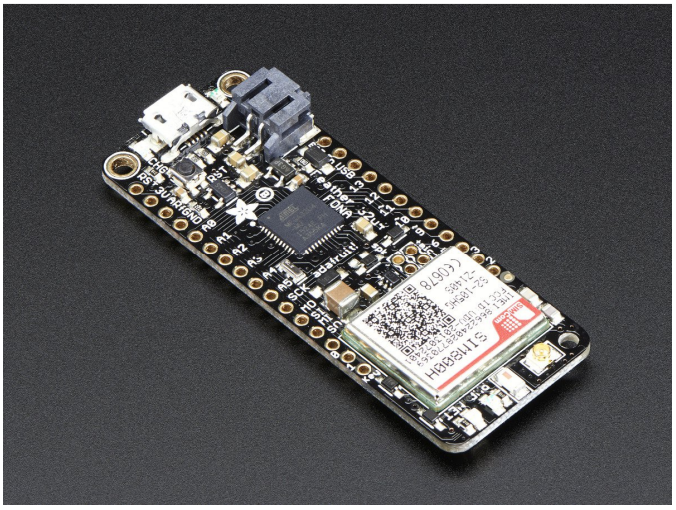
Termometru Simfonic

În cadrul lecției de față vom prezenta o soluție de achiziție (ca exemplificare vom face achiziția temperaturii și umidității mediul ambiant) cu raportare IoT prin GPRS (rețea de telefonie mobilă). Soluția propusă este una simplă și vine în întâmpinarea unei probleme extrem de actuală: supravegherea de la distanță a unor parametrii de mediu în locații izolate unde nu dispunem de infrastructură de rețea Internet – în loc de temperatură și umditate putem face achiziția unor parametrii ai solului (aplicații pentru agricultură inteligentă) sau achiziția unor parametrii de proximitate și integritate (aplicații de securitate).

Soluția propusă se bazează pe placa de dezvoltare Adafruit Feather 32U4 FONA (1), placă ce combină un microcontroler ATmega32U4 la 8MHz / 3.3V și un controler GSM SIM800 quad-band (850/900/1800/1900MHz). Placa necesită alimentarea de la un acumulator LiPo extern de minim 500mAh (se pot utiliza (2), (3), (4)) și o antenă GSM uFL (5).

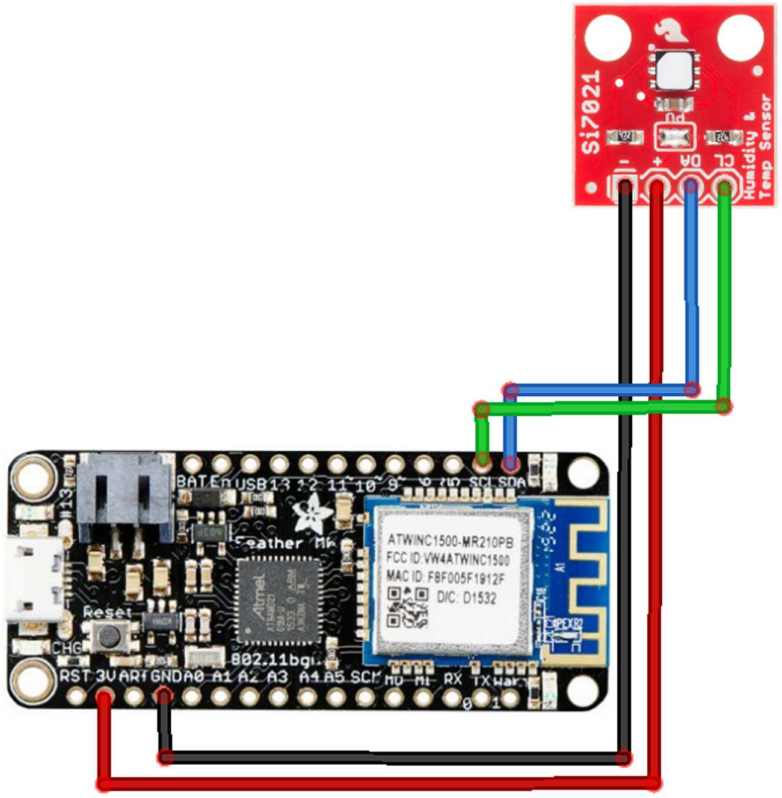


Pentru achiziția parametrilor temperatură și umiditate vom utiliza un senzor digital I2C Si7021 (6) ce are o precizie mare de măsurare și se poate interfața foarte ușor cu placa de dezvoltare utilizată.

Funcționarea sistemului necesită o cartelă GSM 2G cu capabilități de transfer de date. Pentru acest lucru vă propunem utilizarea unui SIM Simfony Mobile M2M (7) – cartelă GSM ce oferă exclusiv servicii mobile de date. Cartela este disponibilă gratuit prin comandă pe site-ul companiei Simfony Mobile SRL (8) sau împreună cu un produs din gama GSM pe site-ul Robofun (9). 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 indepența 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.

Interconectarea componentelor

Interconectarea dintre senzorul Si7021 și placa Feather 32U4 FONA este reprezentată în diagrama următoare:



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

Pentru mai multe detalii despre operarea plăcii Adafruit Feather 32U4 FONA se recomandă parcurgerea materialului ”Adafruit Feather 32u4 FONA - Take your Feather anywhere in the world” (10).

Programarea sistemului

Pentru realizarea și testarea sistemului s-a utilizat Arduino IDE 1.8.1 cu extensia Adafruit AVR Boards 1.4.9 instalată precum și bibliotecile Adafruit FONA 1.3.2, Adafruit Si7021 (11), Sleep_n0m1 1.1.1.

```
#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 <Sleep_n0m1.h>
```

```
Sleep sleep;

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

Dacă se dorește supravegherea funcționării sistemului pe serială se va adăuga următoarea linie, în funcționarea propriu-zisă nu este necesară:

```
#define debug
```

În cadrul secțiunii *setup()* se va realiza inițializarea modulului GSM și a senzorului Si7021:

```
void setup() {
  #ifdef debug
    while (!Serial);
    Serial.begin(115200);
    Serial.println("Initializing...");
  #endif

  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(apnpa
ssword));

  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

  sensor.begin();

  delay(5000);

}
```

Sistemul va raporta către serviciul IoT ThingSpeak (12) valorile achiziționate (temperatură și umiditate) precum și voltajul și procentul de încărcare a acumulatorului ce alimentează sistemul. Secțiunea *loop()* implementează atât partea de achiziție cât și partea de transmisie de rețea prin intermediul comunicației mobile. În cadrul codului trebuie personalizată valoarea parametrului *key* ce se obține în urma înregistrării gratuite pe site-ul ThingSpeak (12). Achiziția și raportarea se realizează la un interval de o oră (3600000ms), în intervalul de inactivitate atât modulul GSM cât și microcontrolerul 32u4 se află în mod de consum redus.

```
void loop() {

  while (!fona.enableGPRS(true)) {
    #ifdef debug
      Serial.println(F("Failed to turn on GPRS"));
    #endif
    delay(5000);
  }
```

```
  uint16_t vbat;
  uint16_t pbat;

#ifdef debug
  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("-------------------------------------
---------------");
#else
  fona.getBattVoltage(&vbat);
  fona.getBattPercent(&pbat);
#endif

  float temperature, humidity;

  temperature = sensor.readTemperature();
  humidity = sensor.readHumidity();

  #ifdef debug
    Serial.print("Humidity:    ");
    Serial.print(humidity, 2);
    Serial.print("\tTemperature: ");
    Serial.println(temperature, 2);
  #endif

  String temp = "api.thingspeak.com/update?key=…&field1="
+ String(vbat/1000.0F,2) + "&field2=" + String(pbat) +
"&field3=" + String(temperature,2) + "&field4=" +
String(humidity,2);

  uint16_t statuscode;
  int16_t length;
```

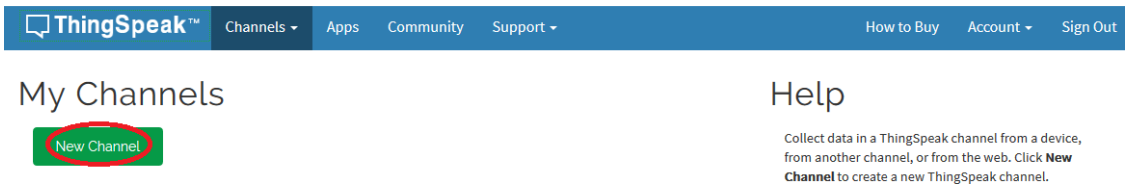
```
  char url[100];
  temp.toCharArray(url,temp.length()+1);

  #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);
  #ifdef debug
    delay(3600000);
  #else
    sleep.pwrDownMode();
    sleep.sleepDelay(3600000);
  #endif
  digitalWrite(FONA_DTR,LOW);
  delay(100);
}
```

Configurarea sistemului de raportare IoT

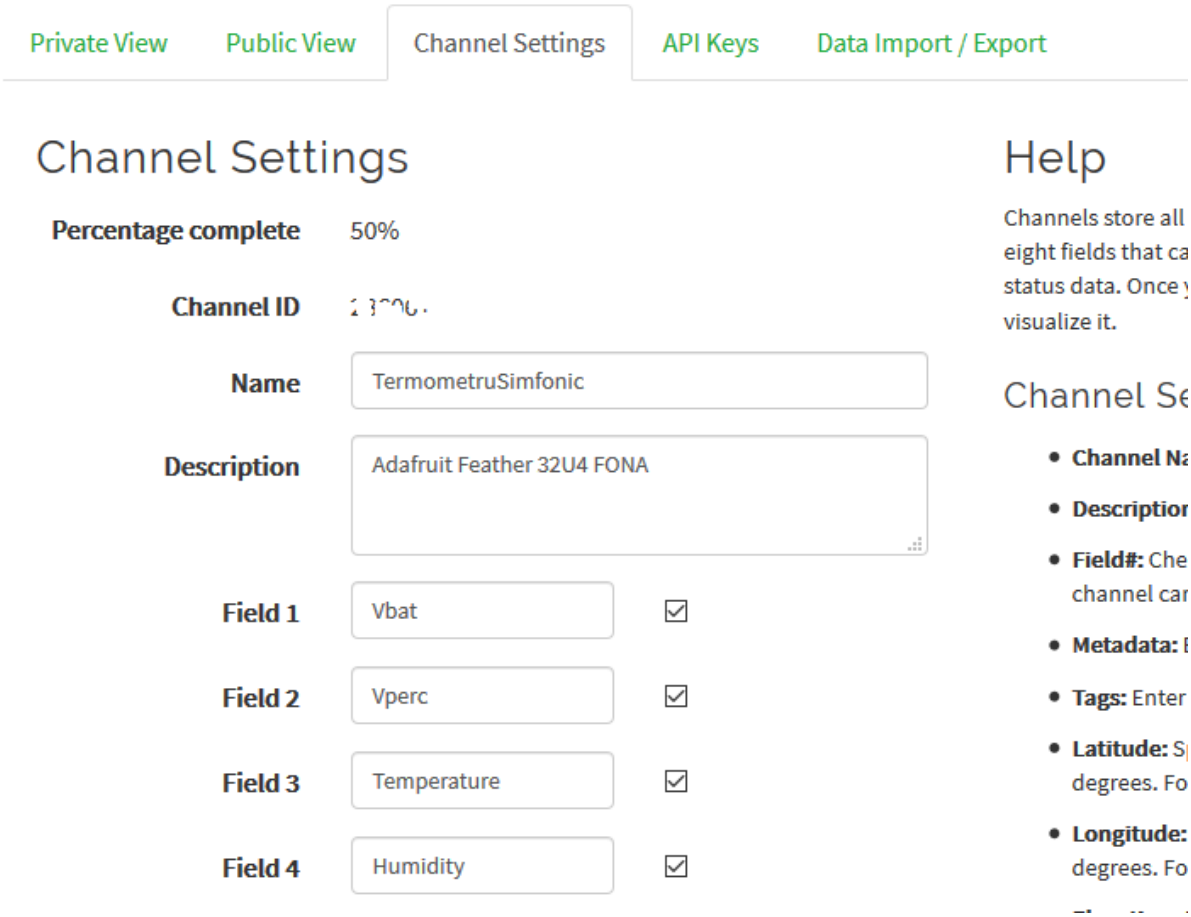
Sistemul nostru este un sistem tipic IoT ce utilizează platforma on-line ThingSpeak (12). Utilizarea platformei necesită înregistrare dar aceasta este gratuită. Platforma ThingSpeak este una dintre cele mai cunoscute platforme IoT ce oferă servicii de stocare, prelucrare și vizualizare a datelor. Unul dintre avantajele majore ale platformei este posibilitatea de execuție de programe scrise în limbajul Matlab (13).

După înregistrare se va defini un nou canal înregistrare a datelor (*My Channels* / *New Channel*). Definirea canalului va genera și cheia (*Write API Key*) ce trebuie utilizată în program.

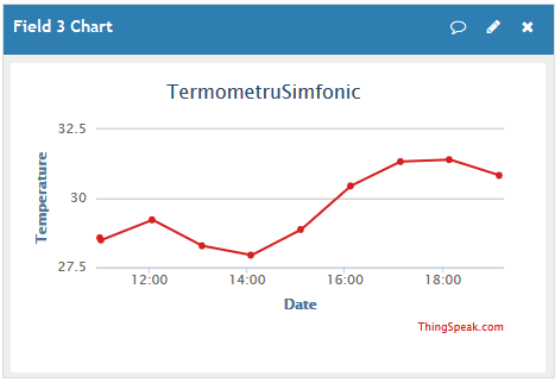


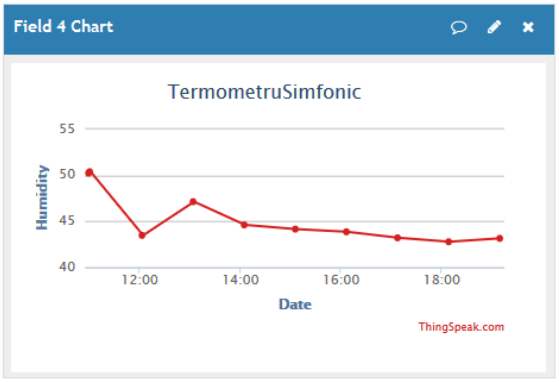
https://www.robofun.ro/forum/

În cadrul acestui canal vom defini patru câmpuri Vbat, Vperc, Temperature și Humidity ce vor stoca efectiv datele trimise de dispozitivul nostru.



După punerea în funcțiune a sistemului se vor putea vedea și datele trimise de dispozitiv (secțiunea *PrivateView*).





Referințe on-line

(1) Feather 32u4 FONA

https://www.robofun.ro/wireless/wireless-gsm/feather-32u4-fona?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(2) Acumulator LiPo - 3.7v 800mAh

https://www.robofun.ro/surse_de_alimentare/acumulatori/Acumulator-LiPo-3.7v-800mAh?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(3) Acumulator LIPo Lithium Polymer - 3.7v 1200mAh

https://www.robofun.ro/surse_de_alimentare/acumulatori/lithium-ion-polymer-battery-3-7v-1200mah?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(4) Acumulator LIPO 3.7 V 1400 mA

https://www.robofun.ro/surse_de_alimentare/acumulatori/lipo-3.7v-1400mA?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(5) Antena GSM Quad-Band - 3dBi uFL

https://www.robofun.ro/slim-sticker-type-gsm-cellular-quad-band-antenna-3dbi-ufl?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(6) 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

(7) 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

(8) M2M Global Connectivity, Internet of Things Platform | Simfony Mobile
http://simfonymobile.com/

(9) Produse GSM

https://www.robofun.ro/wireless/wireless-gsm?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(10) Adafruit Feather 32u4 FONA - Take your Feather anywhere in the world
https://learn.adafruit.com/adafruit-feather-32u4-fona

(11) GitHub - adafruit/Adafruit_Si7021: Arduino library for Adafruit Si7021
https://github.com/adafruit/Adafruit_Si7021

(12) IoT Analytics – ThingSpeak
https://thingspeak.com/

(13) MATLAB – MathWorks
https://www.mathworks.com/products/matlab.html