

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs

CC BY-NC-ND



Codul sursa din acest document este licentiat

Public-Domain

Esti liber sa distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, printare, sau orice alt mijloc), atat timp cat nu aduci nici un fel de modificari acestuia. Codul sursa din acest document poate fi utilizat in orice fel de scop, de natura comerciala sau nu, fara nici un fel de limitari.

How sunny is the Blue?

(partea I)

În cadrul lecției de față vom prezenta implementarea unui sistem de tip IoT (Internet of Things) ce supraveghează radiațiile solare (intensitate luminoasă, factor radiații UV, temperatură ambientală generată) și utilizează serviciul cloud IBM BlueMix™. Spre deosebire de lecția precedentă (How cold is the Blue?) vom utiliza serviciul real, nu o aplicație demonstrativă. Serviciul IBM BlueMix™ este un serviciu comercial dar oferă posibilitatea de evaluare pentru o perioadă de 30 de zile sau, pentru studenții și profesorii din unele instituții de învățământ superior (de exemplu Universitatea Politehnica din București), accesul este gratuit.

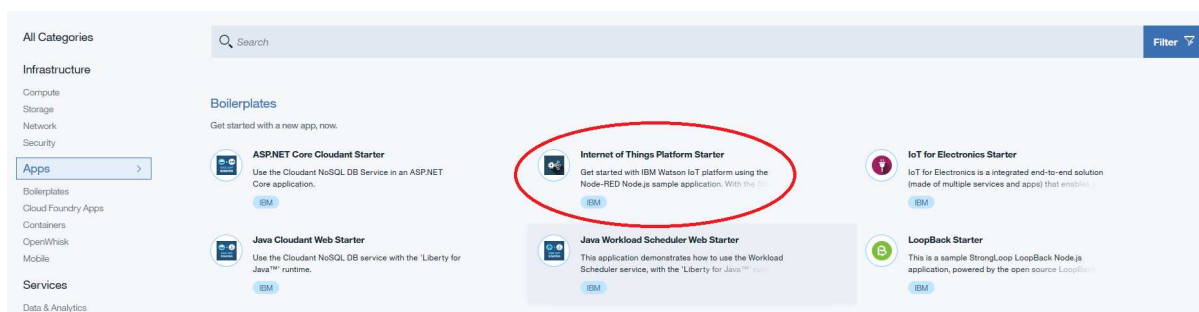


Platforma Cloud IBM BlueMix™ pune la dispoziția utilizatorilor un serviciu IoT specializat și anume IBM Watson IoT Platform. Accesul la acest serviciu face parte din serviciile puse la dispoziție de platforma cloud. Primul pas pentru accesarea acestui serviciu este înregistrarea unui cont trial – spre deosebire de alte servicii similare (Microsoft Azure, Google Cloud Platform sau Amazon Web Services) înregistrarea nu necesită introducerea de informații legate de un card bancar:

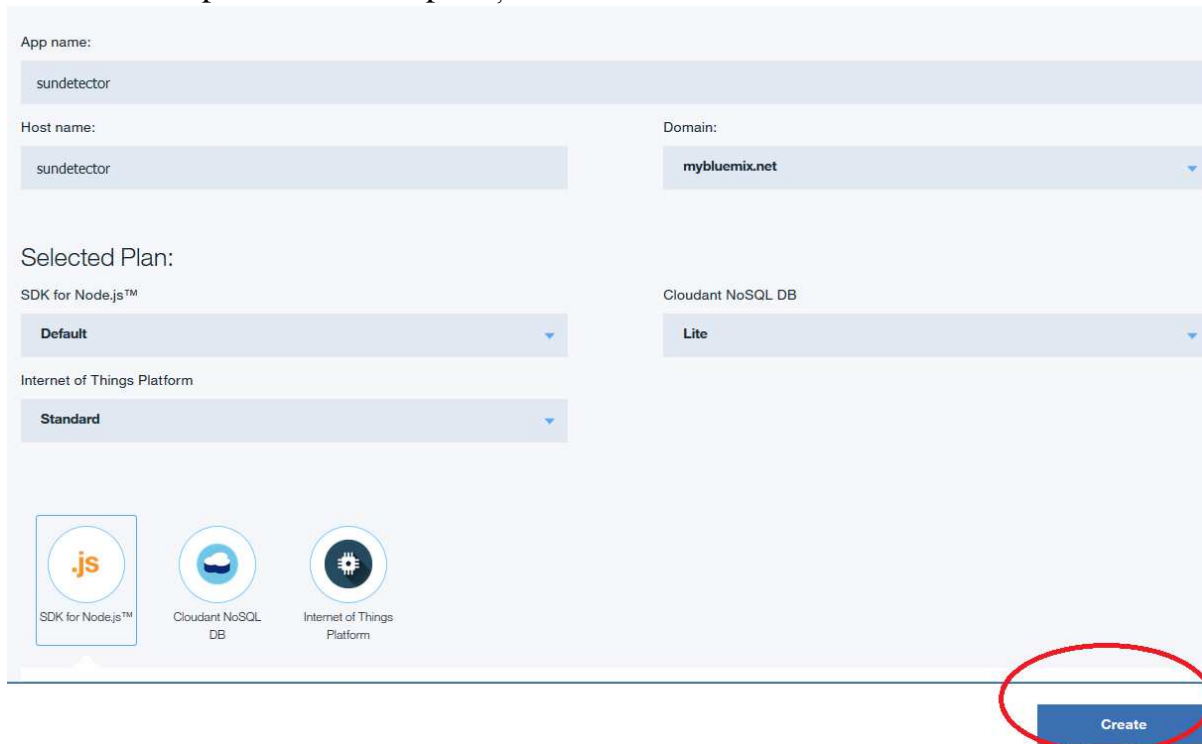
Sign up for IBM Bluemix

<https://console.ng.bluemix.net/registration/>

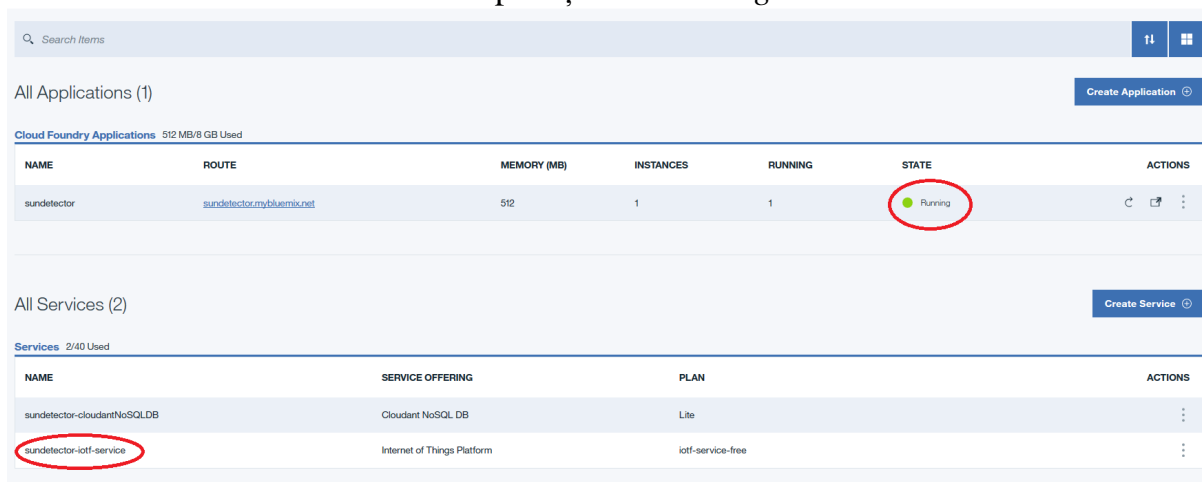
După înregistrare și conectare la platformă se va naviga în zona de aplicații și se va crea o aplicație nouă (*Create Application*) de tipul *Boilerplates / Internet of Things Platform Starter*:



Se va alege un nume pentru aplicație (în cadrul lecției se va utiliza numele *sundetector* dar numele aplicației trebuie să fie unic) și se va da comanda (*Create*) de alocare de resurse cloud pentru această aplicație nouă.



Alocarea de resurse poate dura câteva minute sau zeci de minute. Finalizarea operației este marcată de schimbarea stării aplicației în *Running*.



NAME	ROUTE	MEMORY (MB)	INSTANCES	RUNNING	STATE	ACTIONS
sundetector	sundetector.mybluemix.net	512	1	1	Running	

NAME	SERVICE OFFERING	PLAN	ACTIONS
sundetector-cloudbantNoSQLDB	Cloudbant NoSQL DB	Lite	
sundetector-iotf-service	Internet of Things Platform	iotf-service-free	

Pentru a putea conecta dispozitivul la aplicația nou creată trebuie să mergem în *Dashboard*-ul de administrare a serviciului *sundetector-iotf-service* (*Connect your devices / Launch dashboard*). Comanda ne va direcționa către portalul IBM Watson IoT Platform unde vom crea un nou dispozitiv (*Devices / Add Device*). Nu trebuie să completăm decât tipul dispozitivului (în exemplul nostru *Adafruit_Feather*) și numele dispozitivului (în exemplul nostru *WICED*). La finalizarea adăugării noului dispozitiv trebuie să notăm informațiile de autentificare furnizate de platformă pentru a putea să

le folosim în programul dispozitivului (*Organization ID*, *Device Type*, *Device ID*, *Authentication Method* și *Authentication Token*).

Este posibilă utilizarea platformei IBM Watson IoT Platform și fără autentificare utilizator (fără a înregistra un cont trial pe IBM BlueMix™), ca și *Demo User*, dar nu vom utiliza această variantă deoarece nu oferă nici un fel de securitate sau garanție de funcționare a serviciilor – această variantă este denumită *Playground* (loc de joacă). Dacă se dorește totuși efectuarea unor teste rapide adresa de conectare este:

Play with the IBM Watson IoT Platform dashboard

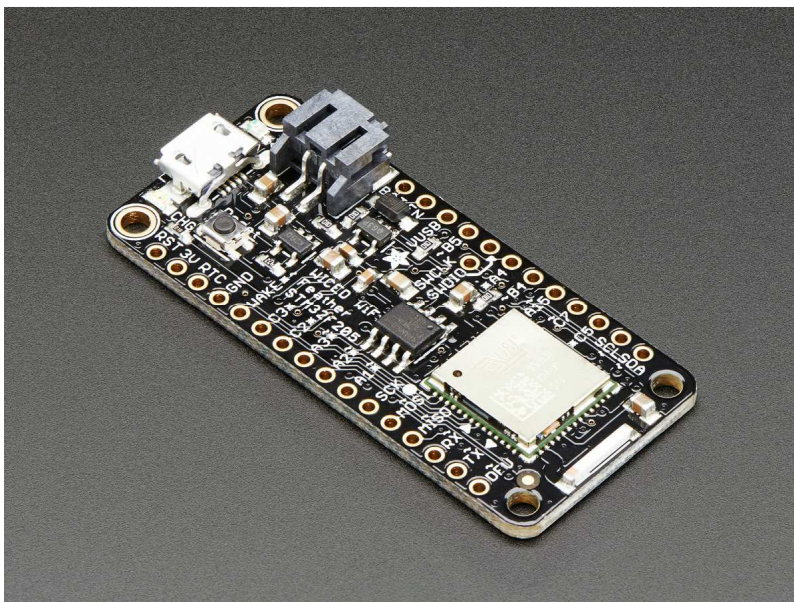
<https://play.internetofthings.ibmcloud.com/dashboard/#/boards/>

Pentru mai multe informații despre utilizarea și configurarea serviciului IBM Watson IoT Platform se poate consulta documentația oficială:

Getting started with Watson IoT Platform

<https://console.ng.bluemix.net/docs/services/IoT/index.html>

Pentru construirea dispozitivului IoT de supraveghere a radiațiilor solare vom utiliza o placă de dezvoltare Adafruit Feather WICED WiFi – cea mai nouă și mai puternică placă WiFi din gama Adafruit Feather ce combină un microcontroler STM32F205RG ARM Cortex M3 ce rulează la 120MHz cu modulul WiFi BCM43362 802.11b/G/N și care poate fi totuși programată cu mediul Arduino IDE.



<https://www.robofun.ro/wireless/wireless-wifi/adafruit-wiced-wifi-feather-stm32f205-with-broadcom-wiced-wifi>

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

Pentru mai multe informații despre instalarea (procedura de instalare sub mediul Arduino IDE necesită puțină mai multă atenție decât alte plăci Adafruit Feather) și utilizarea plăcii Adafruit Feather WICED WiFi se poate consulta:

Introducing the Adafruit WICED Feather WiFi | Adafruit Learning System

<https://learn.adafruit.com/introducing-the-adafruit-wiced-feather-wifi/>

Pentru partea de achiziție a parametrilor propuși vom utiliza doi senzori digitali (I2C):

- Senzorul de lumină și radiații UV Si1145



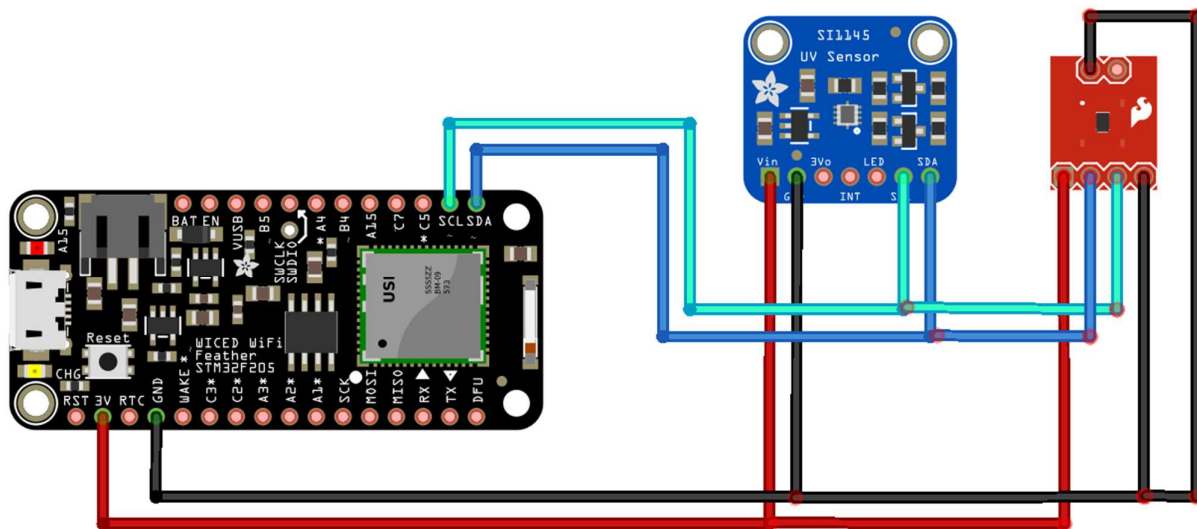
<https://www.robofun.ro/senzori/lumina/si1145-digital-uv-index-ir-sensor>

- Senzorul de temperatură TMP102



https://www.robofun.ro/senzori/vreme/senzor_temperatura_tmp102

Schema de interconectare între senzori și placa de dezvoltare este tipică pentru o magistrală multi-slave I2C:



Atenție!!! Conectarea pinului ADD0 la masă (GND), pentru senzorul TMP102, este necesară doar pentru varianta mai veche (cu pini pe două părți), varianta mai nouă (cu toți cei șase pini pe o singură parte) trage acest pin la masă prin intermediul unui contact intern.

Pentru implementarea software vom utiliza două biblioteci externe mediului Arduino IDE:

- Arduino MQTT – pentru implementarea protocolului MQTT
<https://github.com/256dpi/arduino-mqtt>
- Adafruit SI1145 – pentru utilizarea senzorului SI1145
https://github.com/adafruit/Adafruit_SI1145_Library

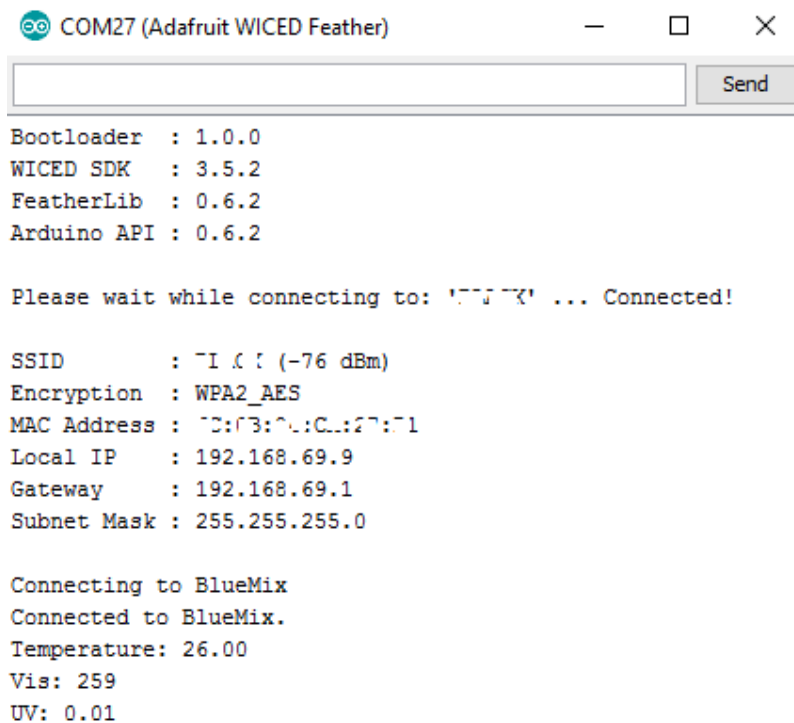
```
#include <adafruit_feather.h>
AdafruitTCP net;
#include <MQTTClient.h>
MQTTClient client;
#include <Wire.h>
#define tmp102Address 0x48
#include "Adafruit_SI1145.h"
Adafruit_SI1145 uv = Adafruit_SI1145();
```

```
#define WLAN_SSID          "....."
#define WLAN_PASS          "....."
```

Trebuie configurate valorile *WLAN_SSID* și *WLAN_PASS* în concordanță cu rețeaua WiFi prin care dispozitivul se va conecta la Internet.

În cadrul secțiunii *setup()* se va inițializa conexiunea cu platforma Watson. Procedura *client.begin* trebuie personalizată cu parametrul *Organization ID* iar procedura *client.connect* cu: *Organization ID*, *Device Type*, *Device ID* și *Authentication Token*.

O inițializare corectă a dispozitivului va afișa în consola serială ceva similar cu următoarea captură de ecran:



```
COM27 (Adafruit WICED Feather)
Send
Bootloader : 1.0.0
WICED SDK : 3.5.2
FeatherLib : 0.6.2
Arduino API : 0.6.2

Please wait while connecting to: '... Connected!

SSID : 'I ( (-76 dBm)
Encryption : WPA2_AES
MAC Address : '3:03:01:01:27:1
Local IP : 192.168.69.9
Gateway : 192.168.69.1
Subnet Mask : 255.255.255.0

Connecting to BlueMix
Connected to BlueMix.
Temperature: 26.00
Vis: 259
UV: 0.01
```

```
void setup() {
    Serial.begin(9600);
    while (!Serial) { delay(10); }
    if (! uv.begin()) {
        Serial.println("Didn't find S1145");
        while (1); }
}
```



```

Feather.printVersions();
while ( !connectAP() ) { delay(500); }
Feather.printNetwork();
client.begin("OrgID.messaging.internetofthings. \
            ibmcloud.com", 1883, net);
Serial.print("Connecting to BlueMix");
while ( !client.connect("d:OrgID:DeviceT:DeviceID", \
    "use-token-auth", "AuthenticationToken") ) {
    Serial.print(".");
    delay(1000);
}
Serial.println();
Serial.println("Connected to BlueMix.");
}

```

Secțiunea *loop()* va fi responsabilă cu citirea parametrilor (temperatură de la senzorul TMP102; intensitate lumină și index radiații ultraviolete de la senzorul Si1145) și cu postarea acestora către serviciul IBM Watson IoT. Publicarea datelor achiziționate se va face o dată pe minut.

```

void loop() {
    client.loop();
    float temperature = getTemperature();
    Serial.print("Temperature: ");
    Serial.println(temperature);
    int light = uv.readVisible();
    Serial.print("Vis: "); Serial.println(light);
    float UVindex = uv.readUV();
    UVindex /= 100.0;
    Serial.print("UV: "); Serial.println(UVindex);
    client.publish("iot-2/evt/status/fmt/json", \
        "{ \"d\": { \"myName\": \"SunDetector\" \
        , \"Temperature\": \"" + String(temperature, 2) + "\" \

```



```

, \"Light\": \"\" + String(light, DEC) + \"\" \
, \"UVIndex\": \"\" + String(UVindex, 2) + \"\"}}");
delay(60000);
}

```

Procedura *messageReceived()* este necesară pentru a respecta caracterul bidirecțional al protocolului MQTT chiar dacă sistemul nostru nu va prelua date sau comenzi de la platforma IBM Watson IoT.

```

void messageReceived(String topic, String payload, \
    char * bytes, unsigned int length) {
    Serial.print("incoming: ");
    Serial.print(topic);
    Serial.print(" - ");
    Serial.print(payload);
    Serial.println(); }

```

Procedura *connectAP()* este folosită în secțiunea *loop()* pentru conectarea la rețeaua WiFi.

```

bool connectAP(void)
{Serial.print("Please wait while connecting to: \" \
    WLAN_SSID \" ... ");
    if ( Feather.connect(WLAN_SSID, WLAN_PASS) )
    { Serial.println("Connected!"); }
    else {
        Serial.printf("Failed! %s (%d)", Feather.errstr(), \
            Feather.errno());
        Serial.println();
    }
    Serial.println();
    return Feather.connected(); }

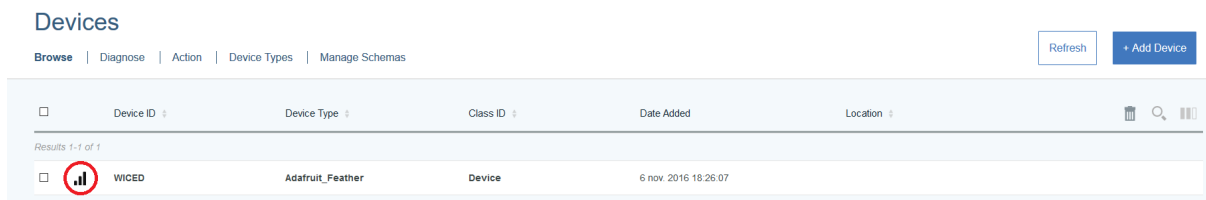
```

Având în vedere că pentru utilizarea senzorului TMP102 nu utilizăm nici o bibliotecă software, funcția *getTemperature()* este utilizată pentru a prelua valoarea temperaturii măsurate de senzor prin intermediul protocolului I2C.

```
float getTemperature()  
{byte MSB = 0x00;  
  byte LSB = 0x00;  
  int TempCistita = 0;  
  float TempCelsius = 0.0;  
  Wire.beginTransmission(tmp102Address);  
  Wire.write(0x00);  
  Wire.endTransmission();  
  Wire.requestFrom(tmp102Address, 2);  
  Wire.endTransmission();  
  MSB = Wire.read();  
  LSB = Wire.read();  
  TempCistita = ((MSB << 8) | LSB) >> 4;  
  TempCelsius = TempCistita * 0.0625;  
  return TempCelsius; }
```

Programul a fost dezvoltat și testat utilizând Arduino IDE 1.6.12, Adafruit WICED 0.6.2, MQTT 1.10.1 și Adafruit Si1145 1.0.0.

După punerea în funcțiune a sistemului în cadrul consolei de administrare IBM Watson IoT (*dashboard*) se va observa schimbarea stării dispozitivului înregistrat anterior.



The screenshot shows the 'Devices' section of the IBM Watson IoT dashboard. At the top, there are tabs for 'Browse', 'Diagnose', 'Action', 'Device Types', and 'Manage Schemas'. On the right, there are 'Refresh' and '+ Add Device' buttons. Below the tabs is a table with columns: Device ID, Device Type, Class ID, Date Added, and Location. The table shows one device with the ID 'WICED', type 'Adafruit_Feather', class 'Device', and date '6 nov. 2016 18:26:07'. The device status is 'Online', indicated by a green circle with a white signal icon.

Device ID	Device Type	Class ID	Date Added	Location
WICED	Adafruit_Feather	Device	6 nov. 2016 18:26:07	

Printr-o simplă inspecție a proprietăților dispozitivului se pot vedea datele trimise de acesta:

Device ID

Device Type

Date Added

Added By

Connection State

WICED

Adafruit_Feather

duminică, 6 noiembrie 2016

indun@adapth.com

Connected on marți, 8 noiembrie 2016 at 21:13:54 from 172.20.230.103 with an insecure connection [Refresh](#)

Recent Events

Event

Format

Time Received

status

json

9 nov. 2016 19:53:41

status

json

9 nov. 2016 19:54:41

Sensor Information

Event

Datapoint

Value

Time Received

status

d.myName

SunDetector

9 nov. 2016 19:54:41

status

d.Temperature

25.31

9 nov. 2016 19:54:41

În lecția următoare vom utiliza informațiile furnizate de sistem pentru a genera grafice de evoluție și alerte. Tot pe baza informațiilor primite de platforma IBM Watson IoT se pot construi și aplicații specifice serviciului cloud IBM BlueMix™, aplicații ce pot prelucra aceste informații în diverse scopuri (statistice, decizionale).

Platforma propusă de această lecție nu este singura variantă hardware de implementare, se pot vedea și alte soluții în următoarele proiecte:

Heart Rate Monitor using IBM Watson IoT Platform + Arduino MKR1000 + Pulse Sensor + NodeRED - developerWorks Recipes

<https://developer.ibm.com/recipes/tutorials/heart-rate-monitor-using-ibm-watson-iot-platform-arduino-mkr1000-pulse-sensor-nodered/>

Connect an ESP8266 with the Arduino SDK to the IBM Watson IoT Platform

https://espressif.com/en/media_overview/articles/connect-esp8266-arduino-sdk-ibm-watson-iot-platform

Arduino + Xbee + Bluemix = Connect Devices to the Cloud

<http://www.paasmag.com/2015/05/13/arduino-xbee-bluemix-connect-devices-to-the-cloud/>