

Textul și imaginile din acest document sunt licențiate

Attribution-NonCommercial-NoDerivs

CC BY-NC-ND



Codul sursă din acest document este licențiat

Public-Domain

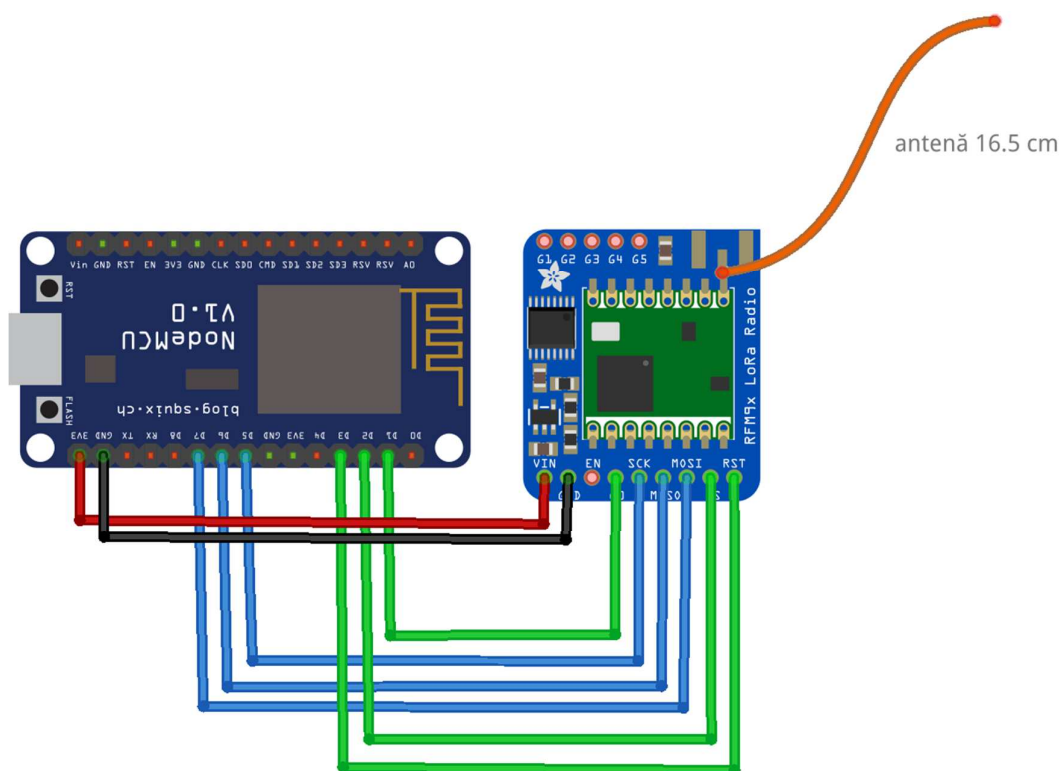
Ești liber să distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, tipărire, sau orice alt mijloc), atât timp cât nu aduci nici un fel de modificări acestuia. Codul sursă din acest document poate fi utilizat în orice fel de scop, de natură comercială sau nu, fără 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).

LoRa meets Robofun IoT



Modulele radio LoRa (1) oferă posibilitatea de a transmite date la distanță mare (sute de metri sau chiar kilometri) utilizând module electronice de cost redus și cu un consum de energie scăzut. Acest lucru constituie o metodă eficientă pentru a extinde aria de acoperire pentru rețelele IoT fără fir. Chiar dacă semnalul WiFi are o acoperire limitată fiind influențat de puterea dispozitivelor de tip AP și de mediul în care operează (câmp deschis, locuințe sau birouri) există posibilitatea să extindem aria de acoperire radio a unei rețele IoT prin intermediul comunicațiilor ISM (2) iar soluțiile LoRa oferă o variantă foarte bună cost / arie de acoperire.

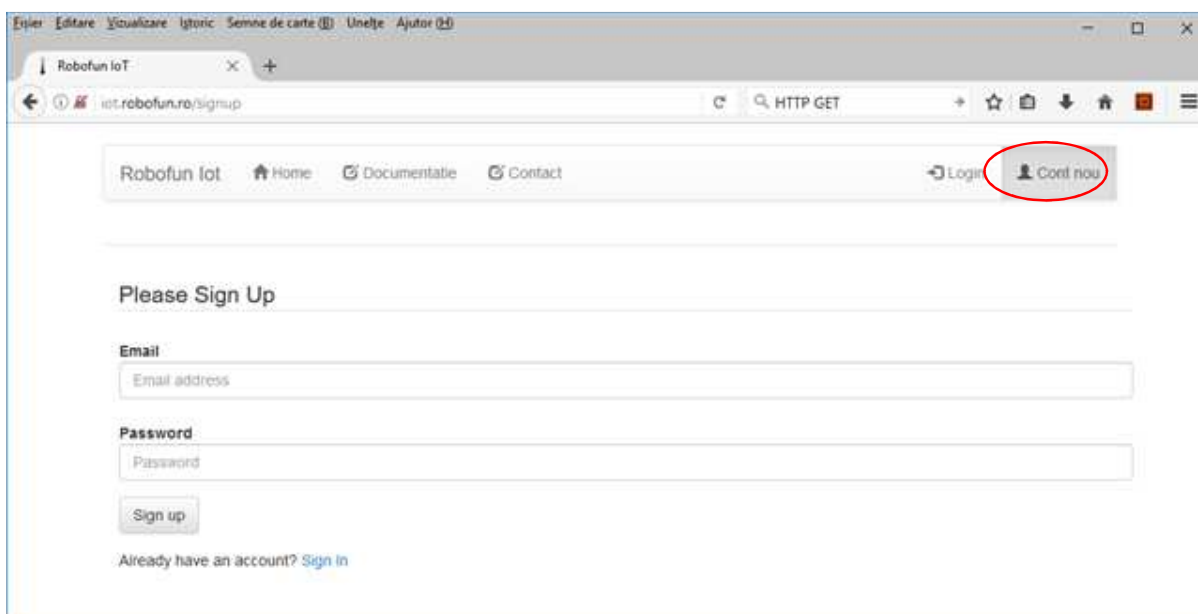
Pentru a implementa o soluție LoRa în vederea extinderii ariei de acoperire IoT vom implementa un sistem gateway ce va realiza transferul datelor provenite de la modulele IoT către un sistem specific IoT și anume Robofun IoT (3). Modulul gateway propus se bazează pe placa de dezvoltare NodeMCU (4) ce oferă conectivitate WiFi și un modul LoRa RFM96W (5) în bandă de 433MHz. Conexiunile între placa de dezvoltare și modulul radio sunt prezentate în diagrama următoare:



Modulul radio RFM96W se interconectează cu placa de dezvoltare prin intermediul magistralei SPI:

- Pinul SCK al modulului se conectează la pinul D5 (GPIO14 – HSCLK);
- Pinul MISO se conectează la pinul D6 (GPIO12 – HMISO);
- Pinul MOSI se conectează la pinul D7 (GPIO13 – HMOSI);
- Pinul CS se conectează la pinul D2;
- Pinul RST se conectează la pinul D3;
- Pinul G0 (INT) se conectează la pinul D1;
- Vin și GND la pinii 3.3V și GND ai plăcii de dezvoltare.

Pentru ca placa să poată transmite prin Internet datele către serviciul Robofun IoT (3) este necesară înregistrarea gratuită pe platformă:

A screenshot of a web browser showing the Robofun IoT registration page. The browser's address bar displays 'robofun.ro/signup'. The page has a navigation bar with links for 'Home', 'Documentatie', and 'Contact', along with 'Login' and 'Cont nou' buttons. The 'Cont nou' button is circled in red. Below the navigation bar, the heading 'Please Sign Up' is followed by input fields for 'Email' (with a placeholder 'Email address') and 'Password' (with a placeholder 'Password'). A 'Sign up' button is positioned below these fields. At the bottom, a link states 'Already have an account? Sign in'.

După înregistrare și conectare este necesară definirea (*Adauga senzor*) a trei senzori: Temperatura, Umiditate și Nivel baterie, pentru a putea transmite datele primite de la sistemul IoT către platforma online. După definirea fiecărui senzor este necesar să copiem cheia de autentificare (Token) pentru a fi utilizată în program.



Programul a fost dezvoltat și testat utilizând Arduino IDE 1.8.3 (pentru instalarea plăcii NodeMCU sub Arduino IDE se poate vedea materialul „Quick Start to Nodemcu (ESP8266) on Arduino IDE” (6)), extensia esp8266 versiunea 2.3.0 și biblioteca RadioHead 1.7.9 (7). În cadrul programului trebuie personalizate datele de conectare la rețeaua WiFi (*ssid* și *password*) precum și cheile de autentificare pentru cele trei canale IoT (*SENSOR_TOKEN1*, *SENSOR_TOKEN2* și *SENSOR_TOKEN3*).

```
#include <SPI.h>
#include <RH_RF95.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>

const char* ssid = "...";
const char* password = "...";

#define RFM95_CS D2
#define RFM95_RST D3
#define RFM95_INT D1
```

```

#define RF95_FREQ 434.0

RH_RF95 rf95(RFM95_CS, RFM95_INT);

#define LED D0

void setup() {
  pinMode(LED, OUTPUT);
  digitalWrite(LED,HIGH);
  pinMode(RFM95_RST, OUTPUT);
  digitalWrite(RFM95_RST, HIGH);
  Serial.begin(9600);
  delay(100);
  Serial.println();
  Serial.println();
  Serial.println("Gateway Module starting...");
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  digitalWrite(RFM95_RST, LOW);
  delay(10);
  digitalWrite(RFM95_RST, HIGH);
  delay(10);

  while (!rf95.init()) {
    Serial.println("LoRa radio init failed");
  }
}

```

```

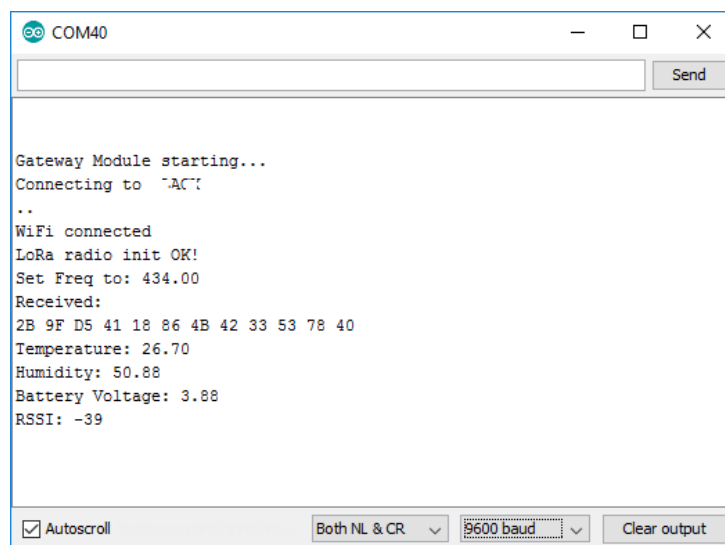
        while (1);
    }
    Serial.println("LoRa radio init OK!");

    if (!rf95.setFrequency(RF95_FREQ)) {
        Serial.println("setFrequency failed");
        while (1);
    }
    Serial.print("Set Freq to: ");
    Serial.println(RF95_FREQ);
}

typedef struct {float temperature; float humidity;
                float bat_voltage;} ParametriiRX;
ParametriiRX parametrii;

```

Ledul de pe placa de dezvoltare (conectat pe pinul D0) se va aprinde pe perioada recepției de mesaje LoRa. Toate datele primite prin intermediul LoRa vor fi postate pe platforma Robofun IoT. Sistemul va raporta și în consola serială toate mesajele recepționate.



```

void loop() {
    if (rf95.available())

```

```

{
    uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
    uint8_t len = sizeof(buf);
    if (rf95.recv(buf, &len)){
        digitalWrite(LED, LOW);
        RH_RF95::printBuffer("Received: ", buf, len);
        parametrii = *(ParametriiRX*)buf;
        Serial.print("Temperature: ");
        Serial.println(parametrii.temperature);
        Serial.print("Humidity: ");
        Serial.println(parametrii.humidity);
        Serial.print("Battery Voltage: ");
        Serial.println(parametrii.bat_voltage);
        Serial.print("RSSI: ");
        Serial.println(rf95.lastRssi(), DEC);
        digitalWrite(LED, HIGH);
        String SENSOR_TOKEN1="...";
        String SENSOR_TOKEN2="...";
        String SENSOR_TOKEN3="...";
        HTTPClient http;
        String data =
            String("http://iot.robofun.ro/api/v1/senzor/") +
            SENSOR_TOKEN1 + "/input?value=" +
            String(parametrii.temperature, DEC);
        http.begin(data);
        int httpCode = http.GET();
        http.end();
        data =
            String("http://iot.robofun.ro/api/v1/senzor/") +
            SENSOR_TOKEN2 + "/input?value=" +
            String(parametrii.humidity, DEC);
        http.begin(data);
    }
}

```

```

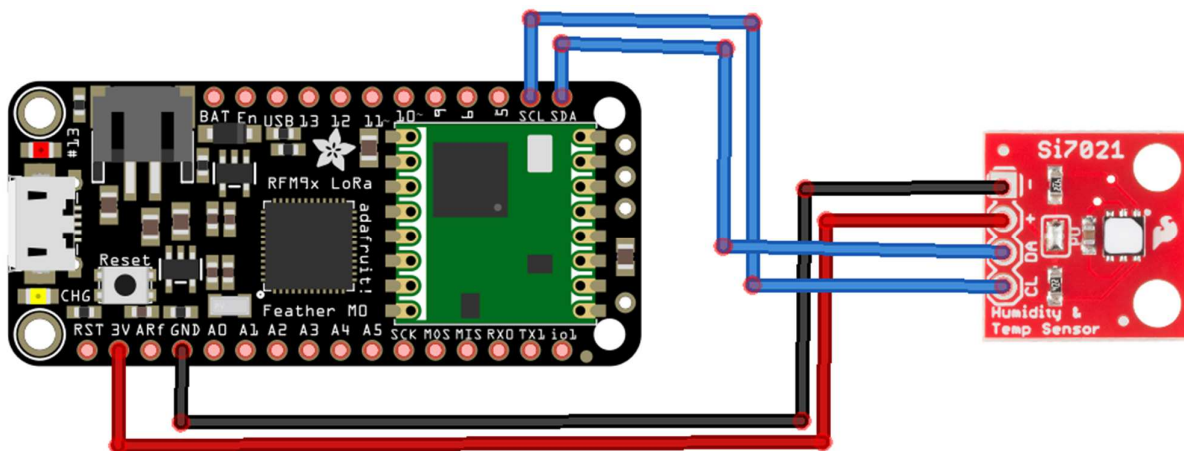
    httpCode = http.GET();
    http.end();
    data =
        String("http://iot.robofun.ro/api/v1/senzor/") +
        SENSOR_TOKEN3 + "/input?value=" +
        String(parametrii.bat_voltage, DEC);
    http.begin(data);
    httpCode = http.GET();
    http.end();
}
else
{
    Serial.println("Receive failed");
}
}
}

```

Modulul LoRa ce va beneficia de extinderea ariei de comunicație (se poate afla la sute de metri, sau chiar kilometri în spațiu deschis, de aria de acoperire WiFi) se bazează pe placa de dezvoltare Feather M0 RFM95 433MHz LoRa ([8](#)) – placă ce combină puterea unui microcontroler ARM Cortex-M0+ ATSAM21 (la fel ca și plăcile Arduino M0/Zero) cu conectivitatea unui modul radio LoRa. Placa de dezvoltare va transmite către modulul gateway date preluate de la un senzor digital I2C de temperatură și umiditate Si7021 ([9](#)) precum și nivelul bateriei proprii (unul dintre avantajele majore ale plăcilor Feather este posibilitatea de alimentare mobilă de la un acumulator LiPo de 3.7V).

Pentru mai multe informații legate de utilizarea plăcii Feather M0 RFM95 puteți consulta și materialul: „Adafruit Feather M0 Radio with LoRa Radio Module” ([10](#)).

Schema de interconectare între senzor și placa de dezvoltare este specifică unei magistrale I2C (pinii SDA și SCL conectați între senzor și placa de dezvoltare, alimentarea senzorului se va face la 3.3V):



Programul a fost dezvoltat și testat utilizând Arduino IDE 1.8.3, extensia Adafruit SAMD Boards 1.0.19 și bibliotecile RadioHead 1.7.9 ([7](#)), Adafruit Si7021(11).

Decomentarea directivei *debug* va permite observarea funcționării modului în consola serială.

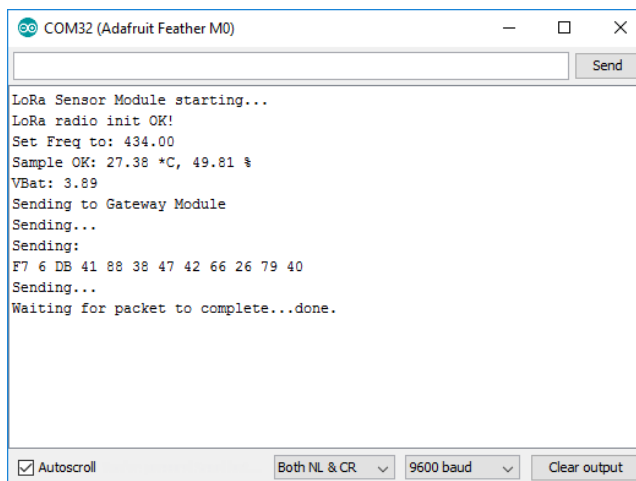
```
//#define debug

#include <SPI.h>
#include <RH_RF95.h>

#define VBATPIN A7

#include <Adafruit_Si7021.h>
Adafruit_Si7021 sensor = Adafruit_Si7021();

#define RFM95_CS 8
#define RFM95_RST 4
#define RFM95_INT 3
#define RF95_FREQ 434.0
RH_RF95 rf95(RFM95_CS, RFM95_INT);
```



```

void setup()
{
  pinMode(RFM95_RST, OUTPUT);
  digitalWrite(RFM95_RST, HIGH);
  sensor.begin();
#ifdef debug
  while (!Serial);
  Serial.begin(9600);
  delay(100);
  Serial.println("LoRa Sensor Module starting...");
#endif
  digitalWrite(RFM95_RST, LOW);
  delay(10);
  digitalWrite(RFM95_RST, HIGH);
  delay(10);
  while (!rf95.init()) {
    #ifdef debug
      Serial.println("LoRa radio init failed");
    #endif
    while (1);
  }
#ifdef debug
  Serial.println("LoRa radio init OK!");
#endif
  if (!rf95.setFrequency(RF95_FREQ)) {
    #ifdef debug
      Serial.println("setFrequency failed");
    #endif
    while (1);
  }
#ifdef debug
  Serial.print("Set Freq to: ");

```

```

    Serial.println(RF95_FREQ);
#endif
    rf95.setTxPower(23, false);
}

typedef struct {float temperature; float humidity;
               float bat_voltage;} ParametriiTX;
ParametriiTX parametrii;

void loop() {
    float voltage = analogRead(VBATPIN);
    voltage *= 2;
    voltage *= 3.3;
    voltage /= 1024;
    parametrii.bat_voltage = voltage;
    parametrii.temperature = sensor.readTemperature();
    parametrii.humidity = sensor.readHumidity();
#ifdef debug
    Serial.print("Sample OK: ");
    Serial.print(parametrii.temperature);
    Serial.print(" *C, ");
    Serial.print(parametrii.humidity);
    Serial.println(" %");
    Serial.print("VBat: " ); Serial.println(voltage);
    Serial.println("Sending to Gateway Module");
    Serial.println("Sending..."); delay(10);
#endif
    RH_RF95::printBuffer("Sending: ", (uint8_t*)
                        &parametrii, sizeof parametrii);
#ifdef debug
    Serial.println("Sending..."); delay(10);
#endif
}

```

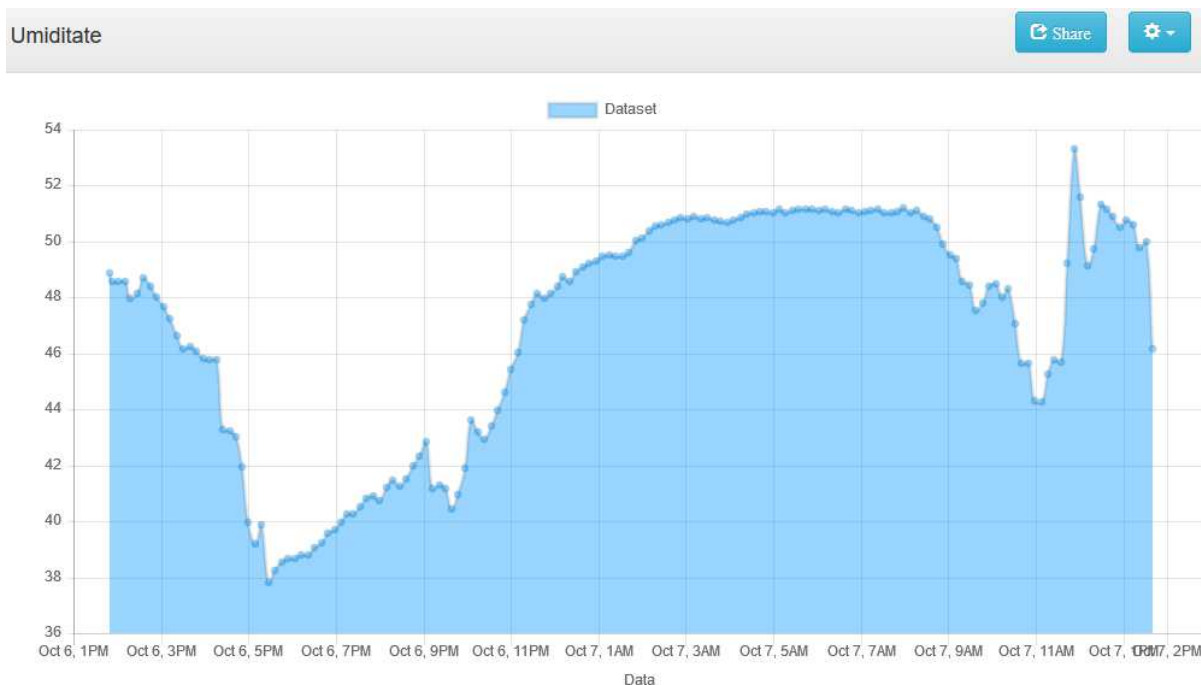
```

rf95.send((uint8_t *)&parametrii, sizeof parametrii);
#ifdef debug
    Serial.print("Waiting for packet to complete...");
    delay(10);
#endif
rf95.waitPacketSent();
#ifdef debug
    Serial.println("done."); delay(10);
#endif
delay(100);
rf95.sleep();
delay(60000);
}

```

După programarea și punerea în funcțiune a celor două sisteme se pot observa datele înregistrate în cadrul platformei Robofun IoT (capturile de ecran de mai jos reprezintă date înregistrate de sistemul de test pentru temperatură și umiditate).





Bineînțeles, sistemul LoRa poate deservi atât sisteme de achiziție (temperatură, umiditate, presiunea în diverse conducte, consumul de energie electrică, nivelul radiației solare etc.) dar și sisteme de acționare (chiar dacă nu a fost exemplificată această parte este posibil să comandăm de la distanță diverse mecanisme de închidere / deschidere, motoare etc).

Referințe on-line

(1) What Is LoRa?

<http://www.semtech.com/wireless-rf/internet-of-things/what-is-lora/>

(2) ISM band - Wikipedia

https://en.wikipedia.org/wiki/ISM_band

(3) Robofun IoT

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

(4) NodeMCU v2 - Lua ESP8266 development kit

https://www.robofun.ro/wireless/wireless-wifi/nodemcu-v2-lua-based-esp8266-development-kit?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

<https://www.robofun.ro/forum/>

(5) RFM96W Radio Transceiver - 433 MHz LoRA

https://www.robofun.ro/wireless/wireless-433/adafruit-rfm96w-lora-radio-transceiver-breakout-433-mhz/?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(6) Quick Start to Nodemcu (ESP8266) on Arduino IDE: 3 Steps

<http://www.instructables.com/id/Quick-Start-to-Nodemcu-ESP8266-on-Arduino-IDE/>

(7) RadioHead: RadioHead Packet Radio library for embedded microprocessors

<http://www.airspayce.com/mikem/arduino/RadioHead/>

(8) Feather M0 RFM95 Radio - 433MHz LoRA

https://www.robofun.ro/platforme/feather/adafruit-feather-m0-rfm95-lora-radio-433mhz/?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(9) Senzor Umiditate si Temperatura - Si7021

https://www.robofun.ro/senzori/vreme/enzor-umiditate-si-temperatura-si7021/?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(10) Adafruit Feather M0 Radio with LoRa Radio Module

<https://learn.adafruit.com/adafruit-feather-m0-radio-with-lora-radio-module>

(11) GitHub - adafruit/Adafruit_Si7021: Arduino library for Adafruit Si7021

https://github.com/adafruit/Adafruit_Si7021