

Phase 3 - Programmation

Phase 3 - Programmation de la station

Objectifs

Compétences techniques:

- Programmation pour stocker les valeurs de température et d'humidité dans la mémoire de l'Arduino Nano.

Déroulement de la séance

1. Introduction:

- Présentation des éléments nécessaires pour la programmation.
- Importance de la programmation dans le projet de station météo.

2. Programmation:

- Commençons par la programmation sur Mblock.

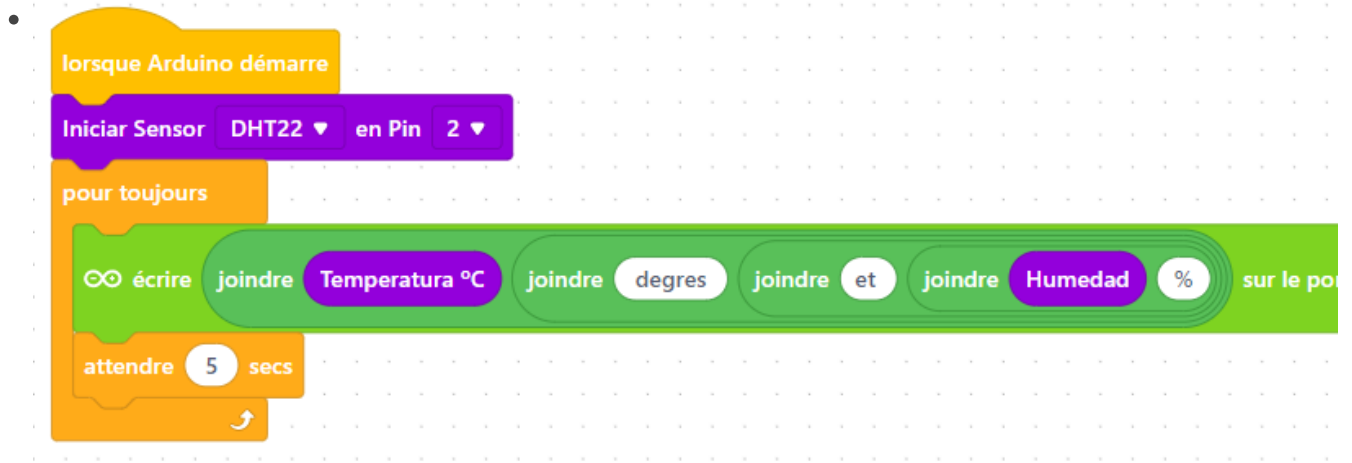


- N'oubliez pas d'installer l'extension pour le capteur en cliquant sur ce bouton [extension](#)

, puis taper dans la barre de recherche "DHT22" et ensuite juste besoin de l'ajouter.



- Maintenant vous avez juste besoin de faire le code qui permet de mesurer la température grâce au DHT 22 et qui la renvoie sur le moniteur série.



- Avant de téléverser le code n'oubliez pas de décocher ces 2 cases ci-dessous qui se situe dans le moniteur série.

☐ Envoyer en hexadécimal
 ☐ Réception hexadécimale

- Pour la suite nous allons passer sur le logiciel "Arduino IDE" car il nous offre plus de possibilités.
- Dans un premier temps télécharger la librairie Arduino DHT de Adafruit

<https://www.arduino.cc/reference/en/libraries/dht-sensor-library/>

- bien installer la dépendance Adafruit Unified Sensor

DHT sensor library by Adafruit

1.4.6 installed

Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors
 Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors
[More info](#)

1.4.6 ▼

REMOVE

```
#include <SoftwareSerial.h>
#include <DHT.h>
#include <EEPROM.h>

#define DHTPIN 9
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

const int EEPROM_SIZE = 512;           // Taille de l'EEPROM en octets
const int ENTRY_SIZE = sizeof(float) * 2; // Taille d'une entrée (température + humidité)
const int NUM_ENTRIES = EEPROM_SIZE / ENTRY_SIZE; // Nombre total d'entrées possibles
```

```

int currentAddress = 0; // Adresse actuelle pour stocker
bool storageFull = false; // Indique si l'EEPROM est pleine

unsigned long previousMillis = 0; // Dernier temps de stockage
const unsigned long interval = 3600000; // Intervalle d'une heure (en millisecondes)

void setup() {
    Serial.begin(9600);
    dht.begin();
    Serial.println("Démarrage du système...");
}

void loop() {
    unsigned long currentMillis = millis();

    // Vérifie si une heure s'est écoulée
    if (!storageFull && currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis;

        // Lecture des données du capteur
        float temperature = dht.readTemperature();
        float humidity = dht.readHumidity();

        if (!isnan(temperature) && !isnan(humidity)) {
            // Vérifie si suffisamment d'espace reste dans l'EEPROM
            if (currentAddress + ENTRY_SIZE <= EEPROM_SIZE) {
                // Stocker la température
                EEPROM.put(currentAddress, temperature);
                currentAddress += sizeof(float);

                // Stocker l'humidité
                EEPROM.put(currentAddress, humidity);
                currentAddress += sizeof(float);

                // Affichage des données
                Serial.print("Données stockées à l'adresse ");
                Serial.print(currentAddress - ENTRY_SIZE);
                Serial.print(": Température = ");
                Serial.print(temperature);
                Serial.print(" °C, Humidité = ");
            }
        }
    }
}

```

```

        Serial.println(humidity);
    } else {
        storageFull = true; // Indiquer que l'EEPROM est pleine
        Serial.println("EEPROM pleine. Arrêt du stockage.");
    }
} else {
    Serial.println("Erreur de lecture du capteur DHT22.");
}
}
}
}

```

- Ce code va nous permettre de stocker la température et l'humidité dans la mémoire de notre Arduino nano et ça toute les heures.
- après avoir stocker les température et l'humidité il faut les récupérer, voici le code :

```

#include <EEPROM.h>

const int EEPROM_SIZE = 512;          // Taille totale de l'EEPROM en octets
const int ENTRY_SIZE = sizeof(float) * 2; // Taille d'une entrée (température +
humidité)
const int NUM_ENTRIES = EEPROM_SIZE / ENTRY_SIZE; // Nombre total d'entrées
(température + humidité)

void setup() {
    Serial.begin(9600);
    Serial.println("Lecture des données stockées dans l'EEPROM...");

    for (int i = 0; i < NUM_ENTRIES; i++) {
        int address = i * ENTRY_SIZE;

        // Lecture des données de l'EEPROM
        float storedTemperature, storedHumidity;
        EEPROM.get(address, storedTemperature);
        EEPROM.get(address + sizeof(float), storedHumidity);

        // Vérification et affichage des données
        Serial.print("Entrée ");
        Serial.print(i + 1);
        Serial.print(" à l'adresse ");
        Serial.print(address);
    }
}

```

```

Serial.print(": ");

if (!isnan(storedTemperature) && !isnan(storedHumidity)) {
    Serial.print("Température = ");
    Serial.print(storedTemperature);
    Serial.print(" °C, Humidité = ");
    Serial.print(storedHumidity);
    Serial.println(" %");
} else {
    Serial.println("Valeurs non valides ou non initialisées.");
}
}

void loop() {
    // Pas besoin de code dans loop pour cette démonstration
}

```

3. Validation:

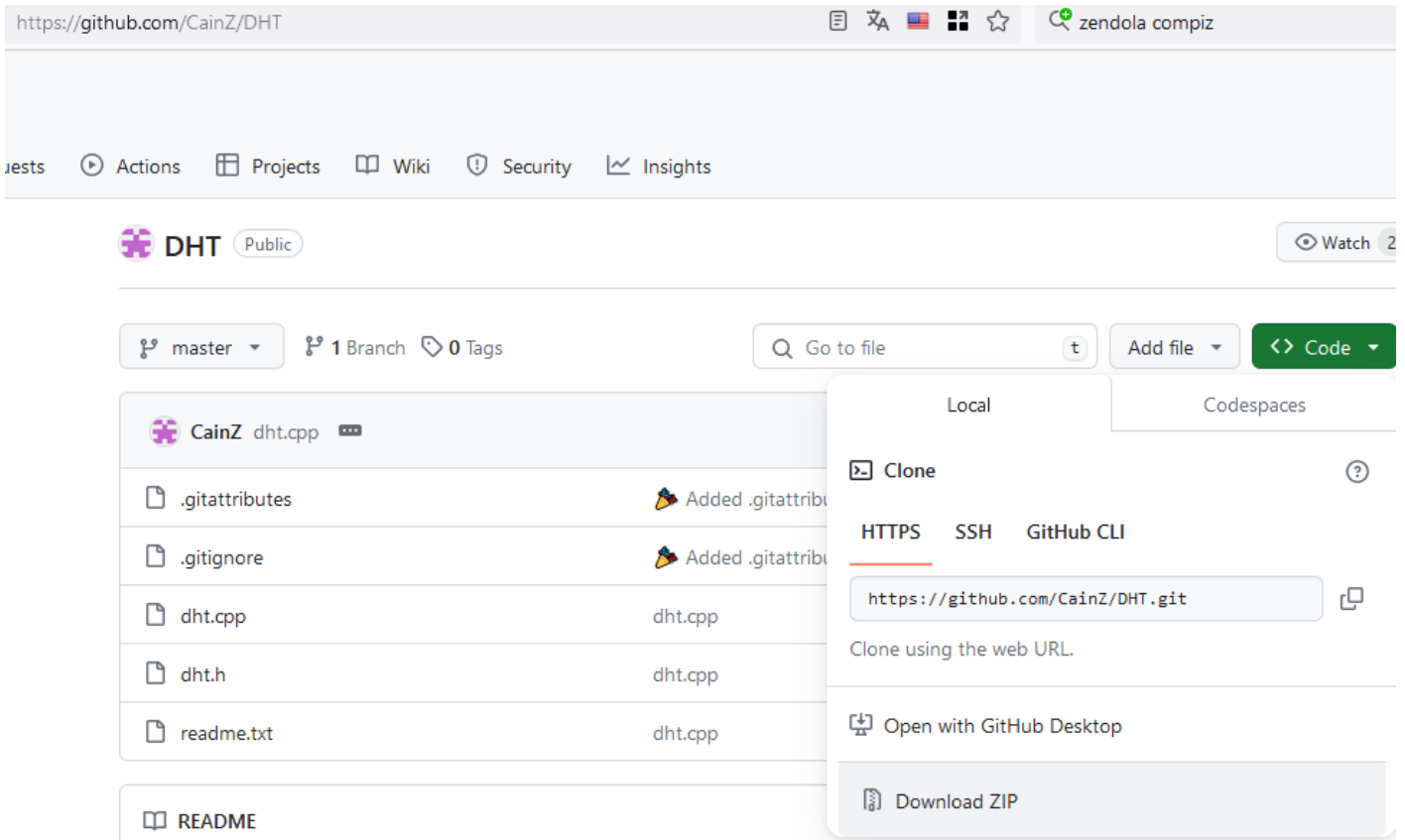
- Vérification du programme pour s'assurer du bon fonctionnement.
- Test du stockage des valeurs dans la mémoire EEPROM.

4. Conclusion:

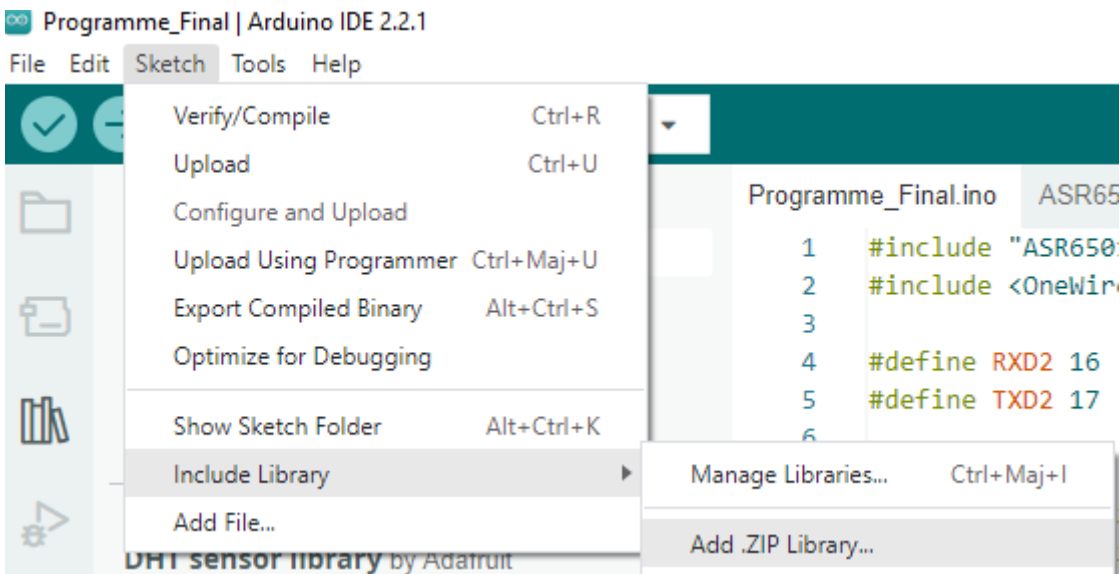
- Récapitulation des points clés de la séance.
- Réponses aux questions éventuelles.

Résolution de problèmes

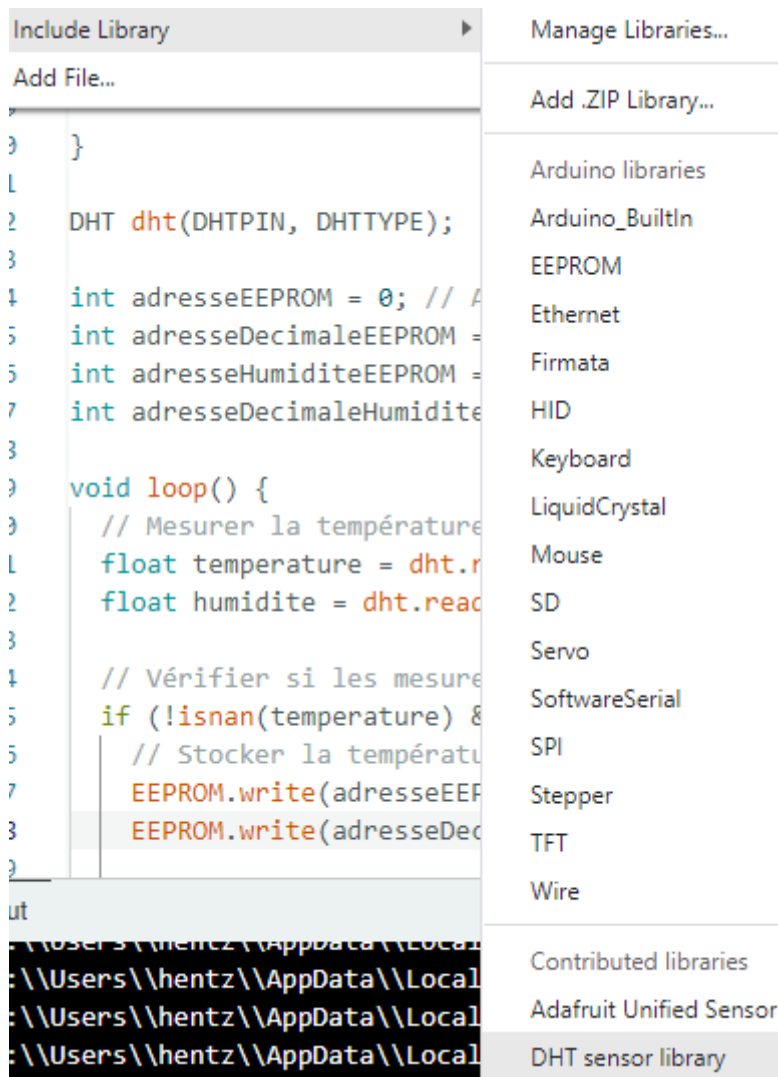
- Si `dht.h` n'est pas reconnu quand vous vérifiez/compilez le fichier, essayer une librairie adaptée à la marque de votre DHT22. Par exemple pour DFRobot :
 - https://wiki.dfrobot.com/DHT22_Temperature_and_humidity_module_SKU_SEN0137#target_2
- Cliquer sur Code > Télécharger ZIP <https://github.com/CainZ/DHT>



- Ajouter la librairie à la main dans ArduinoIDE



- Vérifier la bonne installation de la librairie, dans "contributed libraries"



Revision #14

Created 14 February 2024 08:19:54 by Quentin Petrazoller

Updated 3 December 2024 13:29:16 by Quentin Petrazoller