia 5110 [5] (în fapt LCD-ul de Nokia 5110 se bazează pe controlerul PCD8544) va necesita utilizarea a două biblioteci software: Adafruit GFX [6] și Adafruit PCD8554 Nokia LCD [7]. Biblioteca Adafruit GFX implementează funcțiile grafice de bază cu lucrul cu un ecran LCD grafic (punct, linie, cerc, dreptunghi, text ș.a.m.d.) iar biblioteca Adafruit PCD8554 implementează interfațarea hardware cu controlerul PCD8554. Prin această spargere funcțională între cele două biblioteci codul scris pentru un ecran grafic poate fi portat cu ușurință pe alte modele de afișaje. Alte biblioteci compatibile cu Adafruit GFX: Adafruit SSD1306 (OLED monocrom 128x64 sau 128x32), Adafruit SSD1331 (0.96" OLED color), Adafruit Graphic VFD Library (Vacuum fluorescent display), Adafruit ST7735 și Adafruit HX8340B (TFT color), RGB Matrix Panel (matrice de leduri RGB).

---

---

Chiar dacă toți senzorii utilizați în proiect (în modulul interior și în modulul exterior) sunt senzori digitali cu comunicație I2C, pentru a ușura dezvoltarea programului, se vor utiliza biblioteci specifice pentru fiecare senzor în parte și anume:

- Pentru senzorul de temperatură și umiditate HTU21D [8], utilizat în modulul interior, vom utiliza biblioteca HTU21D Library [9];
- Pentru senzorul de presiune atmosferică BMP180 [10], utilizat în modulul exterior, vom utiliza biblioteca SFE BMP180 [11];
- Pentru senzorul UV Si1145 [12], utilizat în modulul exterior, vom utiliza biblioteca Adafruit Si1145 Library [13].

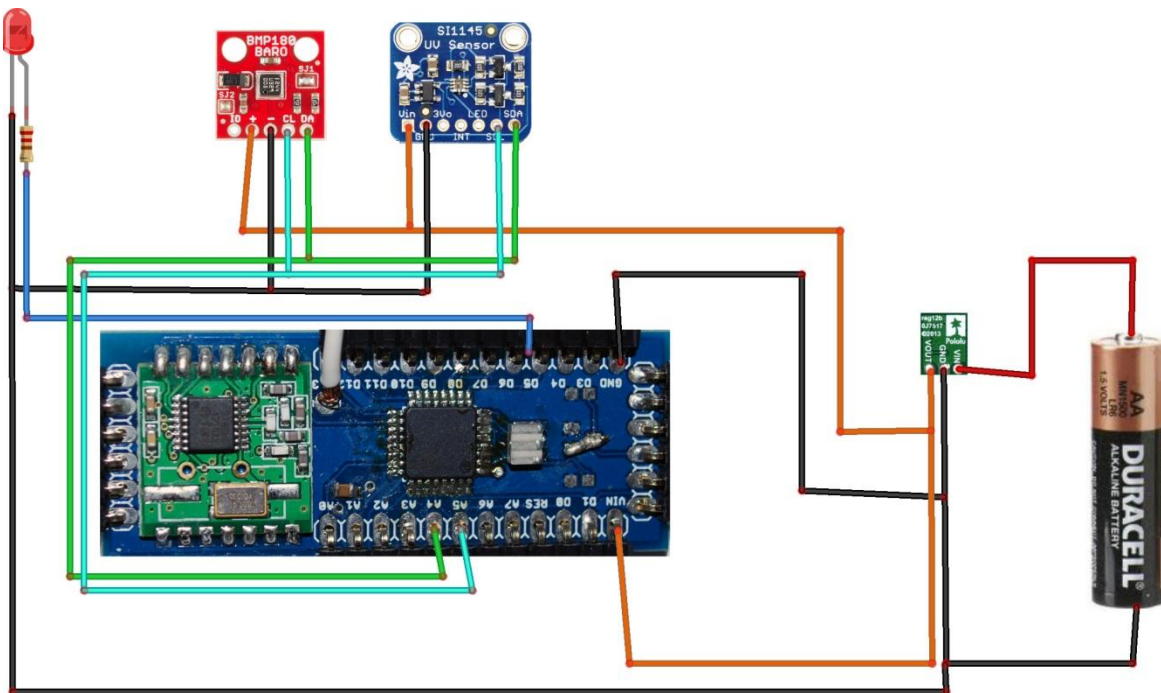
Toate bibliotecile software prezentate trebuie copiate și instalate în directorul de biblioteci a mediului Arduino IDE înainte de a testa exemplele software următoare.

## Arhitectura și modul de funcționare a modului exterior

Așa cum am precizat anterior modulul de exterior se bazează pe placa de dezvoltare Reduino FM [2] și are ca funcționalitate colectarea parametrilor de temperatură, nivel radiații ultraviolete, presiune atmosferică și transmiterea acestora către modulul interior.

Schema electrică (pagina următoare) este foarte simplă având în vedere că ambii senzori funcționează prin intermediul protocolului I2C. Alimentarea modului se va face prin intermediul unei baterii AA de 1.5V conectată la regulatorul step-up Polololu U1V10F3 [14] care va furniza tensiunea necesară (3.3V) funcționării plăcii de dezvoltare și celor doi senzori – alegerea unui alt sistem de alimentare ce furnizează o tensiune stabilizată de 3.3V nu va influența în nici un fel funcționarea sistemului. Conexiunile care trebuie realizate între componente sunt următoarele:

- Masa montajului – se interconectează ”-”-ul bateriei cu pinul GND al regulatorului de tensiune, al plăcii de dezvoltare și al celor doi senzori precum și cu catodul componentei LED (piciorul mai scurt);
  - 1.5V – se leagă ”+”-ul bateriei la pinul Vin al regulatorului de tensiune;
  - 3.3V – se conectează pinul Vout al regulatorului de tensiune la Vin al plăcii de dezvoltare și al celor doi senzori;
  - Comandă LED – se conectează anodul componentei LED prin intermediul unei rezistențe de 220ohm la pinul digital 5 al plăcii de dezvoltare;
  - SCK (I2C) – se conectează pinul CLK (CL sau SCL) la pinul A5 al plăcii de dezvoltare;
  - Data (I2C) – se conectează pinul SDA (DA sau SDA) la pinul A5 al plăcii de dezvoltare;
-



Programul poate fi încărcat pe placa de dezvoltare înainte sau după realizarea montajului electronic. În cazul în care este încărcat după realizarea montajului este bine ca bateria să fie deconectată pe durata conectorului FTDI – conectorul FTDI asigură la rândul său tensiune de alimentare plăcii de dezvoltare.

Programul va conține în partea de început includerea de biblioteci, definirea parametrilor (constantelor) de funcționare și instanțierea obiectelor software necesare lucrului cu modulele hardware. Definițiile *myNodeID*, *network* și *freq* sunt specifice bibliotecii JeeLib și specifică adresa nodului, rețeaua din care face parte și frecvența ISM pe care se va realiza comunicația radio. Definiția *led* specifică pinul pe care se va realiza comanda LED-ului.

```
#include <JeeLib.h>
#include <Wire.h>
#include "Adafruit_SI1145.h"
#include <SFE_BMP180.h>

#define myNodeID 10
#define network 210
#define freq RF12_433MHZ

#define led 5

Adafruit_SI1145 uv = Adafruit_SI1145();

SFE_BMP180 pressure;
```

---

---

În cadrul comunicației radio vom utiliza o structură particulară, *ParametriiTX*, ce conține exact parametri ce vor fi măsurați de modulul exterior și trimiși către modulul interior.

```
typedef struct { int lumina, ir; float uv_index; double presiune,
                temp; boolean lowbat;} ParametriiTX;

ParametriiTX parametrii;
```

În cadrul procedurii setup se vor inițializa componentele hardware ale sistemului: pinul de comandă a componentei LED, modulul radio, senzorul de ultraviolete și senzorul de presiune. În cazul în care unul dintre cei doi senzori nu se inițializează corect se va intra într-o stare blocantă de avarie în care componenta LED va clipi la un interval de o secundă. Componenta LED are rolul de minimă interfață utilizator pentru modulul exterior – va indica starea de avarie prin aprindere intermitentă și se va aprinde pe durata achiziționării datelor de la senzori și trimiterea valorilor achiziționate prin conexiunea radio. Neconectarea componentei LED în cadrul sistemului nu va influența celelalte funcționalități ale acestuia.

```
void setup() {
    pinMode(led, OUTPUT);
    digitalWrite(led, LOW);
    rfl2_initialize(myNodeID, freq, network);
    if (!uv.begin()) avarie();
    if (!pressure.begin()) avarie();
}

void avarie() {
    while(1) {
        digitalWrite(led, HIGH);
        delay(1000);
        digitalWrite(led, LOW);
        delay(1000);
    }
}
```

În cadrul secțiunii principale a programului se vor achiziționa următorii parametri:

- de la senzorul Si1145: nivelul radiațiilor luminoase vizibile, infraroșii și ultraviolete (indexul UV va fi calculat prin împărțirea la 100 a valorii achiziționate);
- de la senzorul BMP180: temperatura și presiunea atmosferică;
- de la modulul radio RFM12B starea bateriei – dacă tensiunea de alimentare a scăzut sau nu sub pragul critic de 3.1V.

Cu valorile achiziționate se populează structura *parametrii* și se transmite prin radio. După fiecare ciclu de achiziție se intră pentru o perioadă de 65 secunde într-o stare de consum redus (sleep) pentru a conserva pentru o perioadă mai mare energia bateriei. Prin această metodă sistemul poate funcționa până la câteva luni pe baza unei singure baterii AA. Funcționarea modului de economisire a energiei se bazează pe mecanismul intern de watchdog [15] și este

---

---

implementat prin funcția *Sleepy::loseSomeTime* a bibliotecii JeeLib și necesită implementarea subrutinei *ISR(WDT\_vect)*. Din acest motiv modulul exterior va trimite către modul interior informații la un interval de 65 de secunde – acest interval poate fi micșorat sau mărit în funcție de necesități și în funcție de durata de autonomie dorită a sistemului.

```
void loop() {
    double P,T;
    char status;

    digitalWrite(led,HIGH);

    parametrii.lumina = uv.readVisible();
    parametrii.ir = uv.readIR();
    parametrii.uv_index = uv.readUV();
    parametrii.uv_index /= 100.0;

    status = pressure.startTemperature();

    if (status != 0)
    {
        delay(status);
        status = pressure.getTemperature(T);
        parametrii.temp = T;
        if (status != 0)
        {
            status = pressure.startPressure(3);
            if (status != 0)
            {
                delay(status);
                status = pressure.getPressure(P,T);
                if (status != 0)
                {
                    parametrii.presiune = P;
                }
            }
        }
    }

    parametrii.lowbat = rf12_lowbat();

    rf12_sendNow(0, &parametrii, sizeof parametrii);
    rf12_sendWait(0);
    digitalWrite(led,LOW);

    rf12_sleep(RF12_SLEEP);
    Sleepy::loseSomeTime(65000);
}
```

---



---

```
    rf12_sleep(RF12_WAKEUP);  
}  
  
ISR(WDT_vect) { Sleepy::watchdogEvent(); }
```

Distanța între modulul exterior și cel interior poate ajunge la maxim 100 de metri în câmp deschis (fără obstacole). În cazul utilizării modulelor radio RFM12B în clădiri această distanță maximă scade destul de mult în funcție de natura obiectelor (ziduri, dulapuri) aflate între modulele radio – în orice caz, la distanțe rezonabil de mici (până în 10 metri) nu ar trebui să existe probleme de comunicație. Modulul exterior a fost gândit să funcționeze... în exterior (expus direct radiațiilor solare pentru a putea măsura nivelul radiațiilor UV) dar pentru a putea face față condițiilor de mediu externe trebuie să fie protejat de o carcasă adecvată.

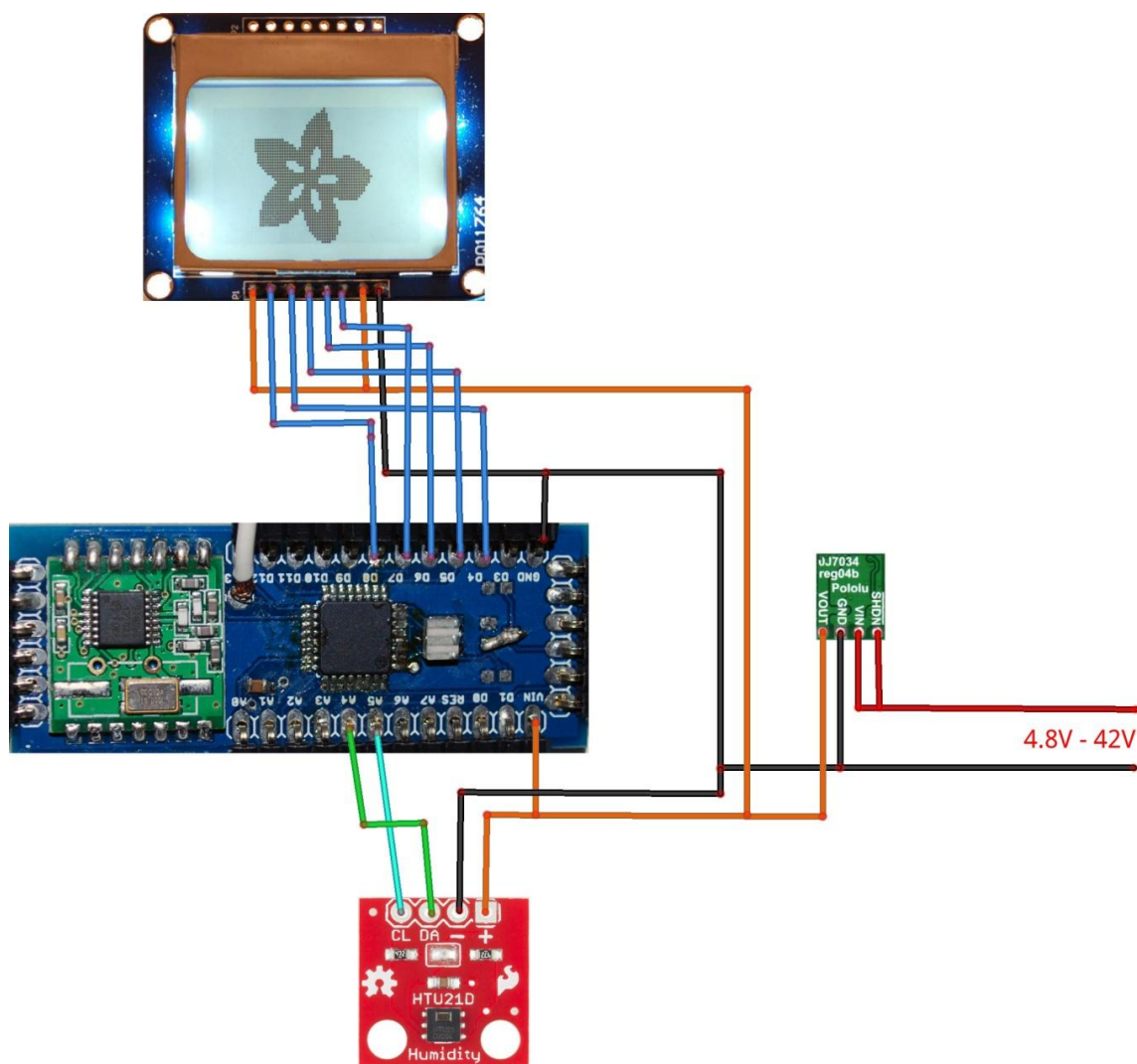
## Arhitectura și modul de funcționare a modului interior

Modulul interior din cadrul proiectului Environmental Station are rolul de a afișa parametri achiziționați de modulul exterior și de senzorul de temperatură și umiditate propriu. Pentru a diferenția cele două seturi de parametri (temperatură exterioară / temperatură interioară) afișarea se va face succesiv: un ecran cu parametri din interior (pe fundal negru, text de culoare albă – prima imagine alăturată) și un ecran cu parametri exteriori (pe fundal alb – a doua imagine alăturată). Alternarea între cele două ecrane se va face în mod automat la aproximativ 10 secunde. Ecranul cu parametri din interior va afișa temperatura interioară (în grade Celsius) cu o dimensiune de text mai mare și umiditatea interioară (în procente) cu o dimensiune de text mai mică. În cazul în care umiditatea interioară va depăși pragul de 50%, în fața valorii relative a parametrului umiditate va fi afișată o formă grafică de tip "picătură" (nu apare în imaginea alăturată deoarece nu este cazul); forma grafică "picătură" avertizează asupra probabilității mari de apariție a condensului în incinta interioară dacă există obiecte mai reci decât temperatura ambientală (cazul perioadelor de iarnă în care ferestrele sau pereții exteriori sunt mai reci decât temperatura aerului din interior). Ecranul cu parametri exteriori va afișa în aceeași manieră temperatura exterioară și în partea de jos a ecranului va afișa indexul UV [16] (în caseta neagră – în cazul imaginii anterioare este un index mic specific unei zile înnoirate) și graficul de evoluție a parametrului presiune atmosferică. Parametrul presiune atmosferică nu va fi afișat sub formă numerică (valoare) ci sub forma unui grafic de evoluție față de valoarea medie a ultimilor 35 de măsurători. Din acest motiv graficul va fi disponibil după efectuarea primelor 35 de măsurători (având în vedere frecvența cu care se face achiziția – graficul va fi disponibil după aproximativ 40 de minute de la pornirea sistemului). Prin intermediul acestui grafic se poate vizualiza tendința de creștere sau scădere a parametrului presiune atmosferică pe ultima perioadă de



timp și astfel se poate previziona evoluția vremii în viitorul apropiat. Bara verticală prezentă în grafic este poziția imediat următoare ultimei valori achiziționate – la stânga sunt ultimele valori achiziționate, la dreapta valori mai vechi. Între zona de afișare a indexului UV și zona cu graficul evoluției parametrului este o zonă în care se va afișa un simbol grafic de baterie descărcată în cazul în care modulul exterior va semnaliza acest lucru (nu este cazul imaginii de ecran anterioare).

Schema electrică a modulului interior este prezentată în imaginea următoare. După cum se poate observa, ca și în cazul modulului exterior, alimentarea sistemului presupune utilizarea unui regulator de tensiune de 3.3V; de această dată se va utiliza un regulator step-down Pololu D24V6F3 [17] ce poate primi ca intrare o tensiune între 4.8V și 42V – sistemul poate fi alimentat de la orice alimentator între 5V și 40V, inclusiv de la o mufă USB. La fel ca și în cazul modulului exterior modificarea soluției de alimentare nu schimbă cu nimic funcționalitatea sistemului (atâta timp cât se asigură o tensiune de alimentare stabilizată de 3.3V).





---

Legăturile care trebuie realizate în schema electrică sunt următoarele:

- Tensiune de intrare 4.8V – 42V (+) se conectează la pinul Vin și la pinul /SHDN al regulatorului de tensiune. Pinul /SHDN permite implementarea funcției de oprire a alimentării în cazul supraîncălzirii sistemului – nu este utilizat în modulul nostru și trebuie conectat la tensiunea de intrare a regulatorului. Dacă este lăsat neconectat, fiind activ pe "0", regulatorul este în stare de blocare și nu furnizează la ieșire tensiunea necesară.
- Masă modul: masa alimentării externe (-) se conectează la pinul GND al regulatorului de tensiune, la pinul GND al plăcii de dezvoltare, la pinul "-" al senzorului HTU21D și la pinul GND al afișajului LCD (pinul 8 de la stânga la dreapta).
- Tensiunea de ieșire a regulatorului de tensiune (3.3V, +) se conectează la pinul Vin al plăcii de dezvoltare, la pinul "+" al senzorului HTU21D, la pinul VCC al afișajului LCD (pinul 7 de la stânga la dreapta) și la pinul LED al afișajului LCD (pinul 1 de la stânga la dreapta). Pinul LED este de fapt alimentarea (+) celor 4 LED-uri ce asigură iluminarea de fundal. Dacă se dorește comanda aprinderii și stingerii iluminării de fundal din program se poate realiza aceste lucru cu ajutorul unui pin digital al plăcii de dezvoltare (comandă binară sau PWM) dar trebuie utilizat un tranzistor de amplificare a curentului deoarece cele patru LED-uri necesită un curent de 80mA (4\*20mA) – mult peste posibilitatea unui pin digital al plăcii de dezvoltare.
- SCK (I2C) – Pinul CL al senzorului HTU21D se conectează la pinul A5 al plăcii de dezvoltare.
- DATA (I2C) – Pinul DA al senzorului HTU21D se conectează la pinul A4 al plăcii de dezvoltare.
- RST LCD (pinul 2 de la stânga la dreapta) se conectează la pinul D8 al plăcii de dezvoltare. O altă variantă posibilă este conectarea directă la pinul de reset (RST) al plăcii de dezvoltare.
- CS LCD (pinul 3 de la stânga la dreapta) se conectează la pinul D4 al plăcii de dezvoltare. O altă variantă posibilă (pentru reducerea ocupării pinilor plăcii de dezvoltare) este conectarea semnalului de Chip Select la "1" logic (la 3.3V).
- D/C LCD (pinul 4 de la stânga la dreapta) se conectează la pinul D5 al plăcii de dezvoltare.
- DIN LCD (pinul 5 de la stânga la dreapta) se conectează la pinul D6 al plăcii de dezvoltare.
- CLK LCD (pinul 6 de la stânga la dreapta) se conectează la pinul D7 al plăcii de dezvoltare.

LCD-ul utilizat folosește o comunicație de tip SPI dar biblioteca Adafruit PCD8544 permite utilizarea oricăror pini pentru comunicația SPI (implementează protocolul SPI software) astfel nu suntem obligați să conectăm pinii CLK și DIN pe pinii D13, D12 (SCK, MISO) ai plăcii de dezvoltare (pe portul SPI hardware).

Similar cu cazul precedent, cazul modulului exterior, programul poate fi încărcat înainte sau după realizarea montajului electric dar în cazul în care încărcăm programul după realizarea

---

---

montajului trebuie să avem grijă ca sursa de tensiune externă să fie deconectată pentru a nu intra în conflict cu tensiunea furnizată de conectorul FTDI.

Secțiunea de început a programului ce va rula pe modulul interior conține includerea de biblioteci (*JeeLib*, *Adafruit\_GFX*, *Adafruit\_PCD8544*, *Wire*, *SPI*, *HTU21D*), definițiile necesare (*myNodeID*, *extNodeID*, *network*, *freq* – necesare configurării comunicației radio), instanțierea obiectelor prin intermediul cărora vom lucra cu componentele hardware (*display*, *myHTU*), descrierea celor două obiecte grafice afișate pe LCD (*picatura\_bmp*, *baterie\_bmp*), vectorul pentru cele 35 de valori ale presiunii atmosferice (*presiune*), structura de comunicație radio (mesaj) și alte variabile utilizate în program. Variabila *mod\_afisare* va indica dacă se face afișare pe LCD a parametrilor exteriori (11-20) sau a parametrilor interiori (1-10). Variabila *p* conține poziția curentă în care se va încărca următoarele măsurătoare pentru parametrul presiune atmosferică iar în variabila *medie\_presiune* se va calcula periodic media celor 35 de valori măsurate.

```
#include <JeeLib.h>
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_PCD8544.h>
#include <Wire.h>
#include "HTU21D.h"

#define myNodeID 20
#define extNodeID 10
#define network 210
#define freq RF12_433MHZ

Adafruit_PCD8544 display = Adafruit_PCD8544(7, 6, 5, 4, 8);

static const unsigned char PROGMEM picatura_bmp[] = {
    B00100000,
    B00100000,
    B01110000,
    B01110000,
    B11111000,
    B11111000,
    B11111000,
    B11111000,
    B01110000 };

static const unsigned char PROGMEM baterie_bmp[] = {
    B00111100,
    B00111100,
    B11111111,
    B11100111,
    B11100111,
```

---

---

```

        B11100111,
        B11100111,
        B11100111,
        B11100111,
        B11100111,
        B11111111 };
```

```

double presiune[35];

HTU21D myHTU;

typedef struct { int lumina, ir; float uv_index; double presiune,
temp; boolean lowbat;} PayloadTX;
PayloadTX mesaj;

byte mod_afisare = 0;
byte p = 0;
double medie_presiune;
```

În cadrul procedurii `setup` se inițializează componentele hardware (*display*, *myHTU*, *rf12*) și se inițializează cu zero vectorul ce va înregistra valorile pentru presiunea atmosferică (*presiune*).

```

void setup() {
    display.begin();
    myHTU.begin();
    rf12_initialize(myNodeID,freq,network);
    display.setContrast(60);
    display.display();
    delay(2000);
    display.clearDisplay();
    display.display();
    for (int i=0; i<35; i++) presiune[i]=0;
    delay(2000);
}
```

Secțiunea principală a programului, *loop*, are două funcționalități majore: afișarea pe LCD și citirea mesajelor primite prin comunicația radio de la modulul exterior. Afișarea pe LCD este condiționată de conținutul variabilei *mod\_afisare*. Modificarea informațiilor pe afișaj se face odată la 10 secunde (*mod\_afisare==1* informații din interior, *mod\_afisare==11* informații din exterior). Informațiile de pe ecran nu se împrăștează la fiecare rulare a procedurii *loop* (din secundă în secundă) pentru a nu produce efectul de clipire a LCD-ului și pentru a rula mai des secțiunea de cod aferentă recepției radio (secțiune de importanță ridicată). În cadrul afișării informațiilor din interior (*mod\_afisare==1*) se observă că se efectuează inclusiv citirea senzorului HTU21D. În cadrul afișării informațiilor din exterior (*mod\_afisare==11*) se utilizează structura *mesaj* (*.temp*, *.uv\_index*, *.lowbat*) și vectorul *presiune*, ambele inițializate în secțiunea

---

---

de recepție radio. Graficul evoluției parametrului presiune atmosferică se va afișa doar după inițializarea completă a vectorului *presiune* (*presiune[34]>0*). Toate valorile reprezentate în grafic sunt relative la valoarea medie a elementelor din vector (*medie\_presiune*).

```
void loop() {
    mod_afisare++;
    if (mod_afisare==1) {
        display.clearDisplay();
        display.fillScreen(BLACK);
        display.drawRoundRect(2,2,80,44,3,WHITE);
        display.setTextColor(WHITE);
        display.setCursor(6,10);
        display.setTextSize(2);
        display.print(myHTU.readTemperature(),1);
        display.setCursor(66,10);
        display.print("C");
        display.setCursor(57,10);
        display.setTextSize(1);
        display.print("o");
        display.setCursor(34,30);
        display.setTextSize(1);
        display.print(myHTU.readHumidity(),1); display.print("%");
        if (myHTU.readHumidity()>50.0) display.drawBitmap(25, 30,
                                                    picatura_bmp, 5, 8, WHITE);

        display.display();
    }
    else if (mod_afisare==11) {
        display.clearDisplay();
        display.drawRoundRect(2,2,80,44,3,BLACK);
        display.setTextColor(BLACK);
        display.setCursor(6,10);
        display.setTextSize(2);
        display.print(mesaj.temp,1);
        display.setCursor(66,10);
        display.print("C");
        display.setCursor(57,10);
        display.setTextSize(1);
        display.print("o");
        display.fillRoundRect(8,30,21,11,1,BLACK);
        display.setTextColor(WHITE);
        display.setCursor(10,32);
        display.print(mesaj.uv_index,1);
        if (mesaj.lowbat) display.drawBitmap(32, 30,  baterie_bmp, 8, 11,
                                                    BLACK);
        display.drawLine(41,35,75,35,BLACK);
    }
}
```

---

---

```
display.drawLine(41+p,40,41+p,30, BLACK);
if (presiune[34]>0) {
    medie_presiune=0;
    for (int i=0; i<35; i++) medie_presiune+=presiune[i];
    medie_presiune = medie_presiune/35;
    for (int i=0; i<35; i++) display.drawLine(41+i,35,41+i,
        35+(medie_presiune-presiune[i]),BLACK);
}
display.display();
}
else if (mod_afisare==21) mod_afisare=0;
```

Secțiunea de cod responsabilă cu recepția radio încarcă valorile primite în structura *mesaj*. La fiecare recepție reușită se transferă din structura *mesaj* parametrul presiune atmosferică în vectorul *presiune*.

```
if (rf12_recvDone()){
    if (rf12_crc == 0 && (rf12_hdr & RF12_HDR_CTL) == 0) {
        int node_id = (rf12_hdr & 0x1F);
        if (node_id == extNodeID) {
            mesaj=(PayloadTX*) rf12_data;
            presiune[p] = mesaj.presiune;
            p++;
            if (p==35) p=0;
        }
    }
}
delay(1000);
}
```

## Posibile dezvoltări ulterioare

Proiectul Environmental Station își propune să demonstreze capabilitățile de comunicație radio ale modului HopeRF RFM12B și ușurința cu care se pot realiza sisteme fără fir bazate pe placa de dezvoltare Reduino RF ce integrează tehnologia Arduino cu modulul RFM12B. Chiar dacă sistemul prezentat nu este format decât din două module (unul exterior și unul interior) iar comunicația este unidirecțională (de la exterior spre interior) tehnologia disponibilă în cadrul plăcii de dezvoltare integrate Reduino RF nu se limitează la acest exemplu simplu:

- Se pot realiza rețele de senzori fără fir cu un număr mare de noduri (câte 255 în fiecare rețea de comunicație fiind posibilă și comunicația între rețele multiple). Se pot imagina scenarii în care se pot monitoriza zone diverse ale unei clădiri sau ale unei hale industriale. Limitarea legată de distanța maximă oferită de comunicația radio ISM FSK
-

---

poate fi depășită prin metode de retransmisie a mesajelor de la un nod la altul (biblioteca JeeLib oferă suport pentru astfel de tehnici).

- Modulul HopeRF RFM12B este un modul care asigură atât transmisia cât și recepția radio fiind posibilă implementarea unor comunicații bidirecționale. De exemplu, în cadrul exemplului nostru era posibilă și afișarea parametrilor de mediu din interior de către modulul exterior nu numai a parametrilor de mediu din exterior către modulul interior. Se poate proiecta o structură în care fiecare nod centralizează și afișează toate informațiile provenite de la toate celelalte noduri.

În cazul proiectării unei rețele de senzori fără fir se poate implementa funcționalitatea de gateway pentru unul dintre noduri către alte protocoale și metode de comunicație (ethernet, bluetooth, Wi-Fi, ZigBee) făcând astfel disponibile informațiile colectate la distanțe mari sau pe dispozitive diverse (rețele de calculatoare, Internet, dispozitive mobile inteligente).

Tehnica de funcționare îndelungate pe baza unei singure baterii implementată în modulul exterior poate fi perfecționată astfel încât să asigure o autonomie mai mare a modulelor izolate sau o autonomie totală utilizând metode de alimentare alternative (energie solară sau eoliană).

Parametrii de mediu achiziționați în cadrul celor două module (temperatură, umiditate, presiune atmosferică, nivel radiații UV) pot fi completați prin utilizarea unei game mai mari de senzori specifici: senzori de nivel zgomot, calitate aer, radiații, vibrații, perturbații electromagnetice ș.a.m.d.

Proiectul de față își propune să fie un punct de plecare nu un sistem final. Pe baza lui se pot imagina o diversitate mare de sisteme de achiziție și comunicație fără fir bazate pe comunicația radio în bandă ISM FSK.

---



---

## Referințe online

[1] RFM12B wireless FSK transceiver module

[http://www.hoperf.com/rf/fsk\\_module/RFM12B.htm](http://www.hoperf.com/rf/fsk_module/RFM12B.htm)

[2] Reduino RF

<http://.....>

[3] Conector FTDI 3.3 V

[http://www.robofun.ro/platforme/arduino\\_dev/conector-ftdi-3V3](http://www.robofun.ro/platforme/arduino_dev/conector-ftdi-3V3)

[4] JeeLib - An easy interface to the port headers, the RF12 driver library, timers, low-power code, and more.

<http://jeelabs.net/pub/docs/jeelib/>

[5] Nokia 5110/3310 monochrome LCD

<http://www.adafruit.com/product/338>

[6] Adafruit GFX Graphics Library

<https://learn.adafruit.com/adafruit-gfx-graphics-library>

[7] Adafruit PCD8544 Nokia 5110 LCD library

<https://github.com/adafruit/Adafruit-PCD8544-Nokia-5110-LCD-library>

[8] Humidity Sensor Breakout - HTU21D

<https://www.sparkfun.com/products/12064>

[9] HTU21D Library

<http://dlnmh9ip6v2uc.cloudfront.net/datasheets/BreakoutBoards/HTU21DLibrary.zip>

[10] Barometric Pressure Sensor Breakout - BMP180

<https://www.sparkfun.com/products/11824>

[11] The latest BMP180 software archive

[https://github.com/sparkfun/BMP180\\_Breakout/archive/master.zip](https://github.com/sparkfun/BMP180_Breakout/archive/master.zip)

[12] Adafruit SI1145 Breakout Board - UV / IR / Visible Sensor

<https://learn.adafruit.com/adafruit-si1145-breakout-board-uv-ir-visible-sensor/overview>

[13] Adafruit Si1145 Library

[https://github.com/adafruit/Adafruit\\_SI1145\\_Library/archive/master.zip](https://github.com/adafruit/Adafruit_SI1145_Library/archive/master.zip)

[14] Pololu 3.3V Step-Up Voltage Regulator U1V10F3

<http://www.pololu.com/product/2563>

---

---

[15] Sleeping Arduino - Part 5 Wake Up Via The Watchdog Timer  
<http://donalmmorrissey.blogspot.ro/2010/04/sleeping-arduino-part-5-wake-up-via.html>

[16] Ultraviolet Radiation  
[http://www.ccohs.ca/oshanswers/phys\\_agents/ultravioletradiation.html](http://www.ccohs.ca/oshanswers/phys_agents/ultravioletradiation.html)

[17] Pololu 3.3V, 600mA Step-Down Voltage Regulator D24V6F3  
<http://www.pololu.com/product/2106>

---