

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.

Realizarea unui sistem de tip Home Automation (Partea a III-a)

Instalarea și configurarea platformei OpenHab pe un sistem Microsoft Windows

Platforma OpenHab este un produs open-source de tip Home Automation, dezvoltat în Java, compatibil cu majoritatea sistemelor de operare actuale: Microsoft Windows, Linux, OS X. OpenHab permite monitorizarea și controlul centralizat ale tuturor echipamentelor inteligente din locuință (senzori, echipamente anti-efracție, sisteme de acționare, sisteme multimedia, sisteme de climatizare) fiind independent de un standard proprietar de comunicație sau de un anumit tip de dispozitiv.



<http://www.openhab.org/>

În cadrul lecției de față vom exemplifica conectarea cu o rețea de elemente MySensors dar în lecțiile viitoare vom extinde integrarea și cu alte platforme.

Interfața utilizator a platformei este disponibilă în format web dar și sub forma de aplicații native Android și iOS.



Instalarea sub platforma Microsoft Windows necesită instalarea prealabilă a mediului de execuție Java:

<http://java.com>

după care se copiază ultima versiune (stabilă de preferat) a OpenHab Runtime Core:

<http://www.openhab.org/getting-started/downloads.html>

se dezarchivează în rădăcina unuia dintre drive-urile sistemului de calcul (de exemplu C:\) sub directorul *openhab* (de exemplu C:\openhab). Instalarea este gata, urmează partea de configurare. Este posibilă copierea unui set de configurații demo (*Demo setup* – conține un set complet de configurații ca exemplu) dar este recomandată configurarea treptată de la zero.

Acești pași de instalare și configurare au fost testați pe un sistem desktop Microsoft Windows 10 x64 având instalat pachetul Oracle Java 8 Update 101 și utilizând versiunea OpenHab 1.8.3.

Primul pas de configurare presupune ca în subdirectorul *configurations* se redenumește fișierul *openhab_default.cfg* în *openhab.cfg*. Acest fișier conține toate setările serviciului OpenHab. Pentru a funcționa în conjuncție cu o rețea MySensors se adaugă la sfârșitul fișierului o secțiune nouă:

```
##### MySensors #####  
mysensors:port=COMXX
```

pentru funcționarea în conjuncție cu un gateway serial (unde COMXX este portul pe care este conectat gateway-ul serial) sau:

```
##### MySensors #####  
mysensors:type=ethernet  
mysensors:host=192.168.100.7  
mysensors:port=5003
```

pentru un gateway ethernet (unde *host* este adresa IP a sistemului gateway).

Comunicația între platforma OpenHab și protocolul serial specific rețelei MySensors este asigurată de o componentă software de legătură (binding add-on) ce trebuie copiată în subdirectorul *addons* al instalării de la adresa:

<http://bkl.linux.dk/org.openhab.binding.mysensors-1.8.0-SNAPSHOT.jar>

Există o colecție implicită de astfel de componente software ce fac legătura cu diverse alte sisteme. Ea se copiază de la aceeași adresă ca și Runtime Core – componenta Addons. Nu este recomandată copierea tuturor componentelor în subdirectorul instalării efectuate anterior deoarece o colecție mare de componente încărcate la pornirea platformei OpenHab îngreunează funcționarea. Pentru început vom copia doar *org.openhab.binding.ntp-1.8.3.jar* – componentă ce ne va permite afișarea orei și datei preluate de un server NTP.

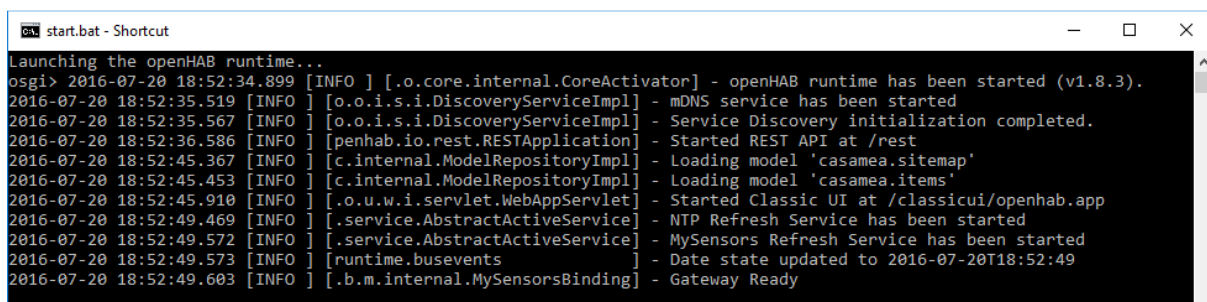
Următorul pas este realizarea fișierului de configurație al ”centrului de comandă” pe care dorim să îl creăm. În subdirectorul *configurations\sitemaps* creăm un fișier *casamea.sitemap* (unde casamea poate fi înlocuit cu ce denumire dorim). Cu ajutorul unui editor text decent (se recomandă Notepad++) se editează acest fișier și se completează:

```
sitemap casamea label="Meniu Principal" {
    Frame label="Senzori de temperatura" {
        Text item=Temperature0 icon="temperature"
        Text item=Temperature1 icon="temperature"
    }
    Frame label="Data calendaristica" {
        Text item=Date icon="calendar"
    }
}
```

Interfața creată în cadrul acestui fișier de configurare va avea două secțiuni: Senzori de temperatura (în care vom afișa datele primite de la cei doi senzori ai sistemului construit în lecția anterioară) și Data calendaristica (în care vom afișa data). Pentru a putea afișa aceste informații trebuie să creăm un fișier *casamea.items* în subdirectorul *configurations\items* ce va conține următoarele definiții:

```
Number Temperature0      "Temperatura interioara [%s °C]" <temperature>
                           {mysensors="10;0;V_TEMP"}
Number Temperature1      "Temperatura exterioara [%s °C]" <temperature>
                           {mysensors="10;1;V_TEMP"}
DateTime Date             "Date [%1$tA, %1$td.%1$tm.%1$tY]" <calendar>
                           {ntp="Europe/Bucharest:ro_RO"}
```

În acest moment sistemul este gata de utilizare. Se pornește platforma utilizând fișierul *start.bat* din directorul de instalare. Se va deschide o fereastră de forma:



```
start.bat - Shortcut
Launching the openHAB runtime...
osgi> 2016-07-20 18:52:34.899 [INFO ] [.o.core.internal.CoreActivator] - openHAB runtime has been started (v1.8.3).
2016-07-20 18:52:35.519 [INFO ] [o.o.i.s.i.DiscoveryServiceImpl] - mDNS service has been started
2016-07-20 18:52:35.567 [INFO ] [o.o.i.s.i.DiscoveryServiceImpl] - Service Discovery initialization completed.
2016-07-20 18:52:36.586 [INFO ] [penhab.io.rest.RESTApplication] - Started REST API at /rest
2016-07-20 18:52:45.367 [INFO ] [c.internal.ModelRepositoryImpl] - Loading model 'casamea.sitemap'
2016-07-20 18:52:45.453 [INFO ] [c.internal.ModelRepositoryImpl] - Loading model 'casamea.items'
2016-07-20 18:52:45.910 [INFO ] [.o.u.w.i.servlet.WebAppServlet] - Started Classic UI at /classicui/openhab.app
2016-07-20 18:52:49.469 [INFO ] [.service.AbstractActiveService] - NTP Refresh Service has been started
2016-07-20 18:52:49.572 [INFO ] [.service.AbstractActiveService] - MySensors Refresh Service has been started
2016-07-20 18:52:49.573 [INFO ] [runtime.busevents] - Date state updated to 2016-07-20T18:52:49
2016-07-20 18:52:49.603 [INFO ] [.b.m.internal.MySensorsBinding] - Gateway Ready
```

Pentru ca platforma OpenHab să pornească automat la pornirea sistemului trebuie copiat un shortcut al fișierului *start.bat* în directorul *startup* al utilizatorului dorit sau se poate defini un *scheduled task*.

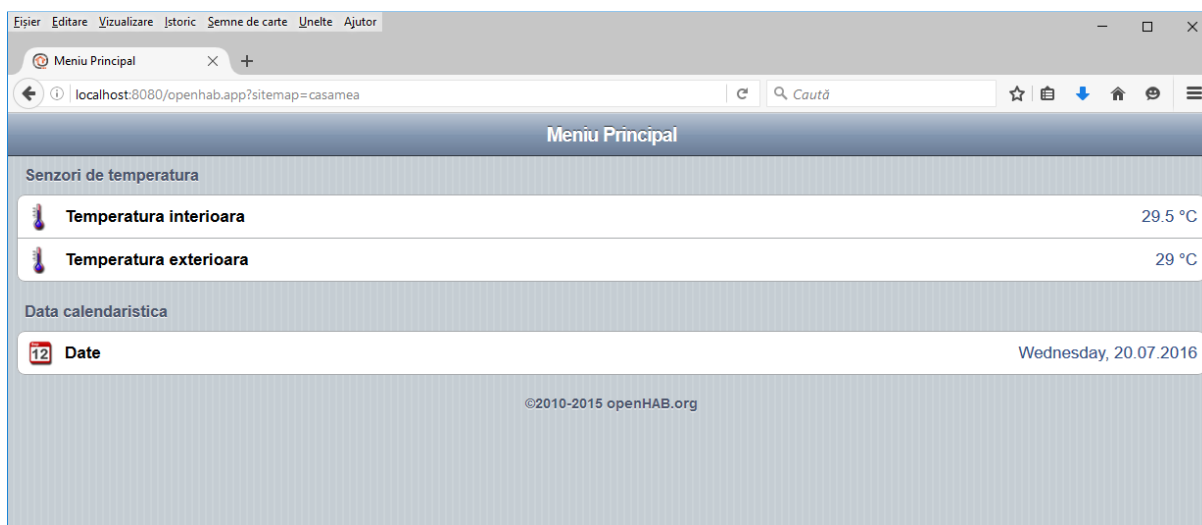
Accesarea interfeței OpenHab se face dintr-un browser web accesând pagina:

<http://192.168.100.2:8080/openhab.app?sitemap=casamea>

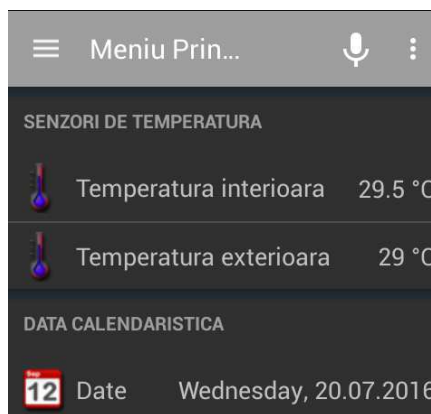
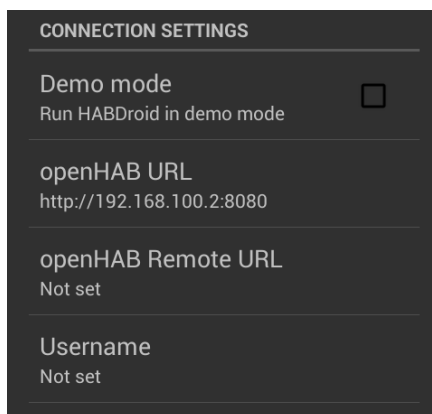
unde 192.168.100.2 este adresa sistemului pe care s-a efectuat instalarea sau

<http://localhost:8080/openhab.app?sitemap=casamea>

dacă se accesează de pe același sistem. Captură de ecran cu fereastra de browser:



Alternativ interfața se poate accesa prin intermediul unei aplicații mobile Android sau iOS configurând adresa platformei OpenHab în aplicație. Capturi de ecran cu setarea aplicației Android și cu interfața "Meniu Principal":



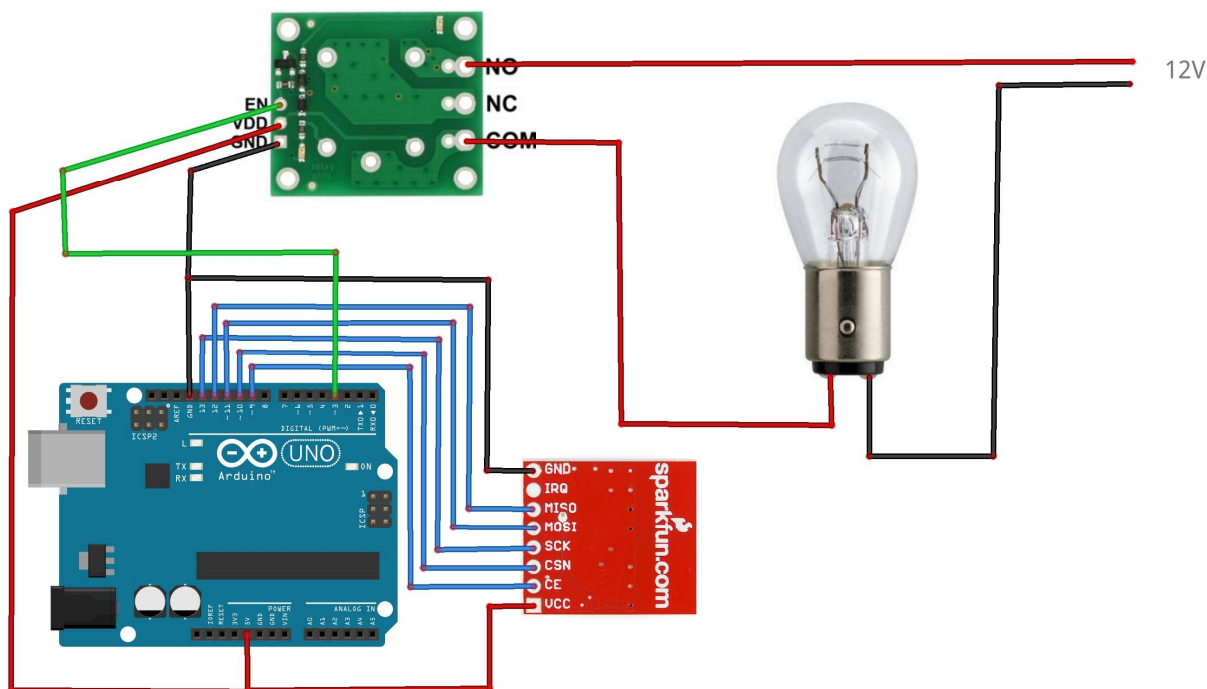
Construirea și integrarea unui element de acționare

Explicațiile de până în acest moment au arătat cum se poate integra un element de achiziție format din doi senzori de temperatură cu platforma OpenHab. Vom continua cu construirea și integrarea unui element de acționare bazat pe un relee ce acționează un bec de 12V (o lampă de birou). Pentru acest lucru vom conecta la sistemul de bază (Arduino Uno + SparkFun Transceiver Breakout - nRF24L01+) o placă relee SPDT 5VDC. Conexiunea între placa de dezvoltare și placa relee se va realiza prin intermediul pinului digital D3 al plăcii de dezvoltare conectat la pinul EN al plăcii relee.



<https://www.robofun.ro/module/module-releu/releu-spd-5V>

Schema sistemului va arăta precum în diagrama următoare:



Atenție, placa relee nu este proiectată pentru tensiuni mari – utilizarea la tensiuni mai mari de 12V este extrem de periculoasă!

Programul va utiliza biblioteca MySensors 2.0.0 (la fel ca și exemplele din lecțiile anterioare):

```
#define MY_DEBUG
#define MY_RADIO_NRF24
#define MY_REPEATER_FEATURE
#define MY_NODE_ID 20
#define MY_BAUD_RATE 9600

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

Exemplul este gândit pentru a funcționa cu un număr mai mare de relee (număr limitat doar de numărul de pini digitali ai plăcii de dezvoltare). Se pot conecta mai multe relee pe următorii pini digitali D4, D5..., D8.

```
#define RELAY_1 3
#define NUMBER_OF_RELAYS 1
#define RELAY_ON 1
#define RELAY_OFF 0
```

Secțiunea *before()* este specifică bibliotecii MySensors 2.0.0 și se execută înainte inițializarea mecanismelor interne bibliotecii. În cadrul acestei secțiuni se vor inițializa pinii digitali ca ieșiri de comandă și se va restaura starea precedentă a pinilor – salvată în memoria EEPROM în secțiunea *receive()* – mecanism util în cazul unor reporniri neprogramate ale elementului de acționare.

```
void before() {
    for (int sensor=0, pin=RELAY_1; sensor<NUMBER_OF_RELAYS;
        sensor++, pin++) {
        pinMode(pin, OUTPUT);
        digitalWrite(pin,
            loadState(sensor)?RELAY_ON:RELAY_OFF);
    }
}
```

În cadrul secțiunii *presentation()* (specifică bibliotecii, se realizează prezentarea elementului de acționare către sistemul gateway) se va transmite numele, versiunea și funcționalitatea elementului de acționare.

```
void presentation() {
    sendSketchInfo("Relay", "1.0");
    for (int sensor=1, pin=RELAY_1;
        sensor<=NUMBER_OF_RELAYS; sensor++, pin++) {
        present(sensor, S_LIGHT);    }
}
```

Secțiunile *setup()* și *loop()*, tipice pentru program Arduino, nu conțin nici o instrucțiune. Întregul mecanism de control al releului se realizează în secțiunea *receive()* – secțiune apelată în mod automat de mecanismele interne ale bibliotecii MySensors la primirea unui mesaj radio.

```
void setup() {  
}  
  
void loop() {  
  
void receive(const MyMessage &message) {  
    if (message.type==V_STATUS) {  
        digitalWrite(message.sensor+RELAY_1,  
            message.getBool()?RELAY_ON:RELAY_OFF);  
        saveState(message.sensor, message.getBool());  
        Serial.print("Incoming change for sensor:");  
        Serial.print(message.sensor);  
        Serial.print(", New status: ");  
        Serial.println(message.getBool());  
    }  
}
```

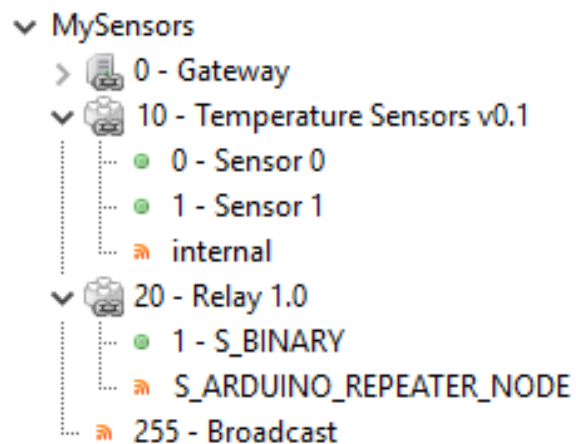
Integrarea noului sistem în cadrul platformei OpenHab presupune următoarele completări în fișierele de configurare:

- în fișierul *casamea.items* vom adăuga:

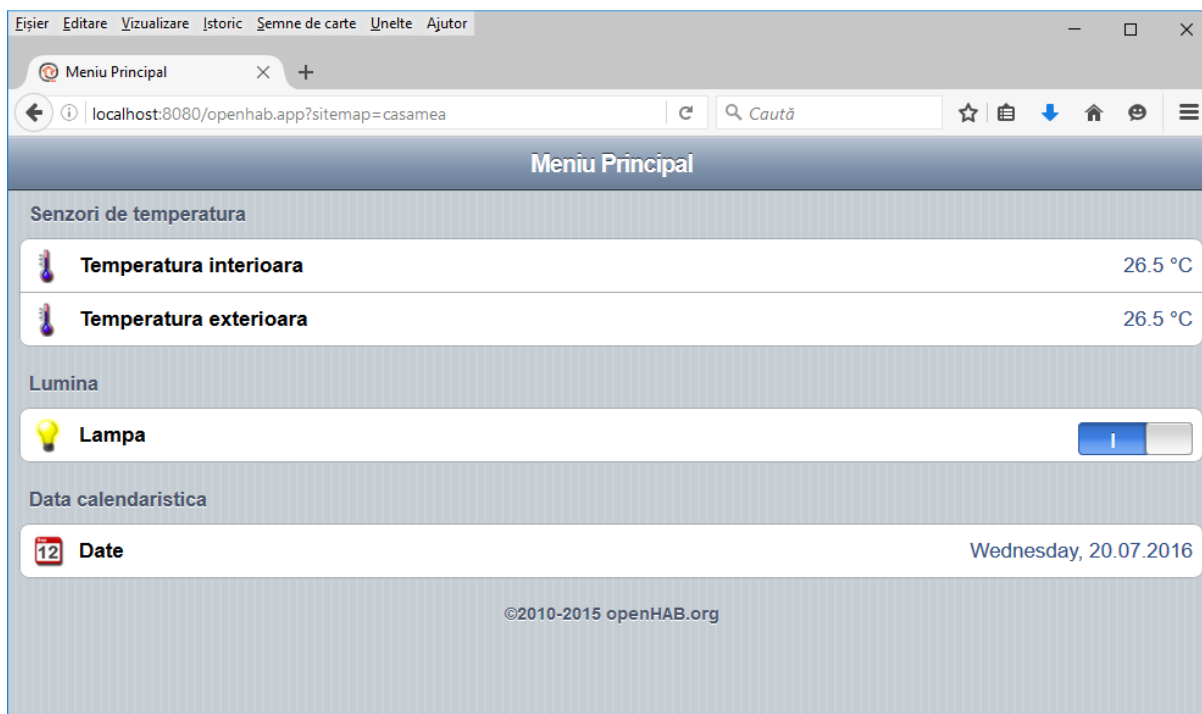
```
Switch Relay "Lampa" {mysensors="20;0;V_STATUS"}
```

- în fișierul *casamea.sitemap* vom adăuga:

```
Frame label="Lumina" {  
    Switch item=Relay icon="switch"  
}
```



În urma acestor modificări interfața de comandă a rețelei MySensors va arăta:



În cadrul materialului viitor vom vedea cum se instalează și configurează platforma OpenHab pe un sistem specializat Raspberry Pi precum și cum putem implementa în cadrul platformei reguli (Rules) ce permit implementarea unor mecanisme automate de comandă.