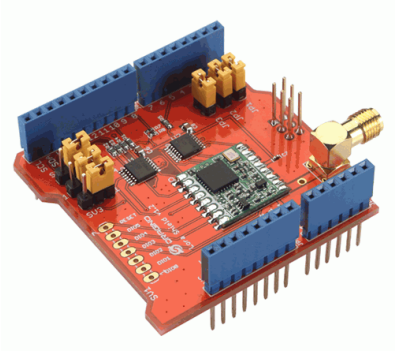
# Cum să realizăm un sistem IoT LoRaWAN

În lecția precedentă (LoRa meets Robofun IoT) am văzut cum putem realiza un sistem IoT utilizând comunicația radio LoRa (1). Utilizând module radio LoRa putem transmite date la mare distanță dar pentru implementarea unui sistem IoT este necesară implementarea atât a modulului de achiziție (sau acționare) cât și a sistemului de tip gateway ce face legătura cu rețeaua Internet și cu serviciile cloud IoT. Specificațiile LoRaWAN (2) permit implementare unor rețele radio LoRa standardizate astfel încât sistemele gateway să permită conectarea dispozitivelor IoT după un set de reguli larg acceptate. Realizarea unui sistem IoT LoRaWAN presupune realizare unui sistem de achiziție / acționare care respectă acest set de reguli și se conectează la o infrastructură de gateway-uri deja extistentă (nu mai este nevoie să realizăm și să operăm sistemul gateway). Există mai multe rețele de gateway-uri LoRaWAN dar în cadrul acestei lecții vom arăta cum putem realiza un sistem ce folosește rețeaua TTN (The Things Network (3)). Accesul în rețeaua TTN este gratuit deoarece se bazează pe gateway-uri particulare partajate între utilizatorii rețelei. Tot ce trebuie să faceți este să verificați dacă vă aflați în aria de acoperire a unui sistem gateway TTN.

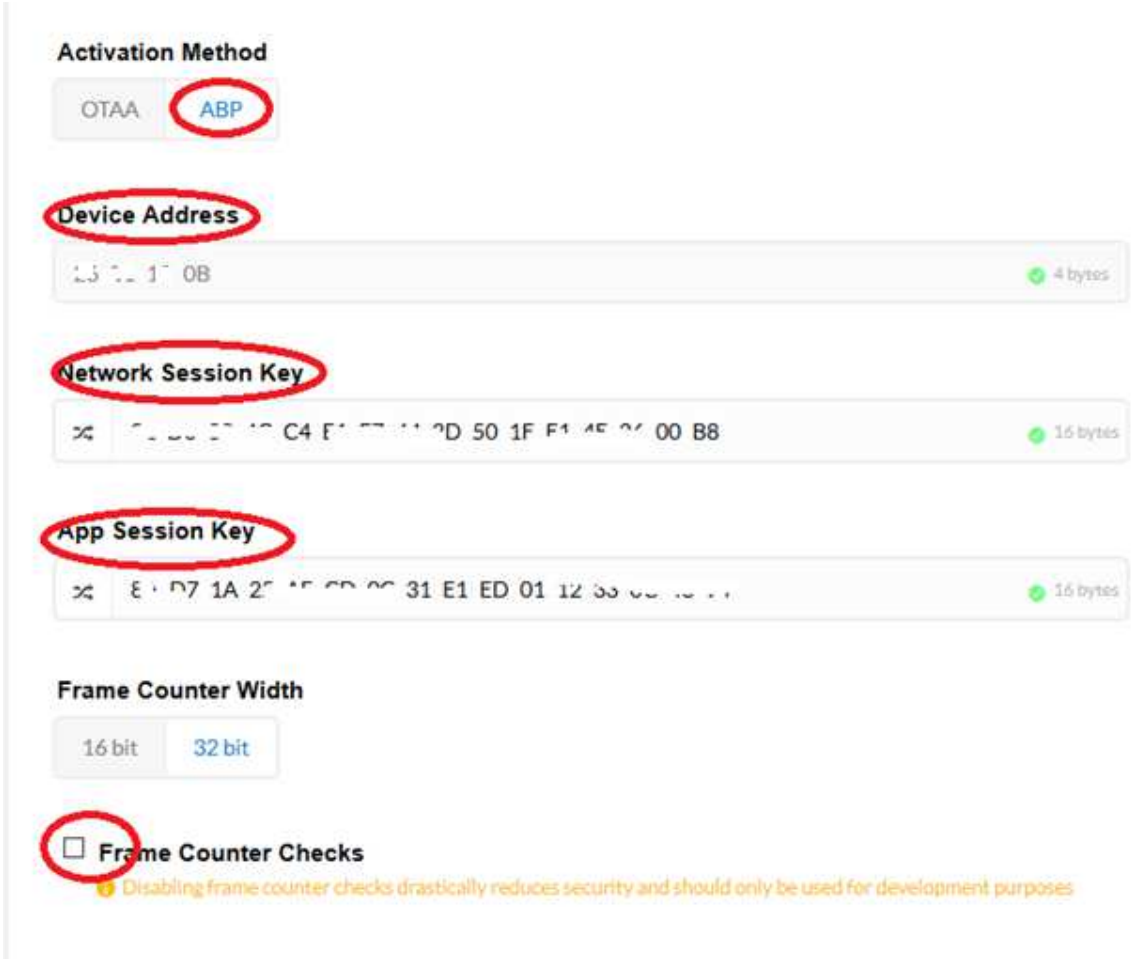
Pentru sistemul IoT vom utiliza o placă de dezvoltare Arduino Uno (4) și un shield Dragino LoRa (5) echipat cu un modul radio LoRa în frecvență de 868MHz. Pentru partea de achiziție vom exemplifica măsurarea temperaturii utilizând un senzor brick (6) conectat pe pinul anologic A0 al plăcii de dezvoltare.



Pentru implementarea comunicației LoRaWAN vom utiliza biblioteca Arduino-LMIC (7). Testele au fost realizate utilizând Arduino IDE 1.8.3 și versiunea 1.5.0+arduino-1 a bibliotecii. Programul pleacă de la exemplul *ttn-abp* al bibliotecii în care vom efectua o serie de mici modificări. În primul rând trebuie să înregistrăm sistemul pe platforma TTN pentru a obține datele de autentificare în rețea:

```
static const PROGMEM u1_t NWKSKEY[16] = { … };
static const u1_t PROGMEM APPSKEY[16] = { … };
static const u4_t DEVADDR = … ;
```

Înregistrarea presupune crearea unui cont de utilizator, definirea unei aplicații (*Applications*) și, în cadrul aplicației, definirea unui dispozitiv (*Device*). În secțiunea se setări (*Settings*) a noului dispozitiv trebuie aleasă metoda *ABP* de activare și debifată opțiunea *Frame Counter Checks*. Tot în cadrul acestei secțiuni se regăsesc datele de autentificare în rețeua TTN. Pentru mai multe detalii legate de definirea aplicației și dispozitivului în rețeua TTN se poate consulta și materialul „LoRaWAN IoT with Arduino Uno, Dragino v1.3 & TheThingsNetwork” (8).




Tot în secțiunea de inițializare a exemplului se va șterge declarația mesajului *mydata* (se va defini din nou în program sub o altă formă) și se va modifica intervalul de postare a mesajelor (postarea la 1 minut este destul de agresivă pentru politica de utilizare a rețelei TTN (9)).

```
const unsigned TX_INTERVAL = 3600;
```

Shield-ul Dragino LoRa necesită următoarea modificare în structura de definire a

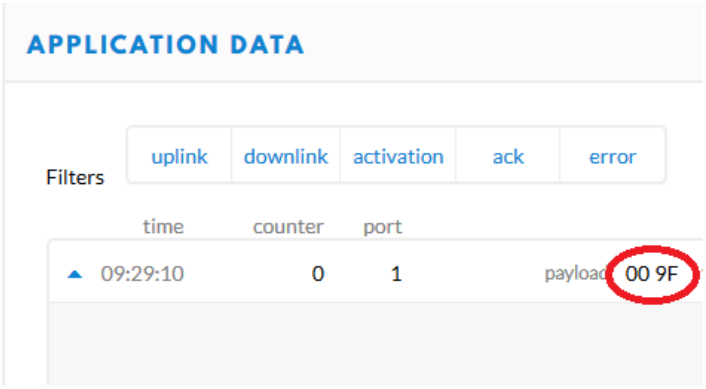
pinilor utilizați:

```
const lmic_pinmap lmic_pins = {
    .nss = 10,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = 9,
    .dio = {2, 6, 7},
};
```

Ultima modificare adusă exemplului *ttn-abp* este rescrierea procedurii *do_send* pentru a trasmite valoare achiziționată de la brick-ul de temperatură în locul mesajului text predefinit. După cum se poate observa se va transmite valoarea returnată de funcția analogRead, prelucrarea numerică pentru a obține valoarea temperaturii se va face în sistemul cloud TTN.

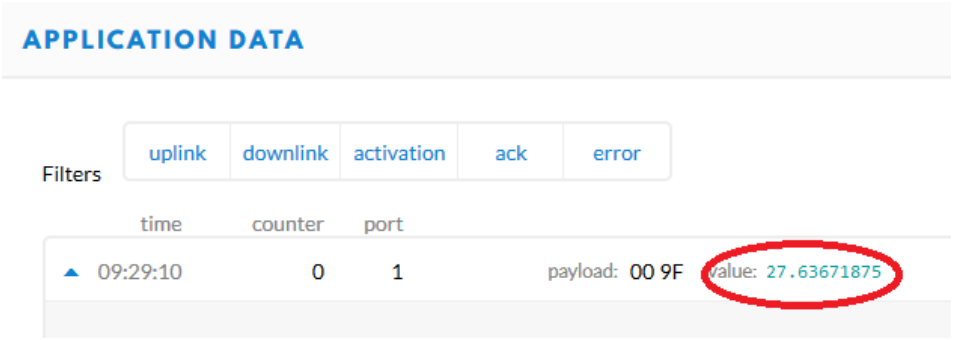
```
void do_send(osjob_t* j){
    static uint8_t mydata[2];
    int reading = analogRead(A0);
    mydata[0] = highByte(reading);
    mydata[1] = lowByte(reading);
    if (LMIC.opmode & OP_TXRXPEND) {
        Serial.println(F("OP_TXRXPEND, not sending"));
    } else {
        LMIC_setTxData2(1, mydata, sizeof(mydata), 0);
        Serial.println(F("Packet queued"));
    }
}
```

După punerea în funcțiune a sistemului, și dacă vă aflați în aria de acoperire a unui gateway TTN, în consola TTN vor începe să apară valorile transmise de acesta (secțiunea *Application Data*). După cum se poate observa datele transmise sunt sub forma unui șir de valori în hexazecimal (2 octeți – 16 biți).




Pentru a transforma datele primite într-o formă mai ușor de înțeles se va scrie o funcție de decodare (în secțiunea *Payload Formats* / *decoder*). Această funcție va avea și rolul de a calcula temperatura echivalentă valorii achiziționate. După implementarea acestei funcții vom putea vedea în secțiunea de *Application Data* valoarea efectivă a temperaturii.

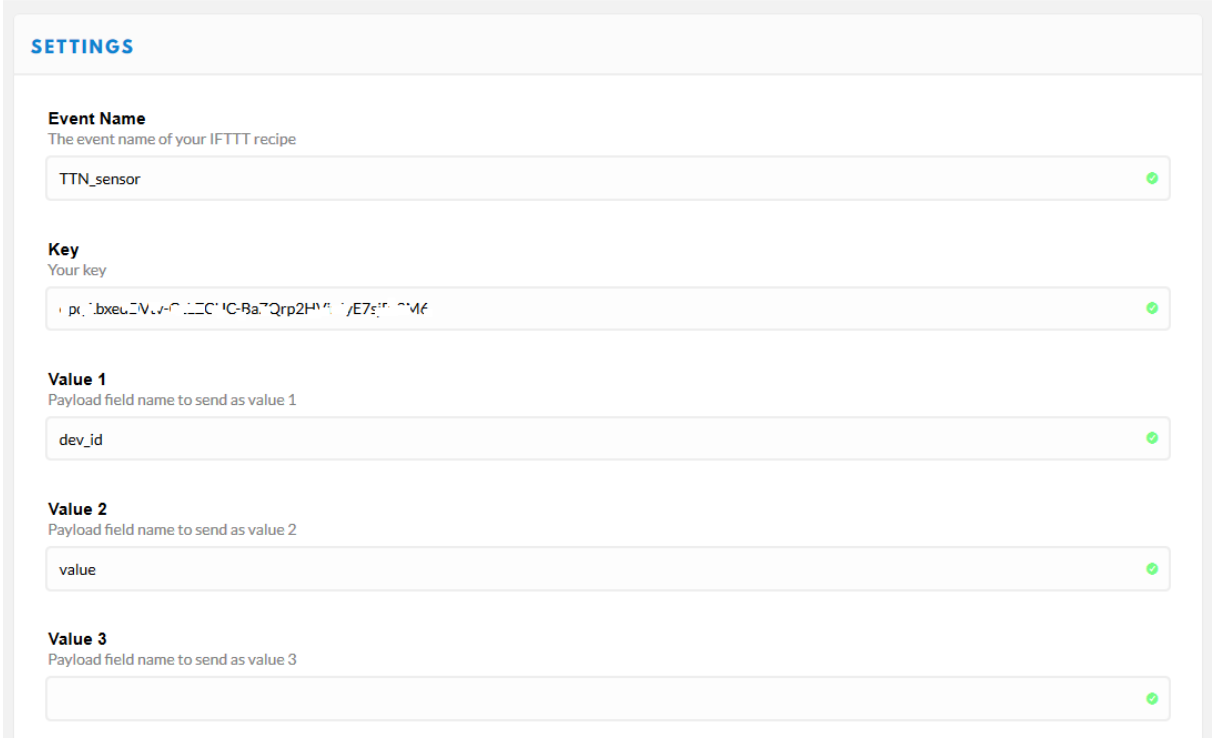
```
function Decoder(bytes, port) {
  var decoded = ((((bytes[0]<<8)|bytes[1])*5.0)/1024.0)-0.5)*100;
  return { value:decoded };
}
```



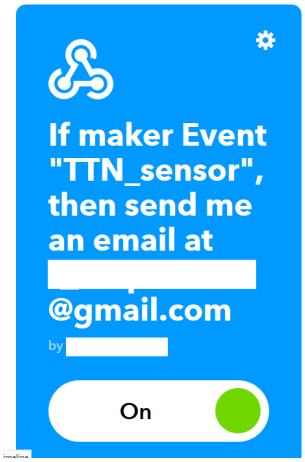

Atenție!!! Platforma TTN nu este o platformă IoT – nu stochează datele preluate de la sistemele LoRaWAN. Datele se pot observa în consolă doar dacă sunt transmise atunci când consola este deschisă. Platforma TTN permite în schimb transmiterea datelor primite prin rețeaua LoRaWAN către alte platforme online inclusiv platforme IoT. În secțiunea *Integrations* se pot defini diverse mecanisme automate de redirectare a datelor către sisteme precum Cayenne, OpenSensors sau IFTTT. Vom explica în cele ce urmează cum putem transmite datele către serviciul IFTTT (10) care ne va trimite apoi valoarea temperaturii prin email. Bineînțeles, multitudinea de opțiuni oferite de platforma IFTTT permite redirectarea datelor către un serviciu IoT, postarea pe o rețea

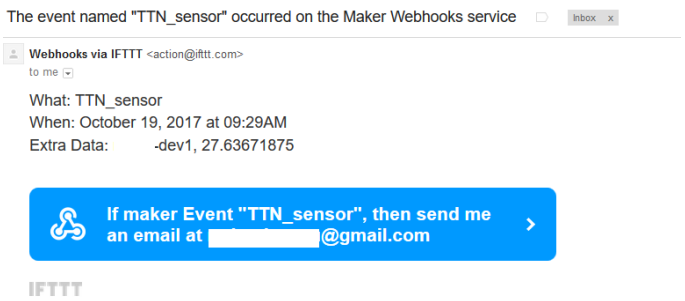
de socializare sau interacțiunea directă cu alte dispozitive IoT.

Definirea mecanismului automat de trimitere a datelor către serviciului IFTTT presupune adăugarea unui integrator de tipul *IFTTT Maker* (*add integration* / secțiunea *Integrations*). Conexiunea între cele două servicii (TTN și IFTTT) se realizează pe baza *Event Name* (trebuie să fie identic cu numele declanșatorului IFTTT) și *Key* (cheie de autentificare oferită de obiectul IFTTT Webhooks).




În cadrul platformei IFTTT se va realiza o regulă ce va avea declanșator serviciul Webhooks (*Event Name* trebuie să fie identic cu cel definit în platforma TTN) și ca efect transmiterea unui email. La fiecare valoare a temperaturii transmisă de sistemul nostru vom primi un email de forma:




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

Pentru mai multe variante de realizare a unui sistem IoT LoRaWAN se pot consulta și următoarele materiale:

- Low Cost LoRaWan nodes with Arduino and Aliexpress RF96 modules (11)
- This is the LoRa-Arduino quickstart project by the IoT Studio (12)
- Getting Started With LoraWAN: Build a LoRa Node With Arduino and RN2483 (13)
- LPWAN – Starting up with LoraWAN and The Things Network (14)
- Build Lora node using Arduino Uno and HopeRF RFM95 LoRa transceiver module (15)

# Referințe on-line


(1) What Is LoRa?

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

(2) lora-alliance | WHAT IS LoRaWAN?

https://www.lora-alliance.org/what-is-lora

(3) The Things Network

https://www.thethingsnetwork.org/

(4) Arduino Uno v3

https://www.robofun.ro/arduino/arduino_uno_v3?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(5) Dragino LoRa Shield

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

(6) Senzor Temperatura Brick

https://www.robofun.ro/senzori/vreme/senzor-temperatura-brick?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(7) matthijskooijman/arduino-lmic: LoraWAN-in-C library, adapted to run under the Arduino environment

https://github.com/matthijskooijman/arduino-lmic

(8) LoRaWAN IoT with Arduino Uno, Dragino v1.3 & TheThingsNetwork

http://www.tamberg.org/chopen/2016/LoRaWANIoTWorkshop.pdf

(9) Limitations: data rate, packet size, 30 seconds uplink and 10 messages downlink per day Fair Access Policy

https://www.thethingsnetwork.org/forum/t/limitations-data-rate-packet-size-30-seconds-uplink-and-10-messages-downlink-per-day-fair-access-policy/1300

(10) Learn how IFTTT works - IFTTT

https://ifttt.com/

(11) Low Cost LoRaWan nodes with Arduino and Aliexpress RF96 modules

https://www.thethingsnetwork.org/community/gold-coast/post/low-cost-lorawan-nodes-with-arduino-and-aliexpress-rf96-modules

(12) This is the LoRa-Arduino quickstart project by the IoT Studio

https://github.com/mikimer/LoRa

(13) Getting Started With LoraWAN: Build a LoRa Node With Arduino and RN2483

http://www.instructables.com/id/Build-a-LoRa-Node-With-Arduino-and-RN2483/

(14) LPWAN – Starting up with LoraWAN and The Things Network

https://primalcortex.wordpress.com/2016/11/25/lpwan-starting-up-with-lorawan-and-the-things-network/

(15) Build Lora node using Arduino Uno and HopeRF RFM95 LoRa transceiver module.

https://www.mobilefish.com/developer/lorawan/lorawan_quickguide_build_lora_node_rfm95_arduino_uno.html