

Module Internet des Objets Arduino

Master Data Science et Big Data

TP 4 : Atelier d'Arduino et Node-Red

Réalisé par :

KARDI Hanane

OUKHELLOU Smail

BADI Zakaria

KHALEQ Mounir

Encadré par :

M. Sate Youness

1. Objectifs

La réalisation d'une maquette Arduino en utilisant la (circuit intégré) PCB Arduino et l'environnement Node-Red.

2. Contexte :

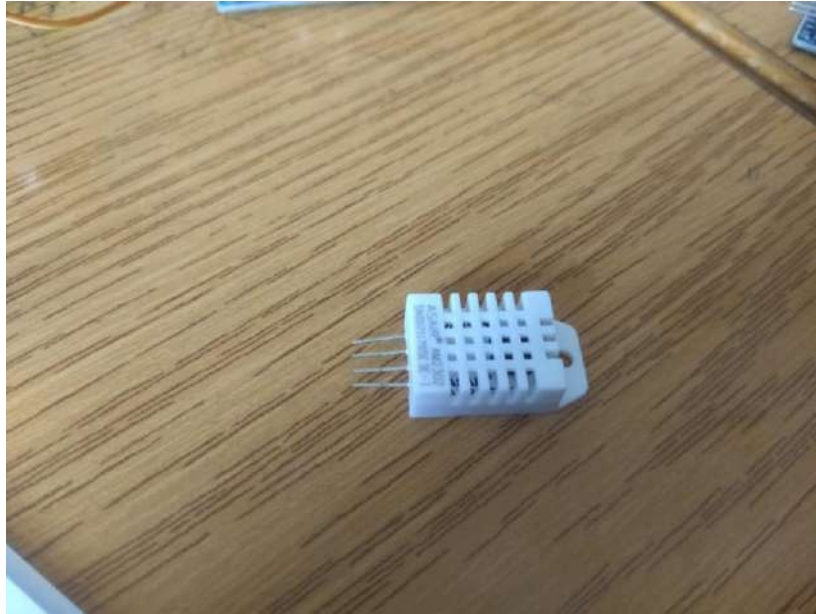
Au cours de ce TP, on va essayer de créer un système Arduino pour contrôler notre environnement en suivant différentes étapes tout en commençant par la liaison des différents composants puis la programmation de la carte Arduino ensuite la connexion entre carte et la Framework Node-Red.

3. Ressources requises :

Carte Arduino Mega



Capteur de Température



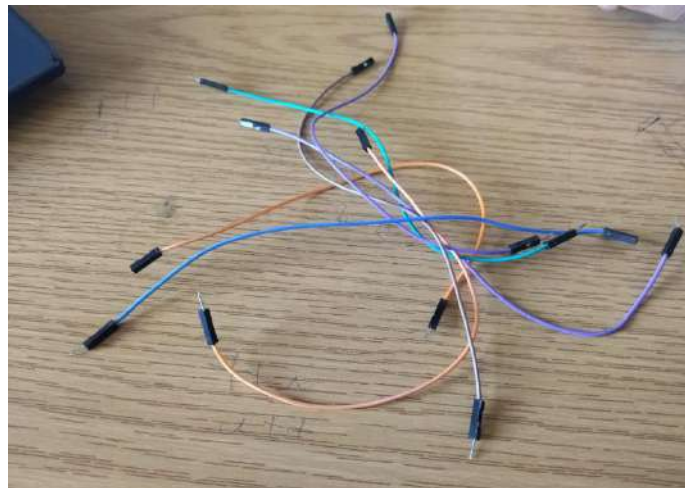
Capteur d'Humidité de Sol



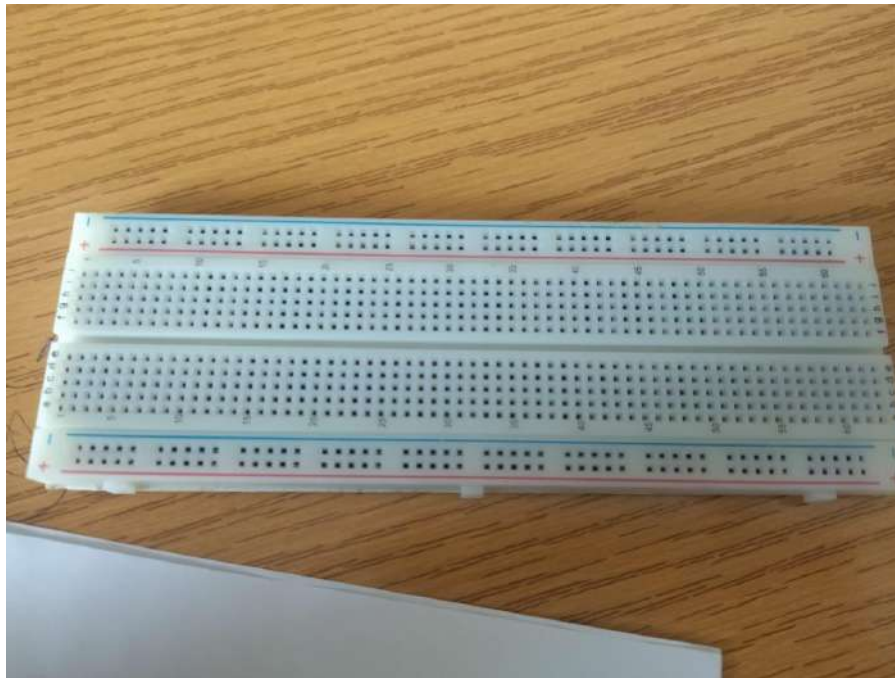
Capteur de Pluie



Fil / Résistance 220 ohm



Board



Arduino IDE

The screenshot shows the Arduino IDE interface. The main editor window displays a C++ sketch for reading temperature and humidity from a DHT sensor. The sketch includes a `void loop()` function that reads data every 2000 milliseconds, converts the humidity value to a percentage, and prints the results to the serial monitor. The serial monitor window on the right shows the output of the sketch, displaying temperature and humidity readings from both the DHT sensor and an analog pin A0. The status bar at the bottom indicates that the sketch uses 5328 bytes of program storage space and 301 bytes of dynamic memory.

```
sketch_nov01a] Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help

sketch_nov01a
void loop() {
  delay(2000); // Wait for 2 seconds between readings

  // Read humidity from analog pin A0
  int humidityValue = analogRead(HUMIDITY_PIN);

  // Convert humidity value to percentage (assuming a range of 0
  float humidityPercentage = map(humidityValue, 0, 1023, 0, 100);

  // Read temperature and humidity from the DHT sensor
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" °C");

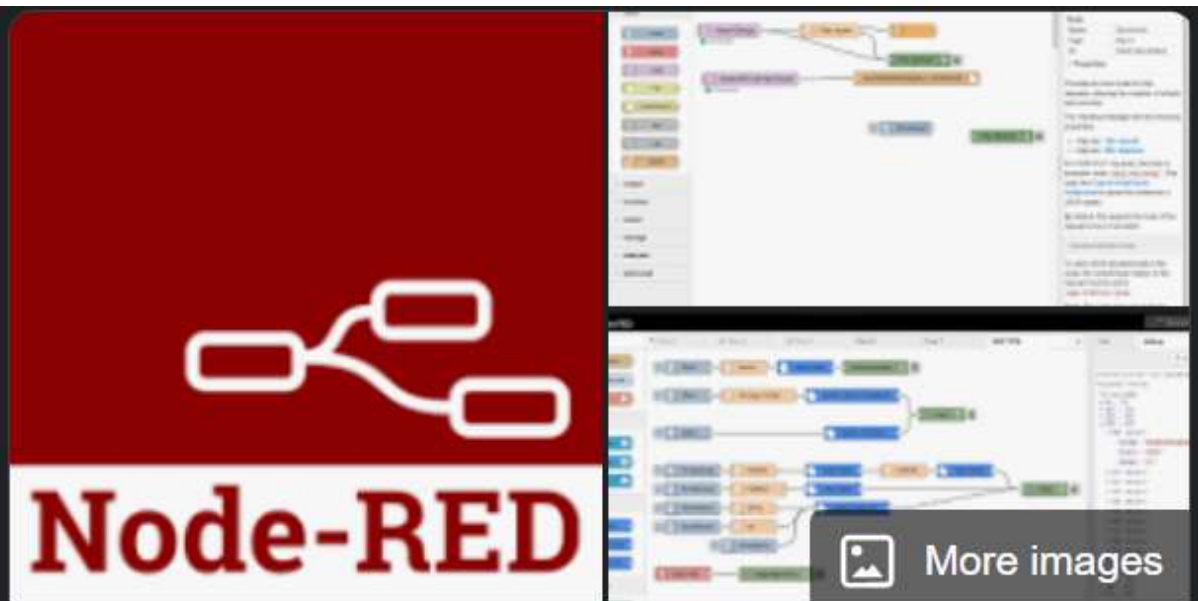
  Serial.print("Humidity (from DHT sensor): ");
  Serial.print(humidity);
  Serial.println(" %");
}

Data uploaded
Sketch uses 5328 bytes (16%) of program storage space. Maximum is
Global variables use 301 bytes (14%) of dynamic memory, leaving 1

COM6
Temperature: 20.10 °C
Humidity (from DHT sensor): 49.70 %
Humidity (from analog pin A0): 35.00 %
Temperature: 20.00 °C
Humidity (from DHT sensor): 49.70 %
Humidity (from analog pin A0): 35.00 %
Temperature: 20.00 °C
Humidity (from DHT sensor): 49.60 %
Humidity (from analog pin A0): 35.00 %
Temperature: 20.00 °C
Humidity (from DHT sensor): 49.80 %
Humidity (from analog pin A0): 35.00 %
Temperature: 20.10 °C
Humidity (from DHT sensor): 49.80 %
Humidity (from analog pin A0): 35.00 %
Temperature: 20.00 °C
Humidity (from DHT sensor): 49.70 %
Humidity (from analog pin A0): 35.00 %
Temperature: 20.00 °C
Humidity (from DHT sensor): 49.60 %
Humidity (from analog pin A0): 35.00 %
Temperature: 20.00 °C
Humidity (from DHT sensor): 49.40 %
Humidity (from analog pin A0): 35.00 %

Autoscroll Show timestamp 9600 baud Clear output
```

Node-Red



Node-RED

Computer program :

Node-RED is a flow-based, low-code development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things. Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions. [Wikipedia](#)

4. Implementation

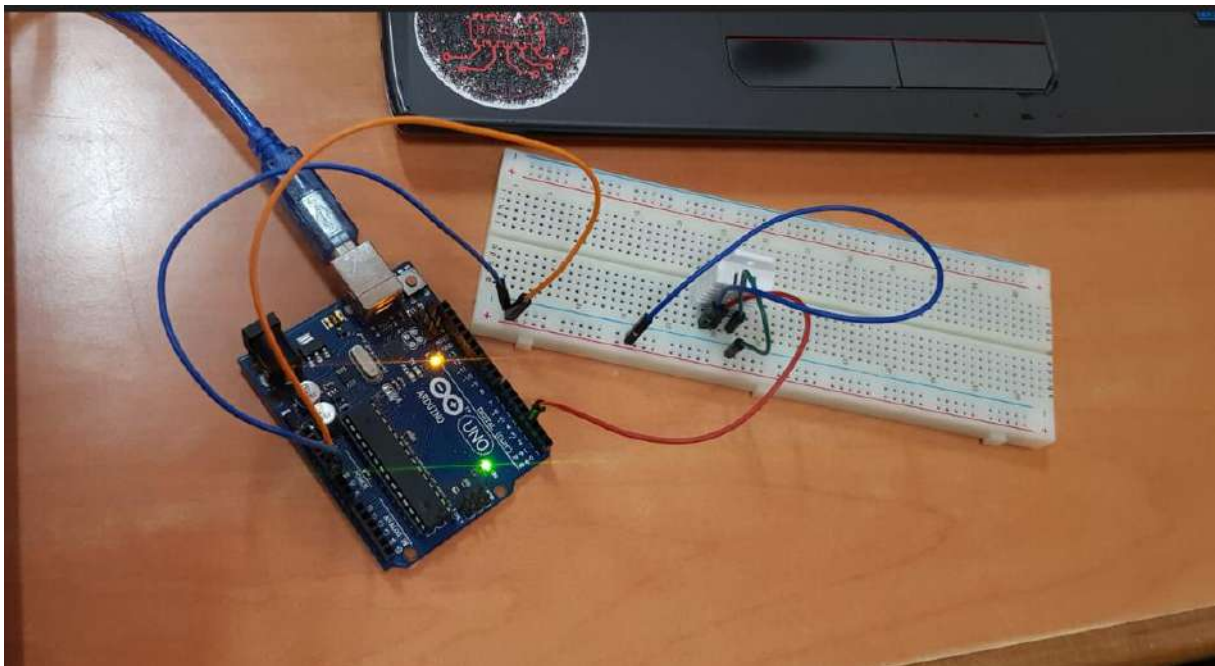
Code Arduino IDE et la maquette électronique

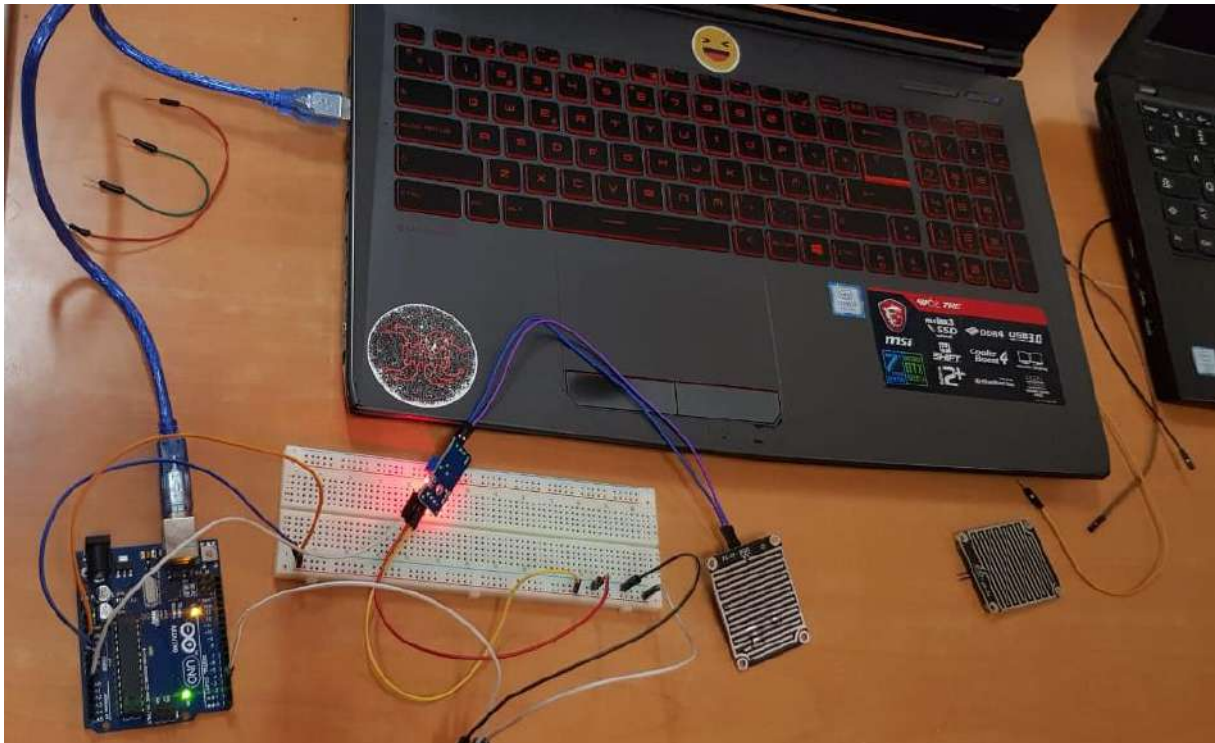
```
#include <DHT.h>

#define DHTPIN 3    // Digital PWM pin for the DHT sensor
#define DHTTYPE DHT22    // Change this to DHT11 if you are using a DHT11 sensor

DHT dht(DHTPIN, DHTTYPE);
```

```
void setup() {  
  Serial.begin(9600);  
  dht.begin();  
}  
  
void loop() {  
  delay(2000); // Wait for 2 seconds between readings  
  
  float temperature = dht.readTemperature();  
  float humidity = dht.readHumidity();  
  
  Serial.print("Temperature: ");  
  Serial.print(temperature);  
  Serial.println(" °C");  
  
  Serial.print("Humidity: ");  
  Serial.print(humidity);  
  Serial.println(" %");  
}
```





Implémentation de flux des données dans Node Red

Dans cette partie on va utiliser les nœuds suivants pour automatiser le flow des données de l'extraction des données à partir d'Arduino en utilisant le port de communication COM 6 :

- Serial Node :

Edit serial in node

Delete

Cancel

Done

Properties

Name

COM6 9600-8N1

Serial Port

Arduino

▼

- Inject Node :pour faire la simulation de la source des données dans un intervalle donne :

Edit inject node

Delete
Cancel
Done

⚙️ Properties
⚙️
📄
🔗

🔑 Name

≡
msg. payload
=
▼ timestamp
✕

≡
msg. topic
=
▼ timestamp
✕

+ add
inject now

☐ Inject once after seconds, then

🔄 Repeat

every

- Function Node : pour générer les données aléatoire de la température et l'humidité similaire de cas réel :

Edit function node

Delete
Cancel
Done

⚙️ Properties
⚙️
📄
🔗

🔑 Name

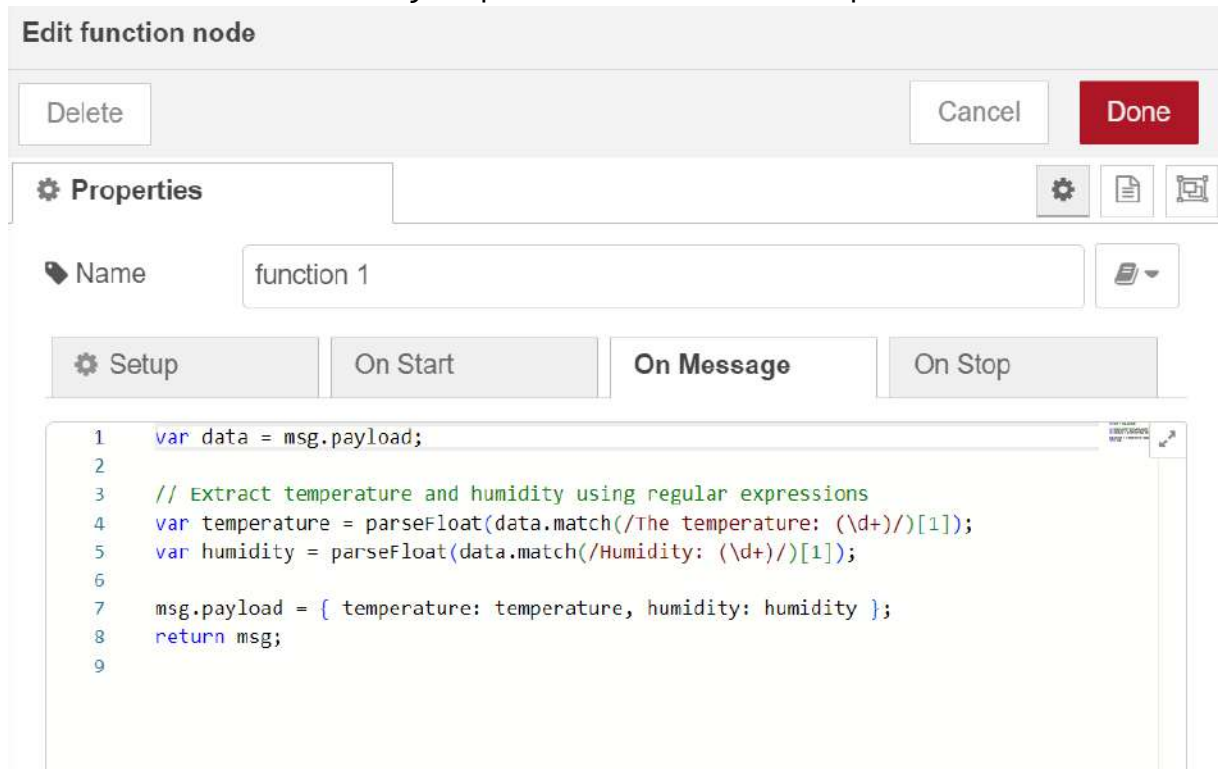
⚙️ Setup
On Start
On Message
On Stop

```

1  var temperature = Math.floor(Math.random() * 100) + 1; // Random temperature betw
2  var humidity = Math.floor(Math.random() * 100) + 1; // Random humidity between 1
3
4  msg.payload = `The temperature: ${temperature}°C\nHumidity: ${humidity}%`;
5  return msg;
6

```

- Extraire les données envoyer par le nœud fonction précédant :



- Construire un fichier csv :

Edit csv node

Delete

Cancel

Done

⚙️ Properties

⚙️

📄

🔗

Columns

temperature,humidity

Separator

comma

Name

Arduino_Data

CSV to Object options

➡️ Input

Skip first 0 lines

☒ first row contains column names

☒ parse numerical values

☐ include empty strings

- Sauvegarder le fichier csv sur la machine :

Edit write file node

Delete

Cancel

Done

⚙️ Properties

⚙️

📄

🔗

📄 Filename

▼ path

C:\Users\ilyass\Desktop

🔄 Action

append to file

▼

☒ Add newline (\n) to each payload?

☐ Create directory if it doesn't exist?

🚩 Encoding

default

▼

🏷️ Name

C:\Users\ilyass\Desktop\Arduino_Data.csv

Tip: The filename should be an absolute path, otherwise it will be relative to the working directory of the Node-RED process.

- Debugger le résultat de chaque fonction :

Edit debug node

DeleteCancelDone

Properties

Output

▼ msg. payload

To

☒ debug window

☐ system console

☐ node status (32 characters)

Name

debug 1

- Voici le résultat final de flux des données dans Node Red :

