

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).

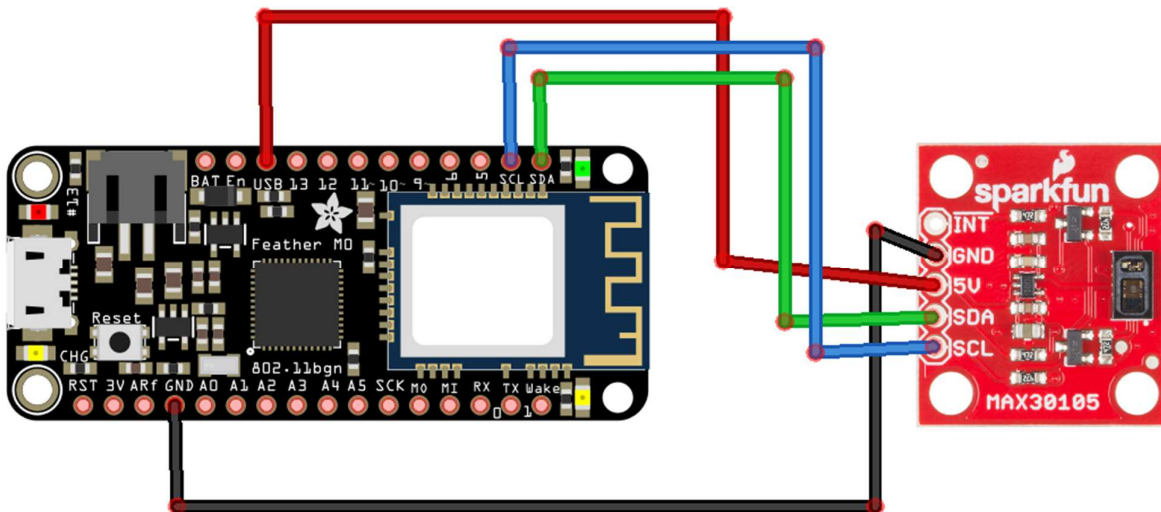
# My Heart Will Go IoT

Măsurarea pulsului, a numărului de bătăi ale inimii pe minut, este un instrument de evaluare atât a stării de sănătate dar și a stării emoționale sau a efortului fizic depus. Apariția de sensori optici performanți fac această evaluare foarte simplă permițând măsurarea pulsului cu telefonul mobil sau cu diverse dispozitive de tip brățară. În cadrul lecției de față vom explica construirea unui astfel de dispozitiv dar, în plus față de dispozitivele clasice de măsurare a pulsului, vom înregistra datele obținute în cloud.



Atenție!!! Dispozitivul prezentat nu este un dispozitiv medical! Dispozitivul din această lecție nu poate fi folosit pentru evaluarea stării de sănătate! Lecția de față prezintă un principiu de măsurare și înregistrare nu și o metodă exactă de calibrare a datelor măsurate.

Pentru implementarea dispozitivului vom utiliza o placă de dezvoltare Adafruit Feather M0 WiFi (1) și un senzor de particule și puls MAX30105 (2). Schema de interconectare dintre cele două componente este următoarea:



Senzorul comunică cu placa de dezvoltare prin intermediul magistralei I2C și atunci conexiunile dintre cele două componente sunt evidente: pinul SCL al senzorului la pinul 4 (SCL) al plăcii de dezvoltare și pinul SDA al senzorului la pinul 3 (SDA) al plăcii de dezvoltare. Pinul /INT al senzorului nu este folosit. Alimentarea senzorului se va face la 5V – pinul 5V se va conecta la pinul USB al plăcii de dezvoltare atâta timp cât placa se alimentează prin intermediul cablului USB. Dacă doriți să transformați sistemul într-un sistem portabil, alimentând placa de la un acumulator LiPo, atunci pinul de 5V al senzorului se va conecta la pinul BAT al plăcii de dezvoltare – senzorul se va alimenta la 3.7V (producătorul specifică posibilitatea de a alimenta senzorul la tensiuni între 3.3V și 5V dar explică că o tensiune de peste 3.5V va asigura o funcționare mai bună a senzorului).

Pentru punerea în funcțiune și pentru integrarea plăcii de dezvoltare cu mediul Arduino IDE este necesară parcurgerea materialului „Adafruit Feather M0 WiFi with ATWINC1500 – ARM Cortex M0+ with fast & fun wireless built in” (3). Pentru înțelegerea mai bună a modului de funcționare a senzorului de puls se poate parcurge materialul „MAX30105 Particle and Pulse Ox Sensor Hookup Guide” (4).

Programul sistemului se bazează pe exemplul *Example5\_HeartRate* al bibliotecii software *Sparkfun MAX3010x Pulse and Proximity Sensor Library* (5). Programul a fost testat utilizând Arduino IDE 1.8.3 cu extensia Adafruit SAMD Boards 1.0.17 instalată și bibliotecile WiFi101 0.14.3 și Sparkfun MAX3010x 1.0.0. Atenție!!! Pentru a compila programul pentru placa Feather M0 este necesară adăugarea unei linii (*#define BUFFER\_LENGTH 32*) în fișierul *MAX30105.cpp* al bibliotecii *Sparkfun MAX3010x*.

```
#include <SPI.h>
#include <WiFi101.h>

#define WINC_CS    8
#define WINC_IRQ   7
#define WINC_RST   4
#define WINC_EN    2
```

Datele de conectare WiFi trebuie personalizate în program. Programul va înregistra datele măsurate în cloud utilizând serviciul Robofun IoT (6).

```
char ssid[] = "...";
char pass[] = "...";
```

```
WiFiClient client;
int status = WL_IDLE_STATUS;
char server[] = "iot.robofun.ro";
```

Constanta *postingInterval* va dicta intervalul (în milisecunde) la care se face înregistrarea datelor în cloud.

```
unsigned long lastConnectionTime = 0;
const unsigned long postingInterval = 30L * 1000L;

#include <Wire.h>
#include "MAX30105.h"

#include "heartRate.h"

MAX30105 particleSensor;

const byte RATE_SIZE = 4;
byte rates[RATE_SIZE];
byte rateSpot = 0;
long lastBeat = 0;

float beatsPerMinute;
int beatAvg;
```

Decomentarea directivei *debug* va permite vizualizarea datelor măsurate în concola serială.

```
//#define debug
```



În cadrul secțiunii *setup()* se va inițializa conexiunea WiFi și comunicația cu senzorul de puls. Mesajele de eroare vor apărea în consola serială doar dacă este activată directiva *debug*.

```
void setup()
{
    #ifdef debug
        Serial.begin(115200);
        while (!Serial) { ; }
        Serial.println("Initializing...");
    #endif
    WiFi.setPins( 8,7,4,2);

    if (WiFi.status() == WL_NO_SHIELD) {
        #ifdef debug
            Serial.println("WiFi shield not present");
        #endif
        while (true);
    }

    while (status != WL_CONNECTED) {
        #ifdef debug
            Serial.print("Attempting to connect to SSID: ");
            Serial.println(ssid);
        #endif
        status = WiFi.begin(ssid, pass);
        delay(10000);
    }
    #ifdef debug
        Serial.println("Connected to wifi");
    #endif

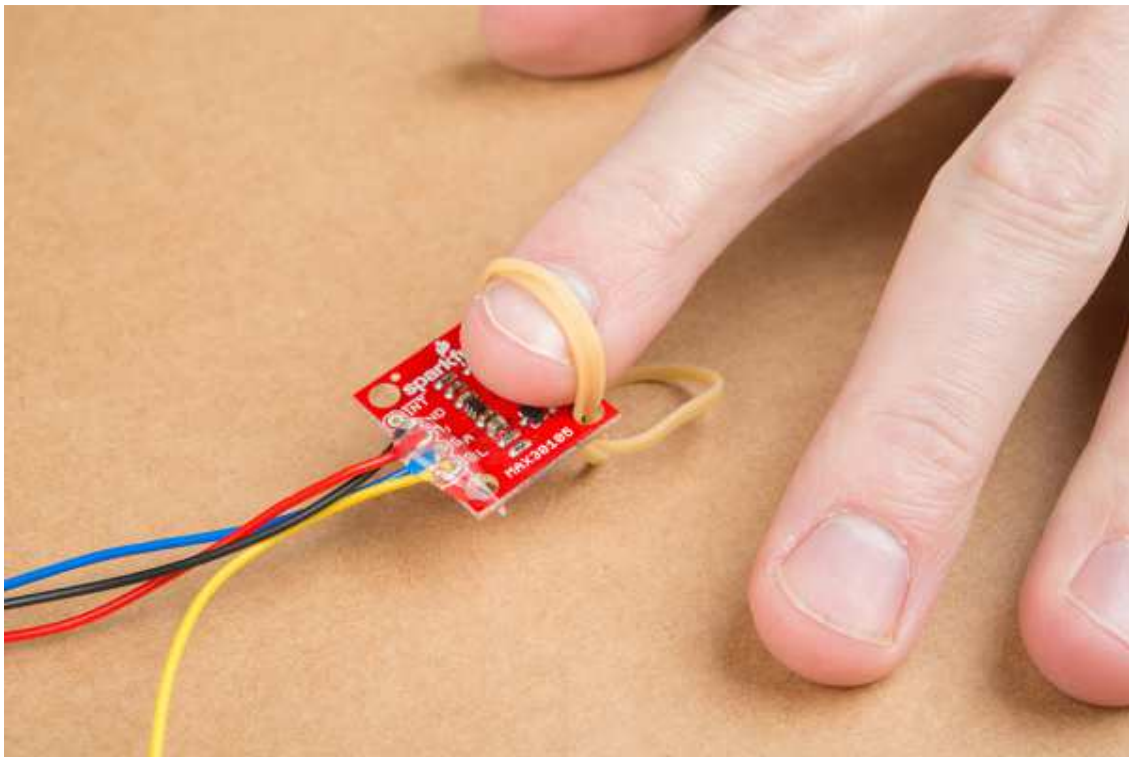
    if (!particleSensor.begin(Wire, I2C_SPEED_FAST))
    {
        #ifdef debug
            Serial.println("MAX30105 was not found. Please check
                            wiring/power. ");
        #endif
        while (1);
    }
    #ifdef debug
        Serial.println("Place your index finger on the sensor
with steady pressure.");
    #endif
}
```

```

#endif
particleSensor.setup();
particleSensor.setPulseAmplitudeRed(0x0A);
particleSensor.setPulseAmplitudeGreen(0);
}

```

În cadrul secțiunii `loop()` se va efectua atât citirea senzorului de puls (BPM – bătăi pe minut, medie – Avg BPM) cât și postarea către serviciul Robofun IoT (6). În cadrul codului trebuie personalizată cheia de autentificare (*TOKEN*) primită la înregistrarea senzorului în cadrul serviciului IoT. Intervalul de postare se contorizează începând cu detecția degetului pe senzor, dacă se ia degetul de pe senzor nu se mai efectuează înregistrarea valorilor în cloud. Se recomandă utilizarea unui elastic pentru a asigura o presiune constantă pe senzor.



```

void loop()
{
    long irValue = particleSensor.getIR();

    if (checkForBeat(irValue) == true)
    {
        long delta = millis() - lastBeat;
        lastBeat = millis();
    }
}

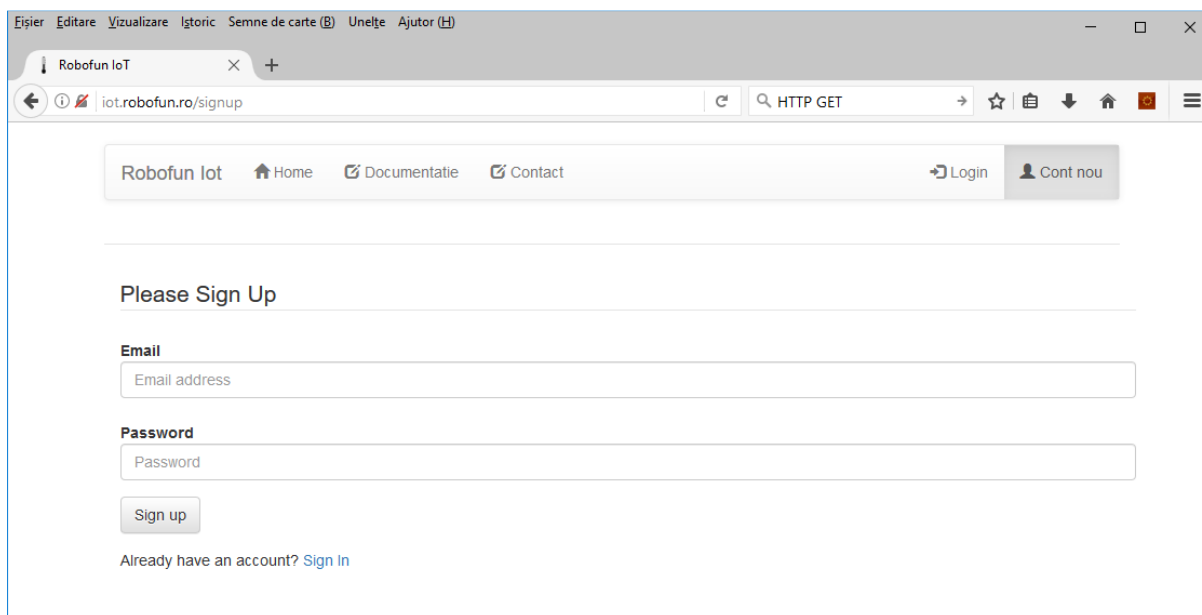
```

```

    beatsPerMinute = 60 / (delta / 1000.0);
    if (beatsPerMinute < 255 && beatsPerMinute > 20) {
        rates[rateSpot++] = (byte)beatsPerMinute;
        rateSpot %= RATE_SIZE;
        beatAvg = 0;
        for (byte x = 0 ; x < RATE_SIZE ; x++)
            beatAvg += rates[x];
        beatAvg /= RATE_SIZE; }
}
#ifdef debug
    Serial.print("IR=");
    Serial.print(irValue);
    Serial.print(", BPM=");
    Serial.print(beatsPerMinute);
    Serial.print(", Avg BPM=");
    Serial.print(beatAvg);
#endif
if (irValue < 50000) {
    #ifdef debug
        Serial.print(" No finger?");
    #endif
    lastConnectionTime = millis();
}
#ifdef debug
    Serial.println();
#endif
if ((millis() - lastConnectionTime > postingInterval) &&
    (beatAvg > 20))
{
    String temp = "GET /api/v1/input/TOKEN/" +
        String(beatAvg) + " HTTP/1.1";
    char param[100];
    temp.toCharArray(param,temp.length()+1);
    if (client.connect(server, 80)) {
        client.println(param);
        client.println("Host: iot.robofun.ro");
        client.println("Connection: close");
        client.println();
    }
    client.stop();
    lastConnectionTime = millis();
    lastBeat = 0;
}
}

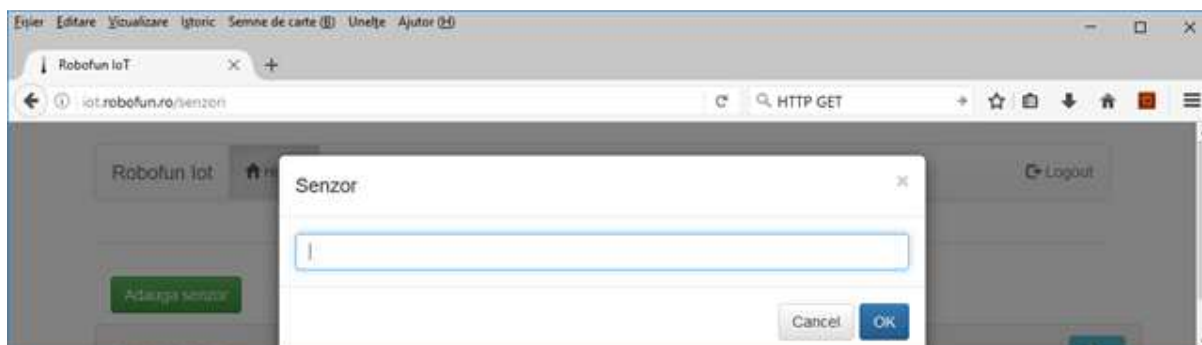
```

Pentru a utiliza serviciul Robofun IoT (6) este necesară înregistrarea gratuită.



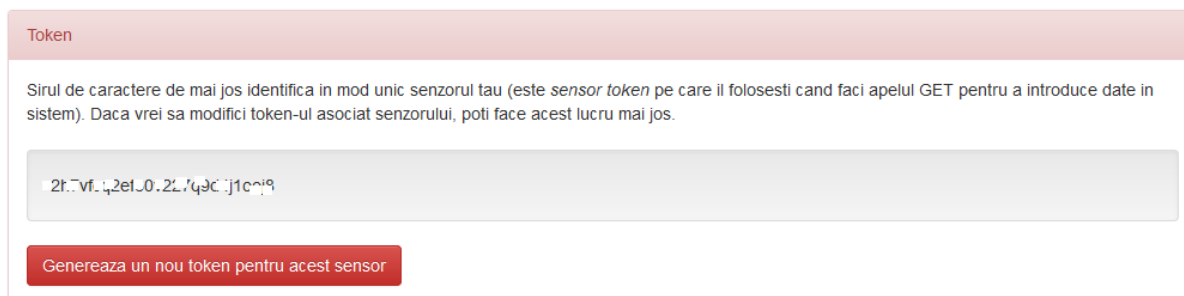
The screenshot shows a web browser window with the URL `iot.robofun.ro/signup`. The page has a navigation bar with links for Home, Documentatie, and Contact, along with Login and Cont nou buttons. The main content area is titled "Please Sign Up" and contains two input fields: "Email" (with placeholder "Email address") and "Password" (with placeholder "Password"). Below these fields is a "Sign up" button and a link that says "Already have an account? Sign In".

După înregistrare și conectare este necesară definirea unui noi senzor (*Adauga senzor*).



The screenshot shows a web browser window with the URL `iot.robofun.ro/senzori`. A modal dialog titled "Senzor" is open, featuring a single text input field and "Cancel" and "OK" buttons at the bottom. In the background, a green button labeled "Adauga senzor" is visible on the page.

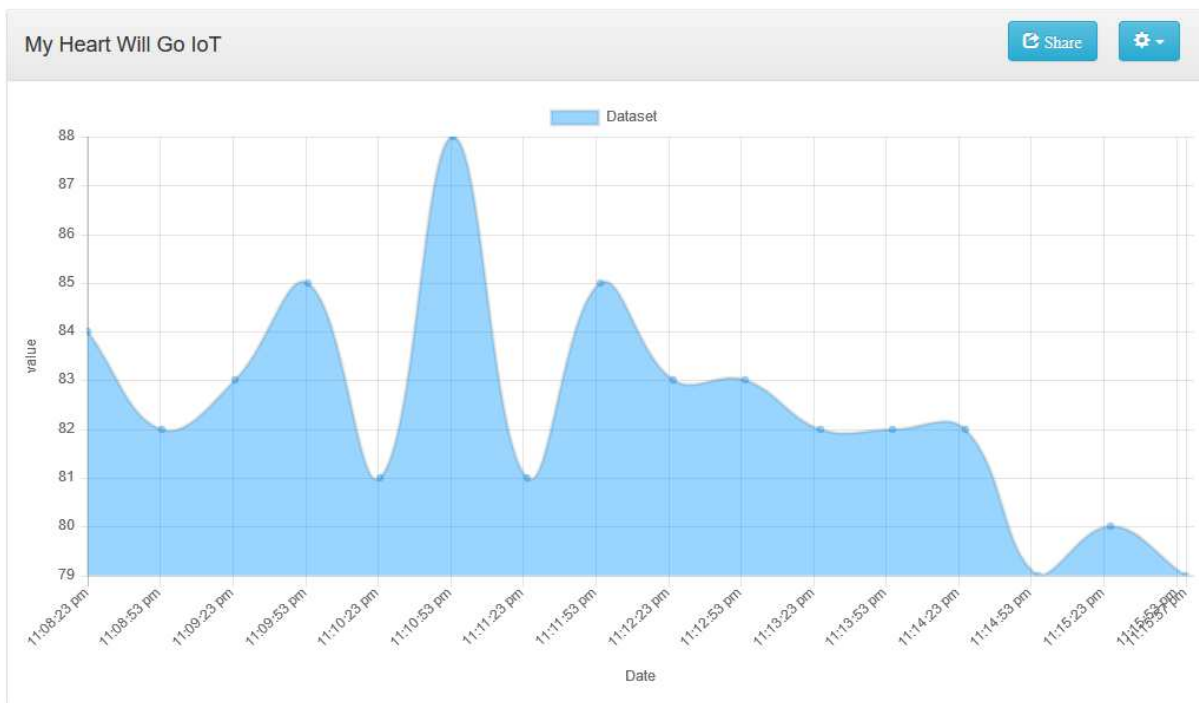
După definirea senzorului este necesar să copiem cheia de autentificare (*Token*) pentru a o utiliza în program.



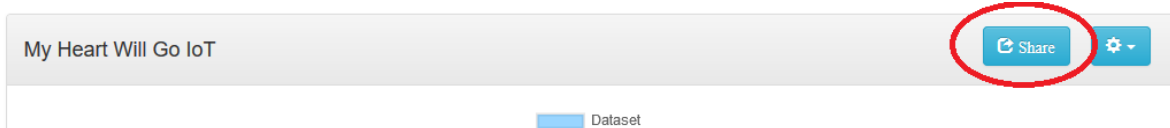
The screenshot shows a web page titled "Token". It contains a paragraph explaining that the token is a unique identifier for the sensor, used for GET requests to add data. Below the text is a text box displaying a token: `2h..vf..q2ef..0..2L..7q3C..j1c~9`. At the bottom, there is a red button labeled "Genereaza un nou token pentru acest senzor".



După configurarea serviciului IoT și încărcarea programului pe placa de dezvoltare putem începe utilizarea dispozitivului și putem observa datele înregistrate on-line:



O facilitate foarte interesantă a serviciului Robofun IoT este posibilitatea de partajare a graficelor (*Share sensor*) sau de integrare a acestora în alte pagini web. Putem astfel să adăugăm paginii web personale sau blog-ului personal un grafic live cu numărul de bătăi ale inimii noastre.



## Referințe on-line

(1) Feather M0 WiFi - ATSAM21 + ATWINC1500

[https://www.robofun.ro/wireless/wireless-wifi/feather-m0-wifi-atsamd21-atwinc1500&utm\\_medium=email&utm\\_content=productLink&utm\\_campaign=CURS\\_EMAIL](https://www.robofun.ro/wireless/wireless-wifi/feather-m0-wifi-atsamd21-atwinc1500&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL)

(2) Senzor de particule - MAX30105

[https://www.robofun.ro/sparkfun-particle-sensor-breakout-max30105?search=particule&utm\\_source=newsletter&utm\\_medium=email&utm\\_content=productLink&utm\\_campaign=CURS\\_EMAIL](https://www.robofun.ro/sparkfun-particle-sensor-breakout-max30105?search=particule&utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL)

(3) Adafruit Feather M0 WiFi with ATWINC1500 – ARM Cortex M0+ with fast & fun wireless built in

<https://learn.adafruit.com/adafruit-feather-m0-wifi-atwinc1500/>

(4) MAX30105 Particle and Pulse Ox Sensor Hookup Guide

<https://learn.sparkfun.com/tutorials/max30105-particle-and-pulse-ox-sensor-hookup-guide>

(5) An Arduino Library for the MAX3015 particle sensor and MAX30102 Pulse Ox sensor

[https://github.com/sparkfun/SparkFun\\_MAX3010x\\_Sensor\\_Library](https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library)

(6) Robofun IoT

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