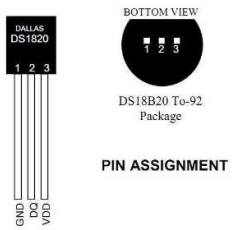
Realizarea unui sistem de tip Home Automation (Partea a II-a)

Realizarea unui sistem de măsurare a temperaturii compatibil cu platforma MySensors

Unul dintre cele mai comune sisteme de măsurare este cel de măsurare a temperaturii. El este utilizat atât din considerente de evaluare a confortului personal (temperatura din locuință) cât și din considerente legate de funcționare a sistemului de climatizare sau considerente legate de supravegherea funcționării altor echipamente electronice – avertizare în caz de supraîncălzire. În cadrul acestei lecții vom implementa un sistem de măsurare a temperaturii cu mai multe zone utilizând senzorul digital de temperatură DS18B20 produs de Maxim Integrated:



https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf
https://www.robofun.ro/senzori/vreme/senzor-temperatura-DS18B20

Acest senzor are o plajă de măsurare de -55°C / +125°C și o acuratețe de 0.5°C. Pentru interconectarea cu placa de dezvoltare acest senzor utilizează protocolul 1-Wire:

https://www.maximintegrated.com/en/products/digital/one-wire.html

permițând încascadarea mai multor senzori pe aceiași linie de comunicație – lucru ce facilitează implementarea funcționalității de măsurare a temperaturii în mai multe zone.

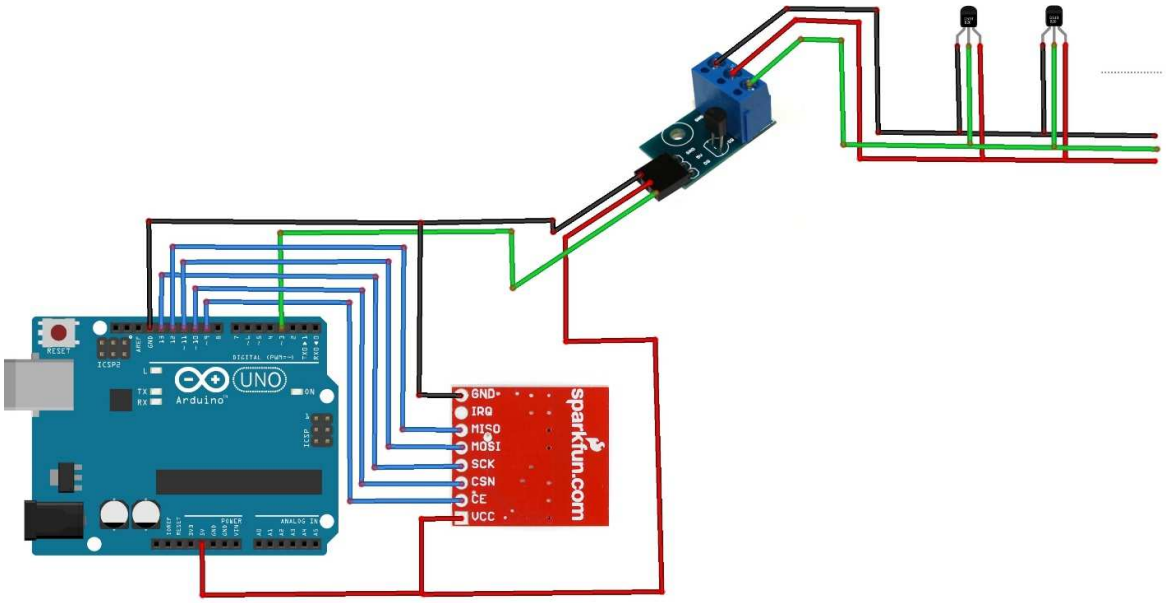
Deoarece magistrala de comunicație necesită o rezistență de pull-up, primul senzor din rețea va fi de tip brick, modul ce include rezistența necesară funcționării corecte a comunicației.

https://www.robofun.ro/senzori/vreme/senzor-temperatura-inlantuibil-brick-DS18B20-motherboard

Soluția bazată pe comunicația 1-Wire a fost aleasă pentru distanța mare la care pot fi plasați senzorii față de placa de dezvoltare (aproximativ 200 de metri) și a unui număr mare de senzori care pot fi conectați simultan (având în vedere faptul că fiecare senzor are o adresă proprie din fabricație pe 64 de biți nu există o limitare de protocol privind numărul de senzori dintr-o rețea). A se vedea și:

Guidelines for Reliable Long Line 1-Wire Networks
http://www.maximintegrated.com/en/app-notes/index.mvp/id/148

Prin intermediul rețelei de senzori DS18B20 proiectul va putea monitoriza din punct de vedere al măsurării temperaturii un număr nelimitat de zone. Conectarea cu placa de dezvoltare implică conectarea componentei brick în felul următor: pinul GND al modulului la unul din pinii de GND ai plăcii de dezvoltare, pinul 5V la unul din pinii de 5V ai plăcii de dezvoltare și pinul DQ al modulului la pinul D3 al plăcii de dezvoltare. Următorii senzori atașați în rețea se vor conecta la modulul brick prin intermediul conectorului cu șurub respectând semnificația pinilor.



Schema sistemului (diagramă anterioară) se bazează pe schema din lecția precedentă (Arduino Uno + SparkFun Transceiver Breakout - nRF24L01+) dar, dacă se dorește realizarea unui sistem portabil alimentat de la baterii, se poate înlocui placa de dezvoltarea Arduino Uno cu o una din variantele mai mici ca dimensiuni: Arduino Pro Mini 5V sau 3.3V:

https://www.robofun.ro/platforme/arduino_dev/arduino_pro_mini_328_8mhz
https://www.robofun.ro/platforme/arduino_dev/arduino_pro_mini_328_16mhz

Programul sistemului utilizează bibliotecile MySensors 2.0.0, OneWire 2.3.2 și DallasTemperature 3.7.6:

https://github.com/mysensors/MySensors/tree/master
https://github.com/PaulStoffregen/OneWire
https://github.com/milesburton/Arduino-Temperature-Control-Library

Inițializările necesare comunicației radio se fac înainte de includerea bibliotecii MySensors deoarece biblioteca conține instanțierea obiectelor de lucru:

```
#define MY_DEBUG
#define MY_RADIO_NRF24
#define MY_NODE_ID 10
#define MY_BAUD_RATE  9600

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

#define SKETCH_NAME "Temperature Sensors"
#define SKETCH_VERSION "v0.1"

#define LONG_WAIT 500
#define SHORT_WAIT 50

unsigned long SLEEP_TIME = 900000;
boolean metric = true;

MyMessage msg_S_TEMP(0,V_TEMP);

#define ONE_WIRE_BUS 3
#define MAX_ATTACHED_DS18B20 10

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

int numSensors=0;
```

În cadrul secțiunii *setup()* se va realiza inițializarea comunicației OneWire:

```
void setup() {
  sensors.begin();
  wait(LONG_WAIT);
}
```

Secțiunea *presentation()* este specifică bibliotecii MySensors și conține partea de inițializare a comunicației radio cu sistemul gateway (prezentare denumire și versiune sistem, solicitare metrică gateway, prezentare senzori atașați sistemului):

```
void presentation()  {
  Serial.print("Send Sketch Info: ");
  sendSketchInfo(SKETCH_NAME, SKETCH_VERSION);
  Serial.print(SKETCH_NAME);
  Serial.println(SKETCH_VERSION);
  wait(LONG_WAIT);

  Serial.print("Get Config: ");
  metric = getConfig().isMetric;
  Serial.println(metric ? "Metric":"Imperial");
  wait(LONG_WAIT);

  Serial.println("Presenting Nodes");
  Serial.println("________________");
  Serial.println("  S_TEMP");
  numSensors = sensors.getDeviceCount();
  for (int i=0; i<numSensors && i<MAX_ATTACHED_DS18B20;
                                                  i++) {
    char sensorname[9];
    sprintf(sensorname,"Sensor %i",i);
    present(i,S_TEMP, sensorname);
    wait(SHORT_WAIT);
  }
  Serial.println("________________");
}
```

În cadrul secțiunii *loop()* se va realiza citirea senzorilor de temperatură atașați sistemului și se va transmite radio către sistemul gateway a valorilor măsurate:

```
void loop() {
  sensors.requestTemperatures();
  wait(SHORT_WAIT);
    for (int i=0; i<numSensors && i<MAX_ATTACHED_DS18B20;
                                                  i++) {
    float temperature = static_cast<float>
    (static_cast<int>((getConfig().isMetric?sensors.get
     TempCByIndex(i):sensors.getTempFByIndex(i)) * 10.))
                                                  /10.;
    send(msg_S_TEMP.setSensor(i).set(temperature,1));
  }
  sleep(SLEEP_TIME); }
```

La începutul programului se pot observa directivele ce influențează în mod esențial funcționarea comunicației radio și a mecanismelor interne ale bibliotecii MySensors:

`#define MY_DEBUG` – permite vizualizarea pe consola serială a mesajelor de depanare;

`#define MY_RADIO_NRF24` – indică utilizarea modulelor radio nRF24L01;

`#define MY_NODE_ID 10` – stabilește adresa sistemului (Node ID);

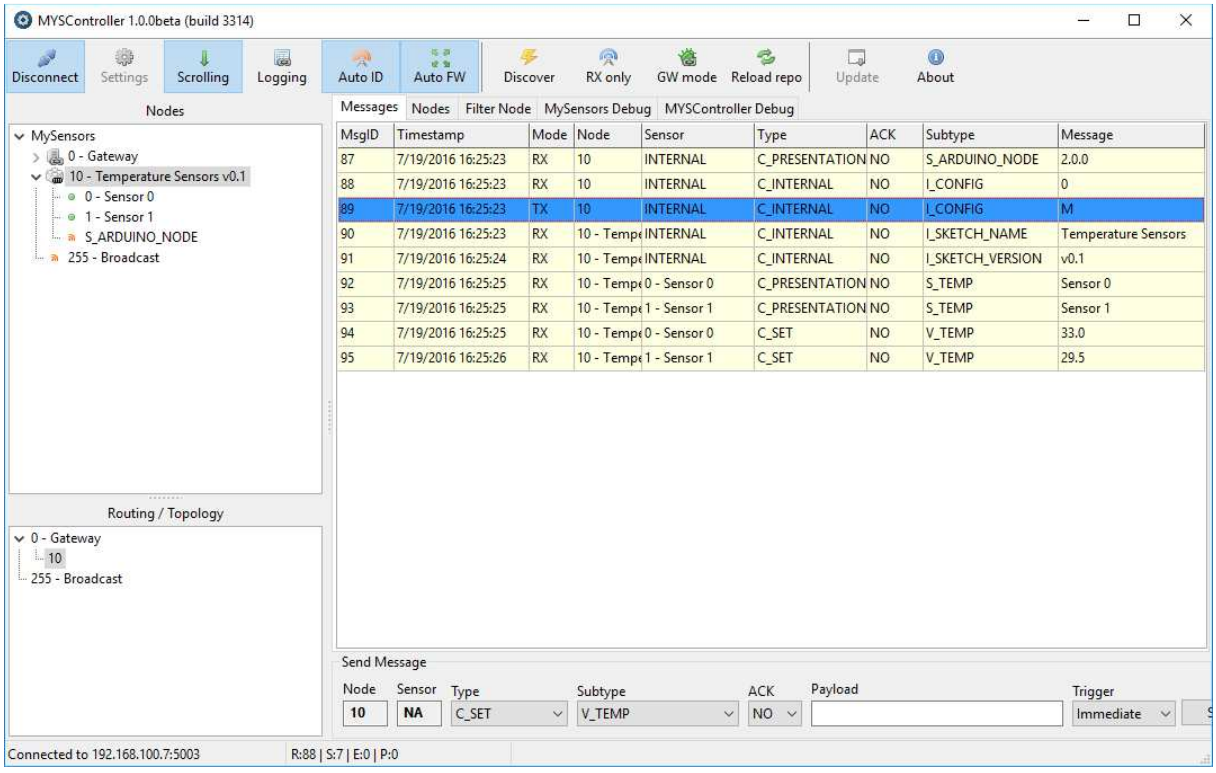
`#define MY_BAUD_RATE  9600` – stabilește viteza de comunicație serială, în cadrul programului nu este necesară inițializarea comunicației seriale prin instrucțiunea Serial.begin(…).

Este interesantă testarea directivei:

`#define MY_RF24_ENABLE_ENCRYPTION`

ce permite cifrarea comunicației între nodurile rețelei radio – trebuie inclusă și în sursa program a sistemului gateway.

Comunicația între sistemul de măsurare și sistemul gateway poate fi examinată cu ajutorul aplicație MYSController. Sistemul de măsurare testat avea instalați doi senzori de temperatură ce pot fi observați ca elemente înregistrate distinct (0 – Sensor 0, 1 – Sensor 1) aparținând nodului 10 – Temperature Sensors v0.1.



Pentru o listă completă a mecanismelor de configurare ale bibliotecii MySensors se poate consulta:

MySensors Arduino Library (v2.0.x)
https://www.mysensors.org/download/sensor_api_20

Pentru mai multe exemple de sisteme de achiziție și acționare compatibile cu platforma MySensors se poate consulta secțiunea *Build* a paginii MySensors:

https://www.mysensors.org/build/

sau secțiunea MySensors în cadrul platformei OpenHardware:

https://www.openhardware.io/explore?search=MySensors

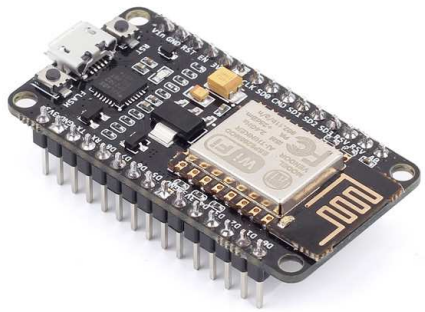
Realizarea unui sistem gateway ethernet

Sistemul gateway prezentat în lecția anterioară, sistem gateway serial, prezintă dezavantajul necesității conectării directe (prin cablu USB) la un sistem de calcul ce rulează o aplicație de interpretare a mesajelor sau, cum vom vedea în lecțiile viitoare, o aplicație specializată de tip Home Automation. Biblioteca MySensors pune la dispoziție și o altă modalitate de retransmisie a mesajelor radio către un sistem de calcul prin intermediul unei rețele ethernet.

Există două exemple în biblioteca MySensors ce pot fi utilizate în acest sens:

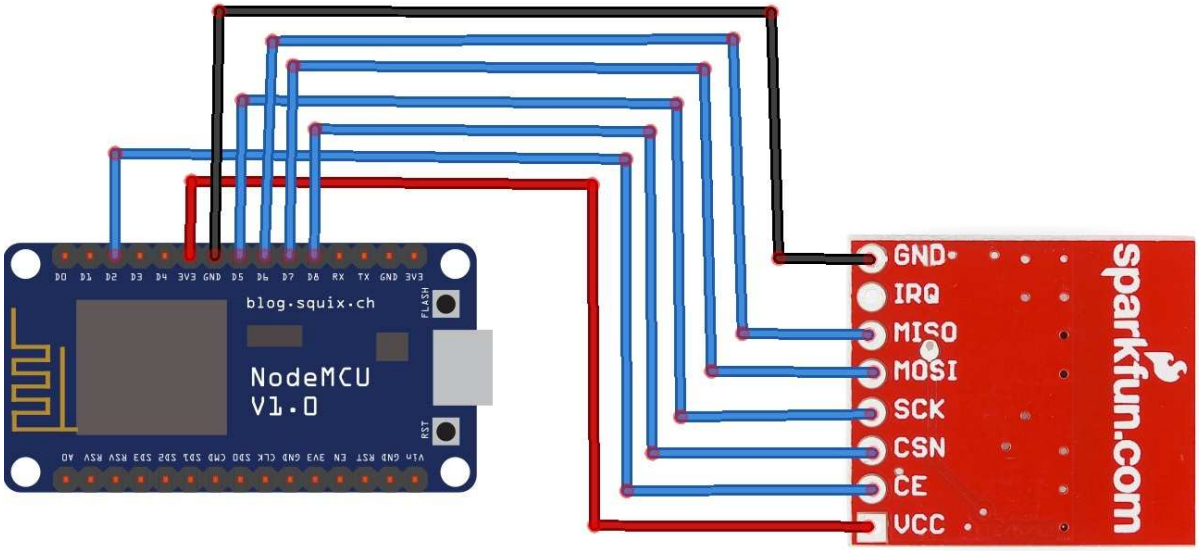
GatewayW5100 – implementează un sistem gateway de retransmisie ethernet a mesajelor radio bazat pe elemente hardware ce utilizează controlerul ethernet Wiznet 5100 precum shield-ul Arduino Ethernet Shield sau placa Arduino Ethernet.

GatewayESP8266 – implementează un sistem gateway de retransmisie ethernet WiFi utilizând placa de dezvoltare NodeMCU bazată pe controlerul WiFi ESP8266 – soluție exemplificată în cele ce urmează.



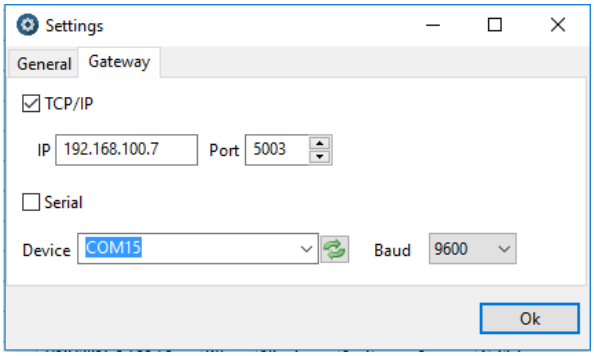
https://www.robofun.ro/wireless/wireless-wifi/NodeMCUv2-ESP8266

Interconectarea între placa de dezvoltare NodeMCU și SparkFun Transceiver Breakout - nRF24L01+ este ilustrată în diagrama următoare:



Conexiunile dintre cele două componente sunt: VCC – 3v3, CE – D2, CSN – D8, SCK – D5, MOSI – D7, MISO – D6 și GND – GND.

Sistemul GatewayESP8266 poate fi utilizat atât ca și gateway serial cât și ca gateway ethernet WiFi– ambele configurații pot fi utilizate în configurarea comunicației cu sistemul de calcul de control – avantajul utilizării variantei ethernet WiFi este decuplarea fizică a sistemului gateway de sistemul de control.



În materialele următoare vom prezenta instalarea și configurarea platformei OpenHab ca platformă de control și vizualizare a datelor provenite de la rețeaua MySensors.